# *PI System Architecture, Planning and Implementation Course*

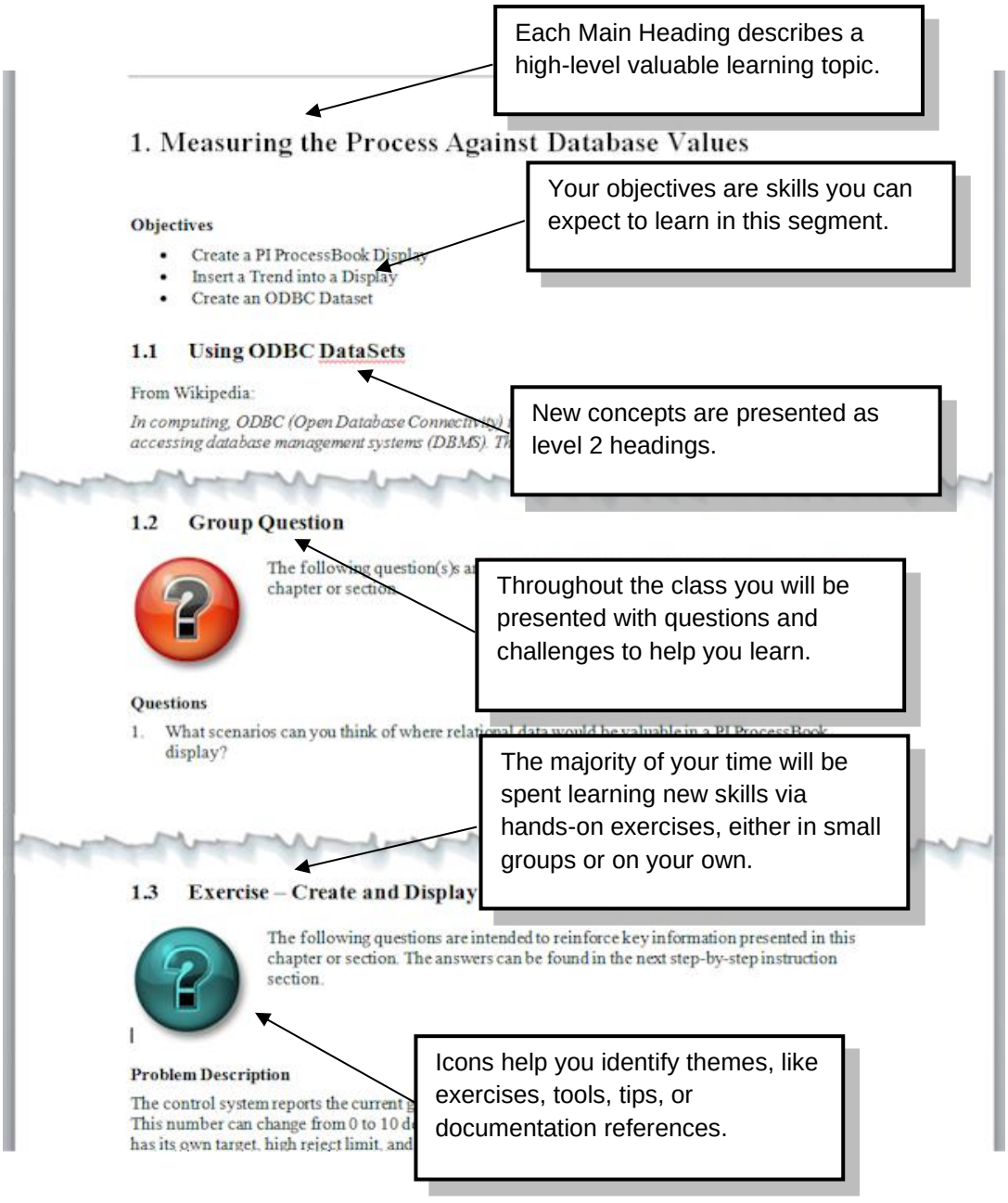
***Version 2020***

**OSIsoft, LLC**
**1600 Alvarado Street**
**San Leandro, CA 94577**

# Table of Contents

Compiled: 18 November 2020

# How to Use this Workbook




User manuals and other materials used in class can be downloaded from https://my.osisoft.com. Login to an OSIsoft customer portal account is required.

## Software Versions Used in this Course and Document

PI Software versions used in this course are:

| Software | Version |
|---|---|
| PI Data Archive | 2018 SP3 (3.4.430.460) |
| PI System Management Tools | 2018 SP3 (3.6.3.41) |
| PI Collective Manager | 2018 SP2 (1.4.3.2) |
| PI Asset Framework Server | 2018 SP3 (2.10.6.195) |
| PI Asset Framework Client | 2018 SP3 Patch 1 (2.10.7.283) |
| PI Buffer Subsystem | 2018 SP2 (4.8.0.18) |
| PI Interface Configuration Utility | 1.5.0.7 |
| PI Interface for OPC DA | 2.6.15.3 |
| PI Interface for Random Simulator | 3.5.0.2 |
| PI Interface for RampSoak Simulator | 3.5.0.2 |
| PI Data Collection Manager | 2.5.19.0 |
| PI Connector Relay | 2.5.19.0 |
| PI Connector for OPC UA | 2.0.1.33 |
| PI Vision | 2019 (3.4.0.7) |
| PI ProcessBook to PI Vision Migration Utility | 2019 (1.0.0.3) |
| PI SQL Data Access Server (RTQP Engine) | 2018 SP3 (1.7.19246.2) |

### PI Reference Library

*PI Server Installation and Upgrade Guide*

Explains how to install, upgrade or restore a PI Data Archive and PI Asset Framework.

*PI Data Archive Introduction to System Management Guide*

Provides information on how to manage the many aspects of a PI Data Archive.

*PI Data Archive Reference Guide*

The *PI Data Archive Reference Guide* provides a reference for the PI Data Archive command line utilities. It also discusses PI Data Archive database files, performance counters, and messages.

*PI Data Archive Security Configuration Guide*

This manual explains how to set up Windows Integrated Security on PI Data Archive servers, how and when to create PI Mappings and PI Trusts, and how to improve PI Data Archive security.

It discusses built-in Data Archive Identities and provides a table of access permissions required for performing different PI Data Archive tasks, as well as access permissions required by specific PI products.

*PI Data Archive System Management Guide*

This book provides detailed instructions for configuring, maintaining, and troubleshooting your PI Data Archive. It also discusses other PI components that are relevant to Data Archive system management. These include PI Interfaces as well as client tools that can be used to manage your PI System.

*PI Data Archive Applications User Guide*

PI Data Archive Applications are a set of processing tools that help you get more out of your data by automating specific processes.

*PI Asset Framework Client Installation and Upgrade Guide*

Explains how to install or upgrade PI AF Clients such as PI System Explorer or PI Builder MS Excel plug-in.

*PI Builder User Guide*

Provides instructions how to use the functions of PI Builder MS Excel plug-in like creating, editing or deleting PI Points or Asset Framework structure.

*PI System Explorer User Guide*

Provides guidance through all functionalities of PI System Explorer application.

*PI Interface for OPC DA User Guide*

Provides detailed information about this specific interface and describes all configuration options. Important document to set up the interface correctly.

*DCOM Configuration Guide*

Helps with configuration of DCOM security between PI Interface for OPC DA and remote OPC Server.

*PI Connector Administration User Guide*

Explains the architecture, installation and configuration of PI Connector Relay and PI Data Collection Manager and how to register the PI Connectors to start data collection.

*PI Vision Installation and Administration Guide*

Provides guidance how to install, configure and administer PI Vision for administrators.

*PI Vision User Guide*

Explains functionalities of PI Vision for user and how to build the visualization display.

### Terminology change

OSIsoft is revising its terminology to reflect the growth of the PI System from its original single-server architecture. In the revised terminology, **PI Data Archive** refers to the component that stores time-series data (formerly called *PI Server*), and **PI Server** refers to both *PI Data Archive* and *PI Asset Framework*. This document uses the revised terminology.

# 1. PI System Basics

## 1.1 Course structure

The course consists of instructor lead training with individual and group activities, coupled with student exercises. You will need to be familiar with your own corporate network topology, and the real-time data environment; including where the data originates and who can best take advantage of the dissemination of this data.

During the course, you will:

- Install and configure a simple PI System (PI Server, PI Interface for OPC DA, PI Connector for OPC UA and PI Vision).
- Modify this simple system to form a **High Availability PI System** with redundant PI Data Archives and PI Interfaces for OPC DA.
- Learn about different PI System architecture deployments.

## 1.2 Why PI? (What problem are you trying to solve?)

Many businesses realise there is a need to take closer control of their production processes. Instead of weekly reports rolled up monthly, CEO's are now requesting timely situation reports so that efficiencies are introduced to ensure the profitability of the company. You have been selected for the task of getting the key performance indicators and process data to the CEO more quickly than is currently the case. Your site will be used as a pilot and when proven, your solution will be rolled out across the enterprise.

Any solution must address the needs of all stakeholders - operators, process engineers, maintenance engineers, site management and enterprise management – as efficiently as possible. Current SCADA system operator stations are too expensive to roll out to non-operations employees.

Before going too much further you must decide exactly what type of data the stakeholders need and just where that data can be found in your organisation. All too often, planning and analysis reports are created with static and inconsistent sources of information – so-called 'silos of information'. Data may be currently locked in spreadsheets or a customized application interface or is on the SCADA system and not easily accessible by mere mortals. You need to unlock the source of this data and present it to all, easily, and in a timely manner.

**This course will lead you into an examination of your users' data requirements and will examine the infrastructure required to meet those requirements. You will then install components of the PI System that demonstrate the way those sections interact to provide you with skills that you use in your own environment.**

First let's examine what the PI System is.

## 1.3 What is the PI System?

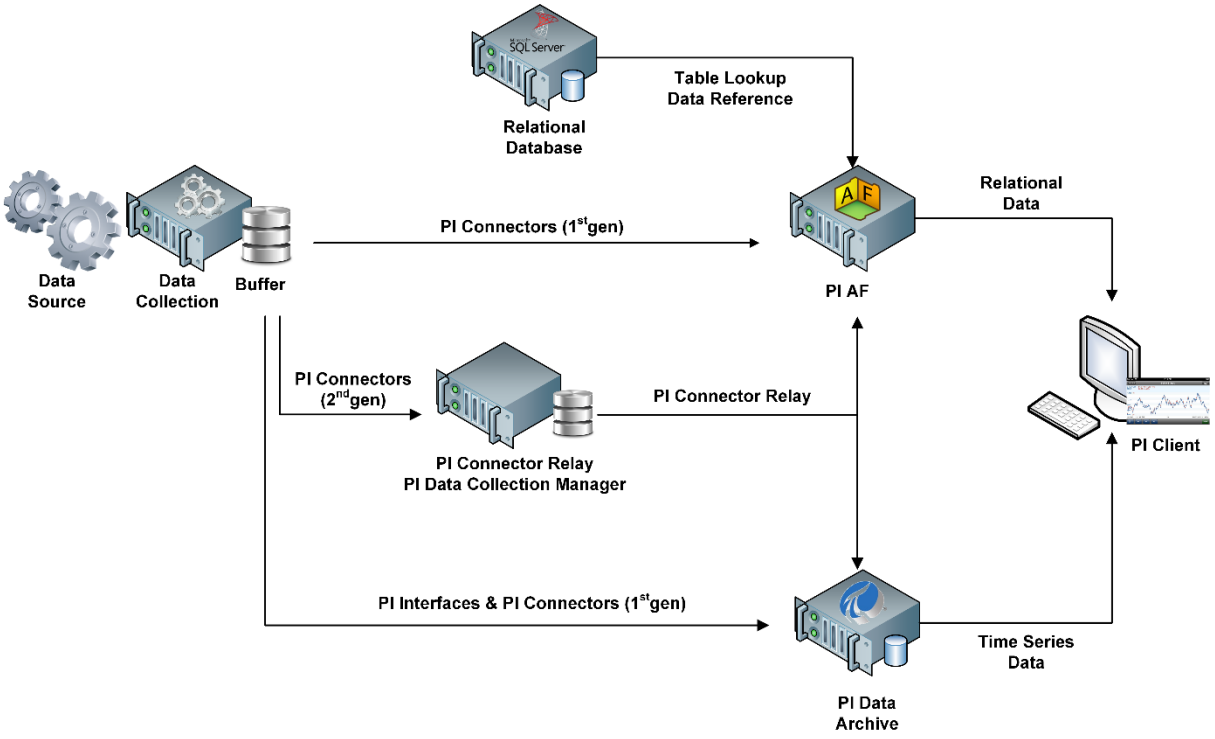
### Objectives

- Define the components of the PI System
- Draw a diagram of the architecture of the PI System

### 1.3.1 The PI System Described

The PI System collects, stores, and manages data from your plant or process. You connect your data sources to one or more Data Collection computers. In turn, the Data Collection computers get the data from your data sources and send it to the **PI Data Archive**. The **PI Asset Framework** (**PI AF**) server is one method of accessing the PI Data Archive server, and there may be other PI and non-PI servers. Users request data from the PI AF Server or PI Data Archive for display in the client tools.

Generally, the parts involved in a PI System are:




- Source data is collected by a **PI Interface** or **PI Connector** application hosted on the data acquisition computer.
- From the **PI Interface** the time series data is sent to the PI Data Archive server (asset data is referenced on the PI AF server) and stored.
- From the **PI Connector (1st gen)** the data is sent to the PI Data Archive with automatic PI Point creation and the data structure from data source is sent to PI AF

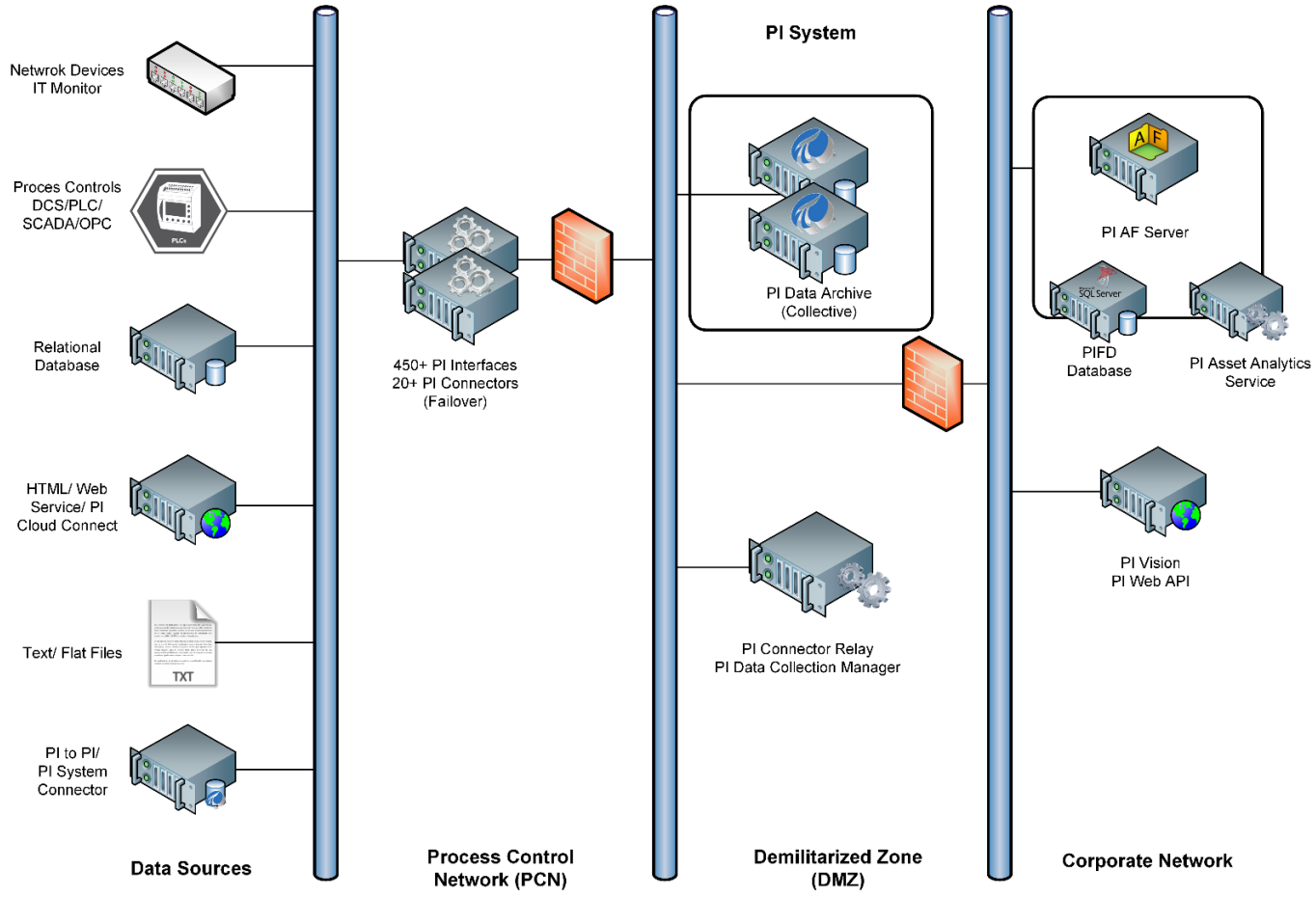
- From **PI Connector (2nd gen)** the data is sent to **PI Connector Relay** and from it the data is forwarded to PI Data Archive and PI AF. PI Connector (2nd gen) configuration is managed via **PI Data Collection Manager**.
- Data is read from the PI System (PI Data Archive and/or PI AF server) by any component of the PI Visualization Suite, such as PI ProcessBook, PI DataLink or PI Vision.

### 1.3.2 Architecture of a Typical PI System

The PI architecture may be very simple. Some customers have as few as one interface or connector feeding data to a PI Data Archive server. Everyone reads the PI Data Archive for their data.

More complex systems may include PI AF, PI Notifications, PI Asset Analytics, PI Event Frames, browser tools, redundancy and failover.

**PI Server = PI Data Archive + PI Asset Framework**



### 1.3.3 Architecture of PI System Architecture, Planning and Implementation Course

In this class, each student has a virtual environment consisting of five servers:

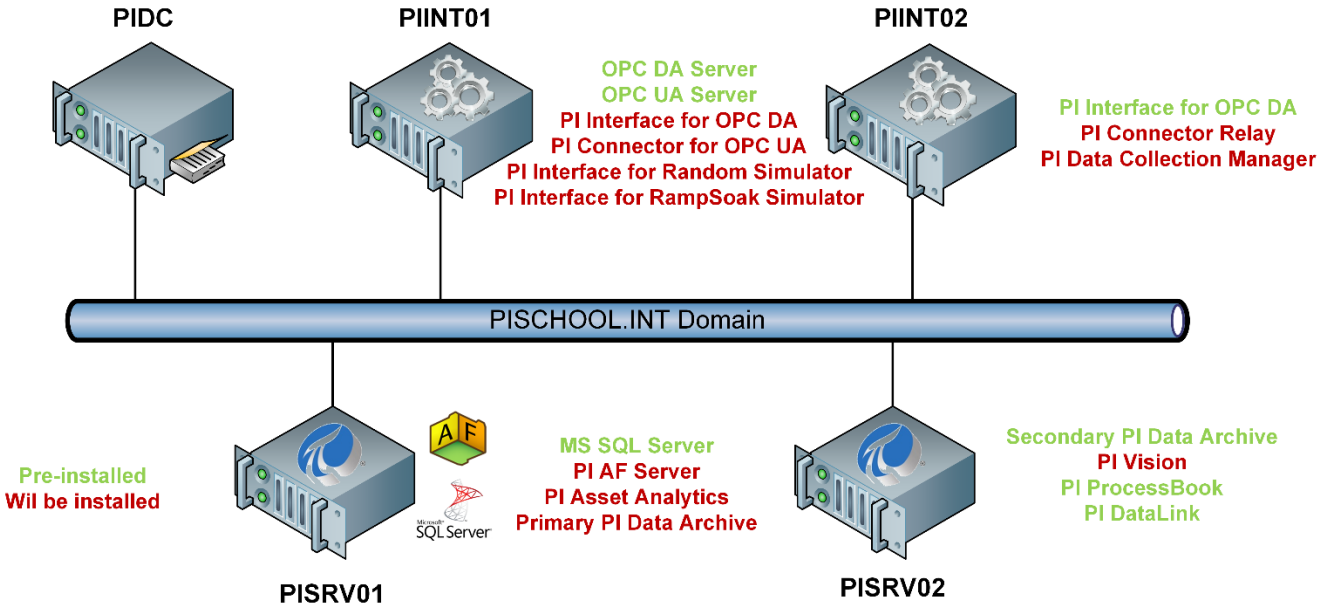
**PISRV01** – Pre-installed MS SQL Server,
PI AF, PI Asset Analytics and PI Data Archive for primary member of PI Collective will be installed.

**PISRV02** – Pre-installed PI Data Archive for the secondary member of PI Collective, pre-installed PI ProcessBook and PI DataLink and Web Server (IIS). PI Vision will be installed.

**PIINT01** – Pre-installed OPC DA and OPC UA Servers,
PI Interface for OPC DA and PI Connector for OPC UA will be installed.

**PIINT02** – Pre-installed PI Interface for OPC DA for failover. PI Connector Relay and PI Data Collection Manager will be installed.

**PIDC** – Domain Controller. This server will not be touched during the class.



## 1.4 PI Time Syntax

In PI there are two ways to specify time:

**Fixed Time**: An expression that signifies a specific date and time. Used when you want to save a view of your PI System data for a specific time in history.

*Example*: A user is creating a report that analyses an equipment failure event which occurred on the 25th of July 2013 at 11 am, so 25-Jul-2013 11:00:00 AM

**Reference Time**: An expression that signifies a date and time relative to the current date and time. Used when you want to create a dynamic view of your data, which can be used to view data in real-time, or re-used on a periodic basis to create periodic reports.

*Example*: A user creates a report that summarizes weekly production totals. By using relative time expressions, the user will be able to re-use this report every week, so define a start date of "Monday" meaning start the report from last Monday.

Both Fixed Time and Reference Time can be used with Time Offsets; Time Offsets can be used alone.

### Fixed Time Syntax

A fixed time expression is an expression which includes a date, and optionally a time.

When the time component is omitted, **Midnight** is assumed. And midnight occurs at the beginning of the day, not the end.

| Expression | Meaning |
|---|---|
| 5-jan-92 12:34 | 12:34 p.m. on January 5, 1992 |
| 25-sep-12 | 00:00:00 (midnight) on September 25, 2012 |

The PI System interprets many different formats for fixed time. In the event of an ambiguous input, the Windows Region and Language settings of the computer where the PI Visualization Tool is installed take precedence.

Note the following:

| Expression | Region and Language Format | Meaning |
|---|---|---|
| 1/5/2015 | English (United States) | 00:00:00 (midnight) on January 5th 2015 |
| 1/5/2015 | Rest of the world | 00:00:00 (midnight) on May 1st 2015 |

### Reference Time Syntax

A reference-time abbreviation represents a time relative to the current time.

| Abbreviation | Meaning | Reference time |
|---|---|---|
| * | Now | Current time |
| t | today | 00:00:00 (midnight) of the current day |
| y | yesterday | 00:00:00 (midnight) of the previous day |
| fri | Friday | 00:00:00 (midnight) on the most recent Friday |
| may | May | 00:00:00 (midnight) on the current day in May of the current year |
| apr-15 | april-15 | 00:00:00 (midnight) on the *15*th day of April in the current year |
| *YYYY* | Year | 00:00:00 (midnight) on the current day and month in year *YYYY* |
| *M-D* or *M/D*<br>*Or*<br>*D-M, D/M* | USA<br><br>The world | 00:00:00 (midnight) on the *D*th day of month *M* in the current year |
| 15 | | 00:00:00 (midnight) on the *15*th day of the **current** month |

Use the first three letters as an abbreviation for any day of the week and any month of the year. For example:

| Expression | Meaning |
|---|---|
| thu | 00:00:00 (midnight) on the most recent Thursday |
| MAR | 00:00:00 (midnight) on the current day in March of the current year |

### Time Offset

When specifying PI time use specific abbreviations that represent time units. These are used in constructing *Time Offsets* as in the table.

| Abbreviation | Time Unit |
|---|---|
| s | second |
| m | minute |
| h | hour |
| d | day |
| mo | month |
| y | year |
| w | week |

Specify the abbreviation, the full-time unit or the plural version of the time unit, such as *s*, *second*, or *seconds*. Time offset is any of the time units with a valid value and a + or – sign included, e.g. +8h.

Time offsets can be used alone in a time field or come with a fixed time or reference-time abbreviation.

### Time Offset Syntax

### Reference Time or Fixed Time and Offset Expression

When included with a reference-time abbreviation or with a fixed time, a time offset adds or subtracts from the specified time (indicated by either + or -) and a time unit with a value

| Expression | Meaning |
|---|---|
| *-1h | One hour ago |
| t+8h | 08:00:00 (8:00 a.m.) today |
| y-8h | 16:00:00 (4:00 p.m.) the day before yesterday |
| mon+14.5h | 14:30:00 (2:30 p.m.) last Monday |
| sat-1m | 23:59:00 (11:59 p.m.) last Friday |
| 1-Jan-20 – 1d | Midnight 31st December 2019 |

### Time Offsets Used Alone

Entered alone in a time field, time offsets specify a time relative to an implied reference time. The implied reference time depends on the field where you enter the expression:

- For a start time, the reference time is the current clock time.
- For an end time, the reference time is the start time.
- For a single timestamp, the reference time is the current clock time.

| Time field | Expression | Meaning |
|---|---|---|
| Start time | -1d | One day before the current clock time (24 hours before the current clock time) |
| End time | +6h | Six hours after the start time |
| End time | -30m | 30 minutes before the start time |
| Time stamp | -15s | 15 seconds before the current clock time |

### Rules to Remember

**Rule 1:** You can only include a single time offset in an expression. Including multiple offsets can lead to unpredictable results. For example, the following time expressions are not valid:

*+1d+4h

t-1d+12h

**Rule 2:** To define a time offset you must include a valid value with any time unit. Only for *seconds*, *minutes*, or *hours*, you can specify a fractional value. You cannot specify fractional values for other time units.

**Rule 3:** A fixed timestamp consists of the fields of Year, Month, Day and Time (hours, minutes and seconds). If any of these fields are not specified in the PI time expression, the following values will be assumed by default:

- If Time is not specified, then the default value would be Midnight.
- If Day is not specified, then the default value would be Current Day.
- If Month is not specified, then the default value would be Current Month.
- If Year is not specified, then the default value would be Current Year.

# 2. PI System Environment Architecture

**Objectives**

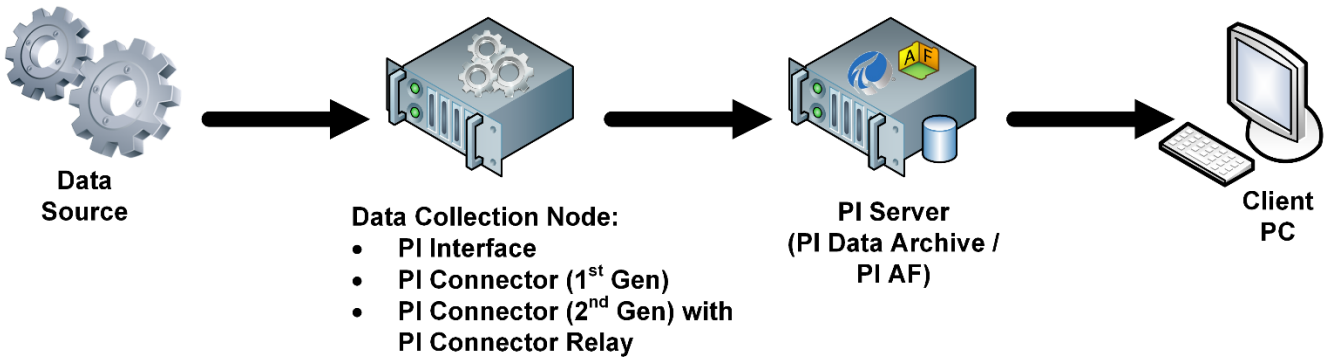
- Describe the PI System components
- Categorise users
- Examine data in the PI Data Archive and the PI AF Server
- Explain where each component can fit into your Enterprise
- List the pros and cons of the various client tools

In this section, you will investigate your users and the data they require and develop an understanding of the IT network topology required to support them.

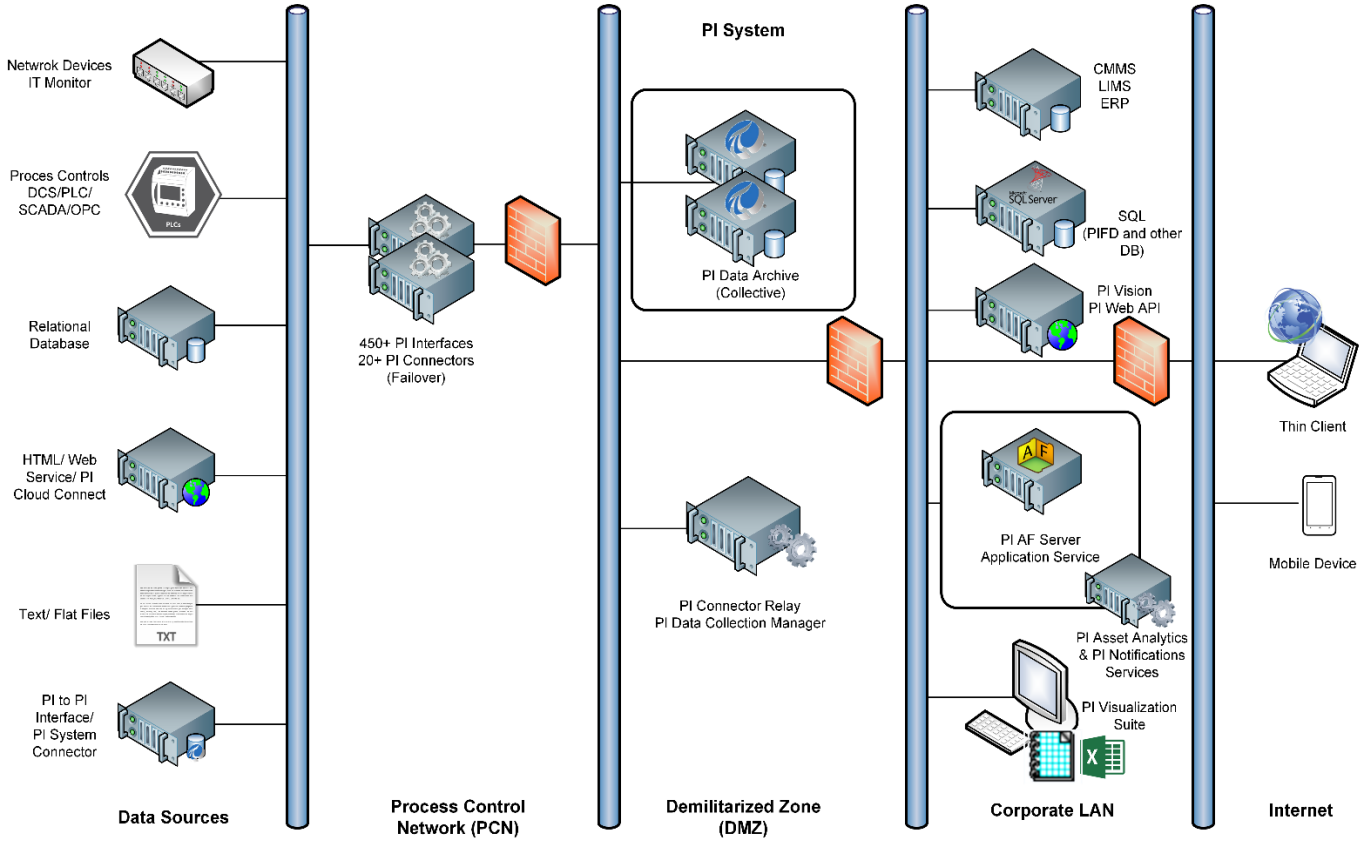
## 2.1 Simple PI System

At its simplest, PI System is a data infrastructure. A simple PI System consists of:

- The data source.
- The data collector for that data source. This may be PI Interface, PI Connector (1st Gen) or PI Connector (2nd Gen) that requires additional PI Connector Relay application.
- The PI Data Archive combined with its PI Asset Framework server.
- An appropriate visualization tool on a PC or web browser.



The PI System environment may be expanded to become a system like that represented below:




**Important:** The PI System environment does not start with the system – it starts with the users and what they need from their data.

## 2.2 Directed Activity – Identify the Parts

In this part of the class, you will perform a learning activity to explore the different concepts presented in this chapter or section. Your instructor will have **directions**.

**Activity Objectives**

- Describe the PI System components
- Define the purpose of each component

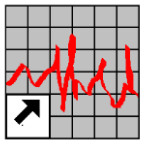
**Approach**

Using the image on the previous page, answer the following questions:

1. Why are there **two Server** applications in a PI Server?

2. What is the difference between smart and thin clients? Advantages or disadvantages? Use the table at the next page to write down your ideas.

3. List some of the ways data gets into the PI System.

| | | Pros | Cons |
|---|---|---|---|
| Thick |  Process Book | | |
| |  PI DataLink | | |
| Thin |  PI Vision | | |

## 2.3 Back to the Data - What Do Your Users Require?

Users' data is any data associated with the diverse equipment required in the modern business environment. This may be data such as:

- pump RPM,
- pump serial number,
- the date the pump had its impellor changed,
- running hours of the pump,
- calculated efficiency of the pump.

It can also record:

- the ping time between computers in corporate network,
- server CPU temperature,
- the memory requirements of a monitored business application.

Users may also require information about:

- equipment shutdown times,
- in-stream analysis,
- shift production limits,
- production totals,
- cost of production,
- data with time stamps in the future,

to name a few types of data.

The data are grossly categorized as:

- real-time data,
- derived (calculated) data,
- grouped (relational) data, or
- future data

Because of the differences in the rate of change of data, it is appropriate to store the data in different ways.

- Real time & future data is best stored in a database that allows for fast storage, fast retrieval and automatic compression such as the **PI Data Archive**.

- Slow changing or static data may be better stored in relational databases structures, or the asset centric elements found in the **PI Asset Framework** (**PI AF**) database.
- Calculated data may be stored in the PI Data Archive or derived as required in the PI AF database with **PI Asset Analytics**.

Numeric data stored in the PI Data Archive may be **compressed** to compact the historian, and to facilitate data retrieval. All data stored in the historian is kept in files called **archives**. When the CEO requests data, the structure of the archives facilitates the rapid retrieval of that data.

On the other hand, data in the PI AF can be:

- Stored in the associated relational database.
- Derived as required from attributes and elements (**PI Asset Analytics**) and stored back into PI Data Archive, or
- Passed through from the PI Data Archive historian.

## 2.4 Combining Different Types of Data

The AF server provides several functions that are useful for building applications upon PI data and other data. The most significant is asset centric and allows users to organize and structure PI data, as well as other, non-real time data according to objects with which the users are familiar. These can be physical objects in their production processes like pumps, transformers, grinding mills, or network hardware. These asset **elements** are flexible and allows hierarchical modes. PI AF asset elements can span PI Data Archives, allowing users to organize and search for PI information across multiple PI systems.

The element is a user-oriented object that contains **attributes**, which reference

- Time-series data from PI Data Archive,
- configuration data,
- data from dissimilar systems.

Other features of PI AF that allow users to leverage PI data include

- object-level security,
- searches,
- access to data from systems other than PI,
- the ability to scale to 100 million element attributes or more.

PI AF also provides an underlying infrastructure for OSIsoft products such as:

- PI Notifications,
- PI Asset Analytics,
- PI Vision,
- PI OLEDB Enterprise,
- PI SQL Client (ODBC, OLEDB, JDBC)
- PI Web API.

*You do not have to configure your entire PI AF installation at once*. Start with a manageable area, such as same type of equipment, or part of a process, and expand the “model” over time.

When planning an PI AF installation, consider the following questions:

- How will you map your company’s assets to AF elements?
- What attributes are required for each element?
- Are these attributes PI data, constants, or calculations?
- Will you use **templates** to speed up definition of these elements?
- How will the asset hierarchy be structured?
- Do you need data from other databases as well as real-time (PI) data?
- What security is required?

The **PI System Explorer** (**PSE**) is a tool used to see AF data. It connects to one AF database at a time. In general, OSIsoft recommends the use of one AF database for your entire asset framework. However, depending on your application, you may need multiple databases:

- Within a single company, different departments may prefer their own unique databases to handle their needs.
- A company with a database that handles current operations may have a second database for testing purposes. Modifications implemented in the future are made and tested on a testing database before transferring the structure to the production database.

PI AF requires a Microsoft SQL Server instance available at the time of installation. PI AF supports all SQL Server editions. Express edition can be installed on a shared computer, together with PI AF and/or PI Data Archive software, or on a remote computer. MS SQL Server Express edition comes with a free license. Scalability limitations make it more suitable for small to medium-size installations.

With PI Connectors, the structure on the data source is replicated to the PI AF automatically.

Consult the *PI Server Installation and Upgrade Guide* for details.

Since PI Server 2018 unified installer, PI AF server is **NO longer** a required prerequisite for PI Data Archive server installation.

## 2.5 Data Sources

Data sources can be any measuring device generating data. They can be almost anything and can connect to the interface data collection computers in a variety of ways. Examples of data sources are:

- Distributed Control Systems (DCS's),
- Programmable Logic Controllers (PLC's),
- Laboratory Information Systems (LIM's),
- Supervisory Control and Data Acquisition Systems (SCADA),
- Manual Loggers,
- EPA Data Loggers,
- Network Devices,
- Business Information (BI) Systems,
- Business Planning Databases.

To acquire data from these sources it is necessary to utilize a PI Interface or PI Connector. The interface or connector may be simple such as a file reader or complex, such as the interface or connector that uses the OPC protocol. For more information on OPC, see www.opcfoundation.org.

## 2.6 Data Collection Computers

Data Collection computers (also known as nodes) run one or more PI Interfaces and/or PI Connectors. PI Interfaces collect data from the targeted data source and send the collected data to the PI Data Archive. Each different data source needs a PI Interface that can interpret the protocol used by the source. OSIsoft has over 450 different interfaces.

PI Connectors collects not only data from the targeted data source, but also checks the structure and automatically create PI Points, AF Elements and AF Attributes accordingly. Currently there are over 20 PI Connectors with more being developed.

To collect data from a source, the PI Data Archive (and PI AF Server for PI Connectors (1st gen) and PI Connector Relay) security must be configured appropriately. The PI Interface itself must be configured, as well as the attributes of the PI Data Archive point - also known as a tag, or more properly, the data stream. There is a correspondence between the metadata (attributes) of the point in the PI Data Archive database and the point's attributes in the data source. Matching the two sets of point attributes is crucial to sustainable and timely data collection by the PI Interface.

PI Connectors (1st gen) and PI Connector Relay for PI Connectors (2nd gen), as mentioned above, create PI Points automatically, therefore the correspondence between the metadata is handled by the same way. There are only several PI Points attributes, which can be modified after the creation by PI Connector or PI Connector Relay.

Necessary steps in collecting data are:

- Ensure the network between the data source and the PI Data Archive and PI AF servers is configured.
- Ensure the data source is working.
- Install and configure the appropriate PI Interface or PI Connector (1st gen).
- Or install PI Connector (2nd gen), PI Connector Relay and PI Data Collection Manager which is used for PI Connector (2nd gen) configuration.
- Create and configure points as required on the PI Data Archive for PI Interface.
- Check that the points are receiving data.

*These steps will be covered in detail later chapters.*

# 3. PI System Planning

**Objectives:**

- Discuss Highly Available Systems
- What you need to get through firewalls
- Service accounts preparation

To ensure the integrity of your PI environment steps must be taken to implement backups, redundancy and high availability of the all components.

## 3.1 PI Data Archive server High Availability

PI Data Archive High Availability (HA) enhances the reliability of the PI Data Archive by providing alternate sources of the same time-series data for users. PI Data Archive Replication enables alternate data sources by synchronizing the configuration of multiple servers. This allows interfaces to buffer data to multiple servers with the same point configuration. PI Clients can retrieve data from any of the servers without changing data references. You will be implementing a highly available architecture on day 3.

Consult the *PI Server Installation and Upgrade Guide* for details.

## 3.2 PI Interface & PI Connector (1st Gen) Failover

PI Interfaces are based on a standard interface template (UniInt) that can potentially support interface failover. Depending on the data source, an interface can automatically switch between redundant copies of the interface run on separate interface computers. This provides uninterrupted collection of process data even when one of the interfaces is unable to collect data for any reason. When maintenance, hardware failure, or network failure causes one interface to become unavailable, the redundant interface automatically starts collecting and buffering data to send to the PI Server.

UniInt interfaces can also restart without a connection to the PI Server. As the interface runs, it receives updates to the list of points and their parameters and writes the information—including the point scan list—to a local disk file. Subsequent starts of the interface can use the local copy of the point information to start up without a connection to the PI Server. This is called “Disconnected startup” and is recommended for production interfaces.

PI Connectors (1st gen) that support failover work internally on a different principle, but the goal is the same. To provide the uninterrupted data collection in case when the connection to data source from one instance is broken or the instance stops working. PI Connectors do not read

the points from PI Data Archive and do not store the point parameters in a local disk file, thus do not need connection to PI Data Archive or PI AF Server to be able to start. Therefore, we can say they always run in “Disconnected Startup”

Each PI Interface or PI Connector (1st gen) manual has comprehensive details of how to implement failover.

There is currently **NO FAILOVER** support for PI Connector (2nd gen) and PI Connector Relay.

## 3.3 Planning for Security

**Objectives**

- Introduce PI security
- Understand how network and environment affect PI Security.
- Discuss PI Mappings
- Introduce Group Managed Service Accounts (**gMSA**)

### 3.3.1 What is Security?

Computer security has two parts:

| | |
|---|---|
| Authentication | Who is the user, and how do we confirm that the user is really whom she says? |
| Authorization | What is the user allowed to do? |

Security is always a balance.


You want to use what is most secure, easy to configure, and provide for minimal Maintenance. That is why we suggest using **Windows Active Directory** security. It is something that you probably have now in your domains and can easily accommodate the PI System. We call this **Windows Integrated Security**.

To connect to the PI AF SQL database OSIsoft recommends that you use Windows authentication because it is more secure than SQL Server authentication. Objects and their effective permissions are based on the Windows user identity. The permission sets for all users

are stored as Windows security descriptors associated with certain types of objects and collections within PI AF.

### 3.3.2 Securing the Environment and the Server

The PI System is consistent with, and best supported by, implementation in a corporate network secured computing environment. This usually includes:

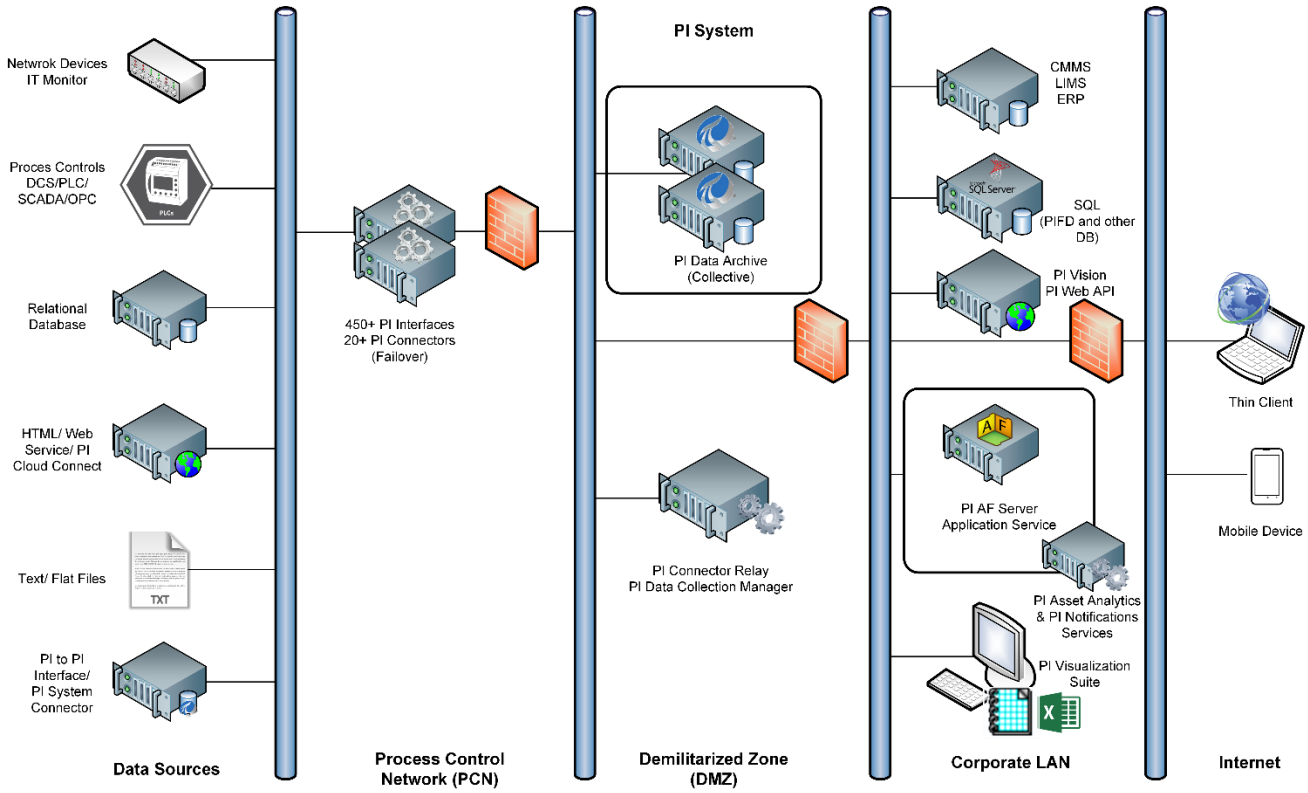
- Domain security for users, directories, and applications
- Router security including router-based firewalls
- Antivirus programs and regular operating system patches
- Controlled access by remote parties

**Fixed IP** addresses are usually applied to interface computers and server computers, while **DHCP IP** addresses is the norm for user clients. PI System is usually implemented in domains and all data communications to PI System are through TCP/IP response packets. However, PI System data can be accessed without joining the domain since file access in the PI Data Archive server from any remote computer is unnecessary. PI Server's lookup server name can be entered into the Domain Name Server associated with its fixed IP address. PI System can stand alone, sharing no files and inaccessible by anyone except the local computer administrator and users. This access can be provided by remote terminal client services.

- Install the server in a secured area with a locked door.
- Have uninterruptible power supply and air conditioning.
- Have unique passwords for both the local administrator and the *piadmin* PI user.
- Have a screen saver, which locks out idle sessions and requires password re-entry.
- Do not use the PI Server computer to access the Internet via a browser. Instead, download the required files on another computer and transfer them by flash drive or other media.
- *Do not* install client software (Microsoft Office, PI ProcessBook, etc.) on your server. *This will encourage the server's use as a client and is not recommended*.

### 3.3.3 How Things Connect to the PI System

In order to plan where things will "live" in your Enterprise it is important to understand how connections are made to the various parts of the PI System.




- PI Interfaces can connect using a **PI TRUST** or **WINDOWS INTEGRATED SECURITY** (since PI API for WIS) via Domain User Account or preferably Group Managed Service Account (gMSA) for authentication.
- PI Connectors must connect to Data Archive and AF via **WINDOWS INTEGRATED SECURITY** via Domain User Account or preferably Group Managed Service Account (gMSA) for authentication.
- Users must connect to the PI Data Archive Server using **WINDOWS INTEGRATED SECURITY.** (Legacy Explicit Login and PI Trust are not recommended) via Domain User Account.
- Users must connect to PI AF and our thin clients using **WINDOWS INTEGRATED SECURITY** via Domain User Account**.**

## What implications does this have in your domain?

## 3.4 Group Managed Service Account (gMSA)

### 3.4.1 Introduction

Although introduced in Windows Server 2012, the Group Managed Service Account (gMSA) still has low adoption within the enterprises. This section will highlight the benefits of gMSAs and how to create and configure them within the domain.

Built-in accounts such as *NetworkService* or *LocalSystem* have decent password management, are simple to use, and can easily run a service or an Application Pool. However, outside the machine boundary, they take on the identity of their host machine (COMPUTERNAME$). This limits their usefulness outside the machine itself, as granting access to a computer object inevitably gives access to all other built-in identities on the computer (and all the services they're running!).

Many companies use standard domain accounts instead, but these accounts suffer from very poor password management (entropy, expiration/resetting, maintenance). Moreover, standard domain accounts' passwords are inevitably exposed to both users (who need to regularly interact with the passwords - typing or copy/pasting) and computers (passwords are stored locally).

**gMSAs combine the best of both worlds**:

- Automatic password management with secure & centralized storage
- gMSA's password is calculated on-demand by Domain Controller's Key Distribution Service (KDS)
- Automatic password changes are done periodically.
- In contrast to passwords used by standard domain user accounts, gMSA passwords are not stored locally on computers nor exposed to users.

The main reasons why the gMSA is not widely adopted could be:

- Creation, configuration and maintenance of gMSA can only be done via *PowerShell*. There is no GUI in Active Directory Users and Computers that would allow you to create the accounts. You will only see them once created.

| Name | Type |
|---|---|
| SVC-PIAF | msDS-GroupManagedServiceAccount |
| SVC-PIANALYT | msDS-GroupManagedServiceAccount |
| SVC-PIBUFFER | msDS-GroupManagedServiceAccount |
| SVC-PICON | msDS-GroupManagedServiceAccount |
| SVC-PIDA | msDS-GroupManagedServiceAccount |
| SVC-PIDCM | msDS-GroupManagedServiceAccount |
| SVC-PIINT | msDS-GroupManagedServiceAccount |
| SVC-PIRELAY | msDS-GroupManagedServiceAccount |
| SVC-PIRTQP | msDS-GroupManagedServiceAccount |
| SVC-PIWEB | msDS-GroupManagedServiceAccount |

- gMSA denies interactive logon. The application of gMSA is limited only to Windows Services.

- Not all PI System applications accept gMSA during installation. But there are workarounds described in the exercises how to implement it.

### 3.4.2 Provisioning and deployment

In order to be able to deploy gMSA there are prerequisites need to be met:

- Windows Server 2012 or later installed on at least one Domain Controller.
- Windows Server 2012 or later installed on the machines hosting services that will use the gMSA.
- Key Distribution Service (KDS) Root Key must exist to enable gMSA creation.

Creating the KDS Root Key is done via the PowerShell command:

```
If (-Not (Get-KdsRootKey)) {
Add-KdsRootKey }
```

By default, it takes 10 hours until the KDS Root Key is in effect (to provide sufficient time to replicate the key in larger domains). However, in the course or test environment, it can be forced to be effective immediately.

```
If (-Not (Get-KdsRootKey)) {
Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10)) }
```

To create gMSA use ***New-ADServiceAccount*** PowerShell cmdlet and specify at least:

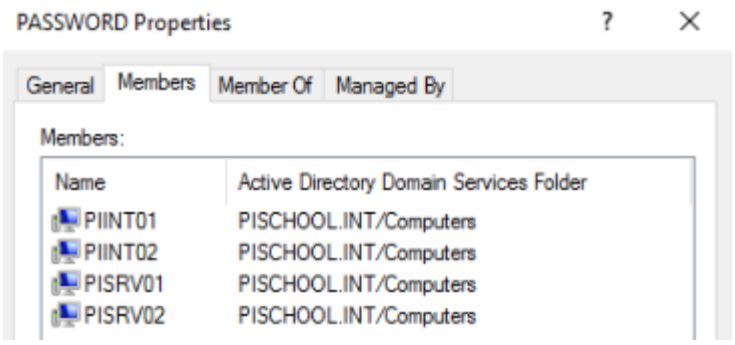
- ***Name***: gMSA account name (must be fewer than 15 characters – avoid $,#,@ etc.)
- ***DNSHostName***: gMSA DNS account name (FQDN of the gMSA)
- ***PrincipalsAllowedToRetrieveManagedPassword***: Principals allowed to obtain gMSA's password. For manageability, it's best practice to create a Domain Group (type: Security) and add the machines hosting services the that will run under the gMSA. However, it's also possible to specify the computer(s) directly

For example, Powershell command to create gMSA account for PI AF Application Service will look like this:

```
New-ADServiceAccount -Name SVC-PIAF -DNSHostName SVC-PIAF.pischool.int
-PrincipalsAllowedToRetrieveManagedPassword  PASSWORD
```

All subsequent gMSA accounts that will be used in this course were created by the same command with appropriate *-Name* and *-DNSHostName* parameter.

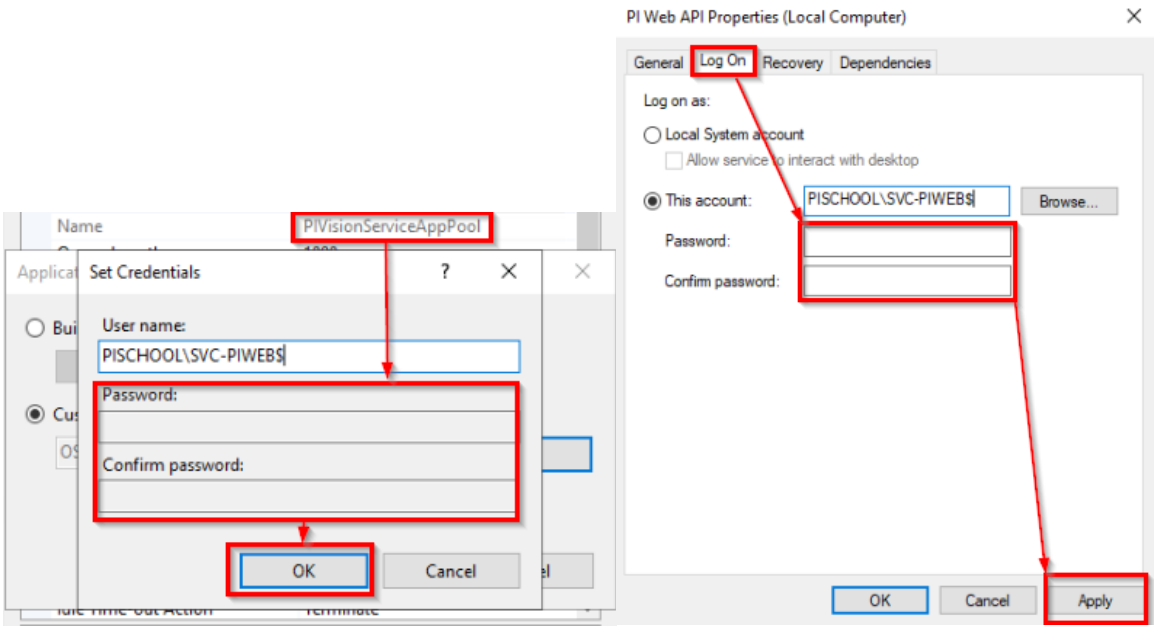
PASSWORD is the Domain Group type Security where machine accounts are added. This means, it is possible to use the gMSA account only on the servers which are members of this group. Any other server is not allowed to retrieve the password from Domain Controller.




For more information, please see Microsoft Docs page [Group Managed Service Accounts Overview](#).

### 3.4.3 gMSA in PI Environment

When specifying gMSA identity (always with **"$"** symbol in the end, same as for computer account) in *Services Manager* snap in or in *IIS Manager*, simply type in the name of the account and leave the password box blank:




After the Log On account of a Service is set to gMSA, the Log On tab will be permanently unavailable (greyed out). To fix this, open an elevated Command Prompt and execute:

**sc managedaccount** <*ServiceName*> **false**

Here is the short summary how gMSA can be applied to PI System applications:

**PI Data Archive** (2017 R2 and later)

- Use Services Manager or PowerShell to switch PI Network Manager service (pinetmgr) to run under the gMSA. Older PI Data Archives do not support custom accounts.

**PI AF Server, PI Analysis Service, PI Notifications Service, PI SQL DAS (RTQP)**

- gMSA accounts are accepted by PI Server installation kit during installation.

**PI Vision**

- Use IIS Manager or PowerShell to switch PI Vision Application Pools to run under gMSA

**PI Web API** (2017 and later)

- Run through PI Web API Admin Utility and select the default NT Service accounts.
- Use Services Manager or PowerShell to switch PI Web API and PI Web API Crawler services to run under the gMSA.
- Re-run PI Web API Admin Utility and make no changes. Utility will automatically grant all required permissions to gMSA.

**PI Interfaces**

- Use Services Manager or PowerShell to switch the service to run under gMSA

**PI Buffer Subsystem**

- If already configured: Add gMSA account to PI Buffering Administrators local group and then use Services Manager or PowerShell to switch the service to run under gMSA and restart the service
- If about to be configured: Use Services Manager or PowerShell to switch the service to run under gMSA, add the gMSA to PI Buffering Administrators local group and then finish the configuration in PI Buffering Manager.

**PI Connectors, PI Connector Relay, PI Data Collection Manager**

- Add the gMSA to PI Connector Administrators or PI Trusted Installers local group. Then use Services Manager or PowerShell to switch the services to run under gMSA.

### 3.4.4 MSA vs. gMSA

*Managed Service Accoun*t is an older type of account than *Group Managed Service Account*. It was introduced in Windows Server 2008 R2. It is working on a similar principle as gMSA, but it has limitations such as MSA account is bound only to one computer and it cannot be used on other computers. It cannot be shared in clustered services and in load-balanced services in a web farm.

# 4. PI Server Requirements

**Objectives:**

- Define the hardware requirements of the PI Data Archive
- Define the hardware requirements of the PI AF Server
- Describe the virtualization support for the PI System

## 4.1 Hardware Sizing

On OSIsoft's Customer Portal MS Excel spreadsheet *"Hardware and PI System Sizing Recommendation Spreadsheet*" can be found and downloaded (link)

This spreadsheet gives recommendations for the following hardware resources based on current or anticipated load and data volume.

- Storage Capacity
- Disk Throughput (IOPS)
- CPU Count
- Memory (RAM)
- Network Bandwidth

The amount of disk space required depends primarily on the number of PI points that the PI system will collect. See below to calculate the necessary disk space.

| PI Data Archive Component | Space Requirement | Notes |
|---|---|---|
| PI Data Archive and PI Software Development Kit (PI SDK) | 160 MB<br>100 MB temporary disk space for installation | PI SDK includes the PI API. |
| PI Data Archive databases | 11 MB per 1000 points + 1 MB | Located on the same computer as the PI Server. |
| Message log files | 10 MB | Located on the same computer as PI Server, in the pi\log directory. |
| PI Archive files | 10 MB per 1000 points | |
| PI event queue | 5 MB per 1000 points | OSIsoft recommends that you place the event queue on a different physical drive from the PI archive files. |
| Rollback backup (upgrade only) | Size of PI Data Archive databases + primary archive | The PI Archive setup program performs a minimal backup during the upgrade. |

For PI Data Archive, PI AF server, and Microsoft SQL Server, one or more Microsoft Windows compatible computers, with a 64-bit operating system is required.

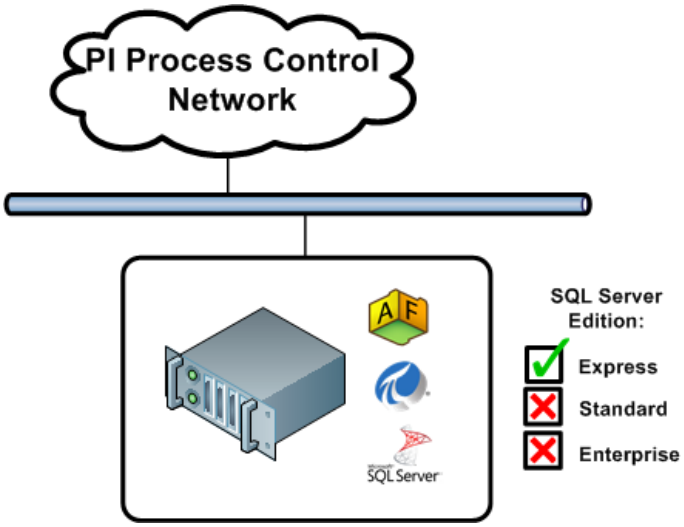
For best performance and improved security, OSIsoft recommends that you install MS SQL Server on a different computer from PI Server. If you plan to install MS SQL Server on the same node as the PI Data Archive, the SQL Express Edition should be installed so that the MS SQL Server does not compete with system resources of the PI Server. The MS SQL Express Edition is limited to using only a single physical CPU and 1 GB of RAM.

It is recommended that the PI AF server and PI Data Archive server be installed on *different computers* if:

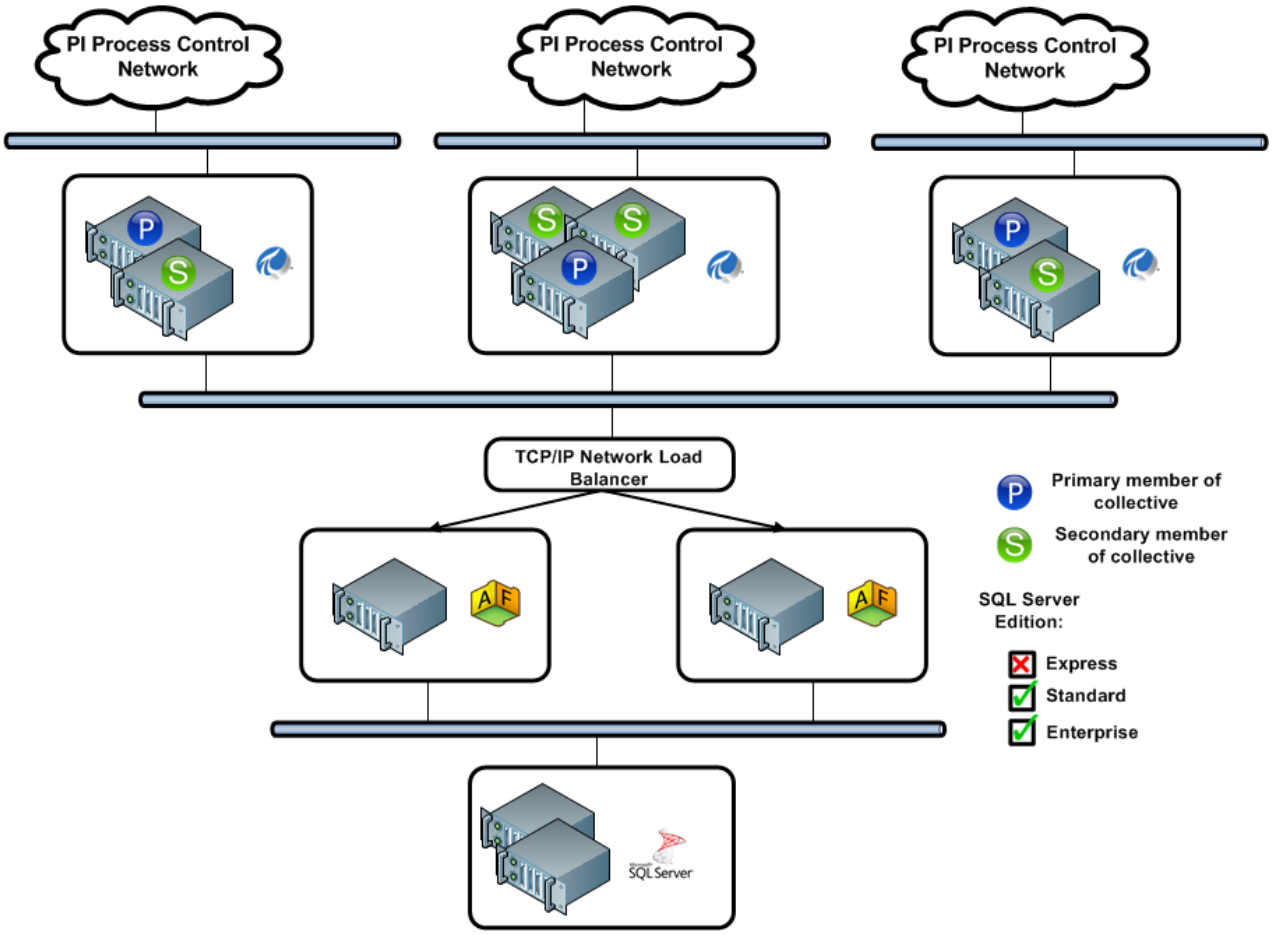
- PI AF server will use time-series data from multiple PI Servers
- PI AF server is configured for high availability (such as a load balanced PI AF server, PI AF servers connected to a mirrored MS SQL Server, or PI AF servers connected to clustered MS SQL Servers).

The number of required computers depends on the size and complexity of the PI System. The size of a PI System depends on the number of PI points and the number of units (elements) of equipment (such as mixers, tanks, or meters or whatever else you have added into the asset database).

A simple, small system will have the PI Data Archive, the PI AF Server and MS SQL Server (the free MS SQL Server Express may be used) installed on the same hardware (or virtualised) server, as shown below:




For distributed systems with large workloads and point counts, and with multiple PI Servers or PI Data Archive Server collectives that link to a central PI AF database, OSIsoft recommends that you install PI Data Archive Server collectives with two PI AF Servers with a Network Load Balancer over them, and Microsoft SQL Server on separate, redundant computers to achieve the best level of performance and scalability. This type of system is depicted below:


PI Process Control Network
PI Process Control Network
PI Process Control Network
TCP/IP Network Load Balancer
Primary member of collective
Secondary member of collective
SQL Server Edition:
Express
Standard
Enterprise
SQL Server

# 5. Installing the PI Server

**Objectives**

- Review the pre-installation check list
- Describe the steps to obtain a License File
- Know the installation steps of PI Server
- Become familiar with the directory structure of PI Data Archive (PI folder)
- Start and stop the PI Data Archive server

## 5.1 Pre-installation Checks

It is critical that you perform the pre-installation checks. If you neglect this step in some cases you will get an error, and in others, the installation will stop.

- **Log on as Administrator** (or with administrative privileges). The installer must be either the administrator or member of the local Administrators group. In addition, the account must have write permission to MS SQL Server. Validate that the user has the correct permissions.

- ***Always check the PI Data Archive Server operating system clock when installing any PI System.*** Ensure the clock on each machine has the correct time and it is in the correct time zone. In your work environment, all clocks should be synchronised from a network time source. Changing the clock after installation will cause problems.
- **Update Windows**. A properly updated Windows Operating System will have the required prerequisites. If you require any prerequisite components, you will need to install them before the installation proceeds.
- **Install Microsoft SQL Server.** The version you use is your choice. You now should know the pros and cons of each of the available offerings.
- **Obtain your PI Server License File.** Download the file from the OSIsoft Customer Portal https://my.osisoft.com; an explanation is forthcoming.

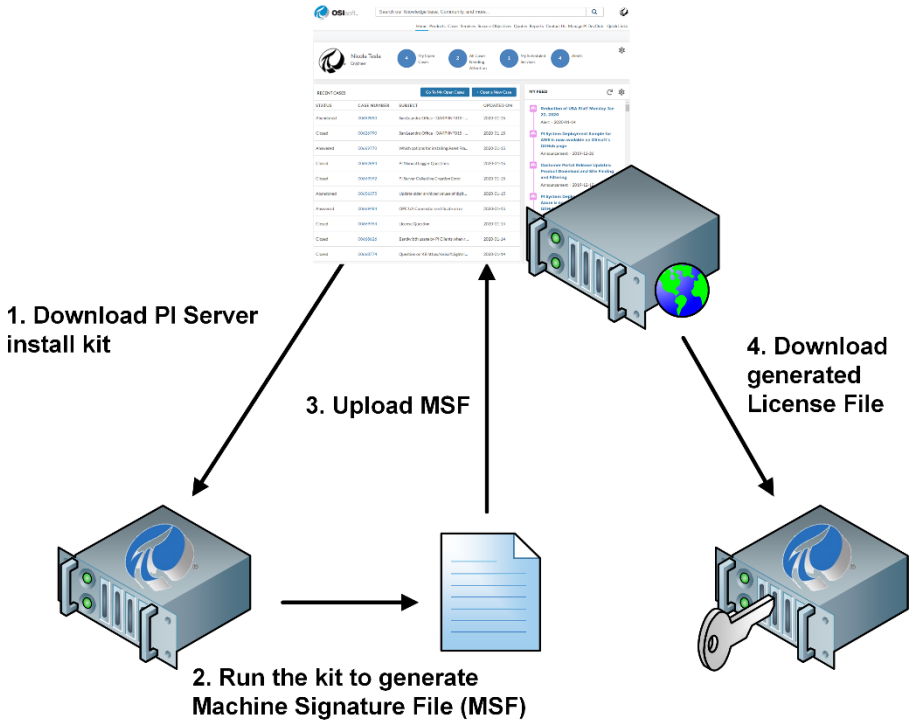
## 5.2 Supported Operating Systems

Starting with PI Data Archive 2016, only 64-bit version is available and requires a 64-bit Operating System. For production systems, PI Data Archive version 2018 SP3 or higher can be deployed on the following Microsoft Windows Server operating systems, in decreasing order of recommendation:

- Windows Server 2019 – all editions; in both Full and Core installations
- Windows Server 2016 – all editions; in both Full and Core installations
- Windows Server 2012 R2 – all editions; in both Full and Core installations
- Windows Server 2012 – all editions; in both Full and Core installations

## 5.3 License File Activation

A **License File** must be generated before the PI Data Archive server is installed. The OSIsoft Customer Portal allows you to generate your site-specific PI Server license file. This license file controls which applications can run on the PI Data Archive Server and displays running parameters, such as the point count limit.


When the license file is generated, view the PI Server Manifest to verify the server details.

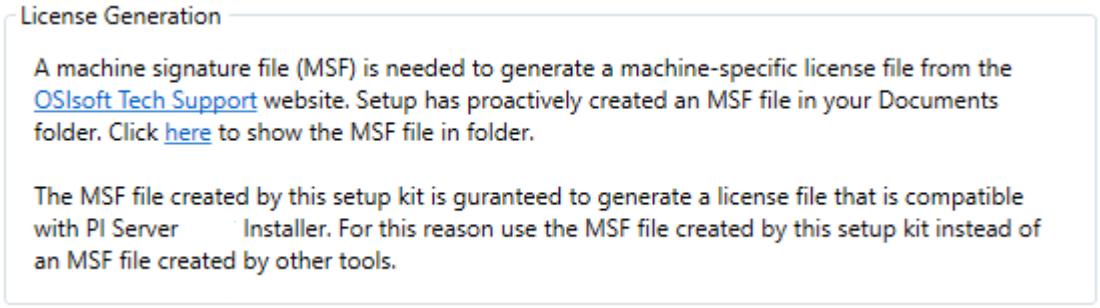
In order to generate license file, a **Machine Signature File (MSF)** is required. The PI Server install kit is capable to generate the MSF by itself, you must copy the install kit to a local disk on the PI Data Archive Server computer and then run the kit.

The kit will generate the MSF file to Documents folder of user running it.

i.e. *C:\Users\<username>\Documents*

If the PI Data Archive Server is on a virtual machine (VM), run the utility on the VM. If you generate the MSF on the wrong computer (on your laptop, for example) then the license activation file will match the laptop computer. If you install PI Data Archive Server on a different computer or VM, the server will not run as expected. The license file must be present during the installation. It can be on a flash drive, CD, or any media that can be read by PI Data Archive

Server during installation. The setup program copies the license file to the %PISERVER%\dat directory during installation; the original file will no longer be used.




Consult the *License PI Data Archive* section of the *PI Server Installation and Upgrade Guide* for full details.

## 5.4 Gathering Installation information.

The following information is requested during the installation:

- Location of the PI server license file – ask your instructor.

Normally, the following installation locations apply:

- PI AF files (.exe, .dll, SQL scripts) are installed in *%PIHOME64%\PIPC\AF (i.e. ..\Program Files\PIPC\AF)*
- PI Data Archive binary files (.exe) are installed in *%PISERVER%\bin (i.e. ..\Program Files\PI)*
- Archives are installed on the largest drive
  - e.g. if the C drive is the largest *C:\Program Files\PI\arc*
  - if D: is the largest drive *D:\Program Files\PI\arc*
- Event Queue files are installed on the 2nd largest drive
  - If C: is 2nd largest then *C:\Program Files\PI\queue*
  - If E: is 2nd largest then *E:\PI\Queue*

The defaults are:

- 64-bit PI applications - \Program Files\PIPC → %PIHOME64%
- 32-bit PI applications- \Program Files (x86)\PIPC → %PIHOME%
- PI Data Archive path - \Program Files\PI → %PISERVER%
- Archives path - \PI\arc
- Future Archives path - \PI\arc\future
- Event Queues - \PI\queue
- For the default archive, size see below.

### 5.4.1 Archive Sizing

Archives are sized with at least 2KB for each point in the system. If your PI Data Archive will have 5,000 points or less, then you can safely use the default value (currently 256MB). The default archive size is 3 KB x the total number of points rounded to the nearest power of 2 with a minimum of 512MB and maximum of 10GB, as per the following:

| Licensed Point Count | Archive Size (MB) |
|---|---|
| 0 to 87,381 | 256 (2^8) |
| 87,382 to 174,762 | 512 (2^9) |
| 174,763 to 349,525 | 1,024 (2^10) |
| 349,526 to 699,050 | 2,048 (2^11) |
| 699,051 to 1,398,101 | 4,096 (2^12) |
| 1,398,102 to 2,796,202 | 8,192 (2^13) |
| 2,796,203 or greater | 10,240 (capped) |

Select a size so that at least 2 archive files can fit in the Windows File System Cache (FSC). This is because mostly the PI Data Archive Server will write/read from the 2-3 most recent archive files. The FSC can use all the RAM on 64-bit systems. The following guidelines for memory apply:

| Physical Memory (MB) | Archive Size (MB) |
|---|---|
| 0 to 1,535 | 256 (2^8) |
| 1,536 to 3,071 | 512 (2^9) |
| 3,072 to 6,143 | 1,024 (2^10) |
| 6,144 to 12,287 | 2,048 (2^11) |
| 12,288 to 24,575 | 4,096 (2^12) |
| 24,576 to 30,719 | 8,192 (2^13) |
| 30,720 or greater | 10,240 (capped) |

## 5.5 Directed Activity – PI Server Installation Pre-requisites

In this part of the class, you will perform a learning activity to reinforce the concepts presented in this section.

**Activity Objectives**

- Understand the necessary information to install PI Server. In other words: Read the Manual! (RTM)

**Approach**

Answer the following questions as a group:

1. Name the supported MS SQL Server versions.

2. At what stage will the PI Server installation install the MS SQL server?

3. Can the PI AF SQL (PIFD) database be created manually?

4. Do end users connect to the MS SQL Server?

5. How can you verify if MS SQL Server is installed?

## 5.6 Directed Exercise – Install the PI Server

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

- Install the PI Server and related services and features on PISRV01

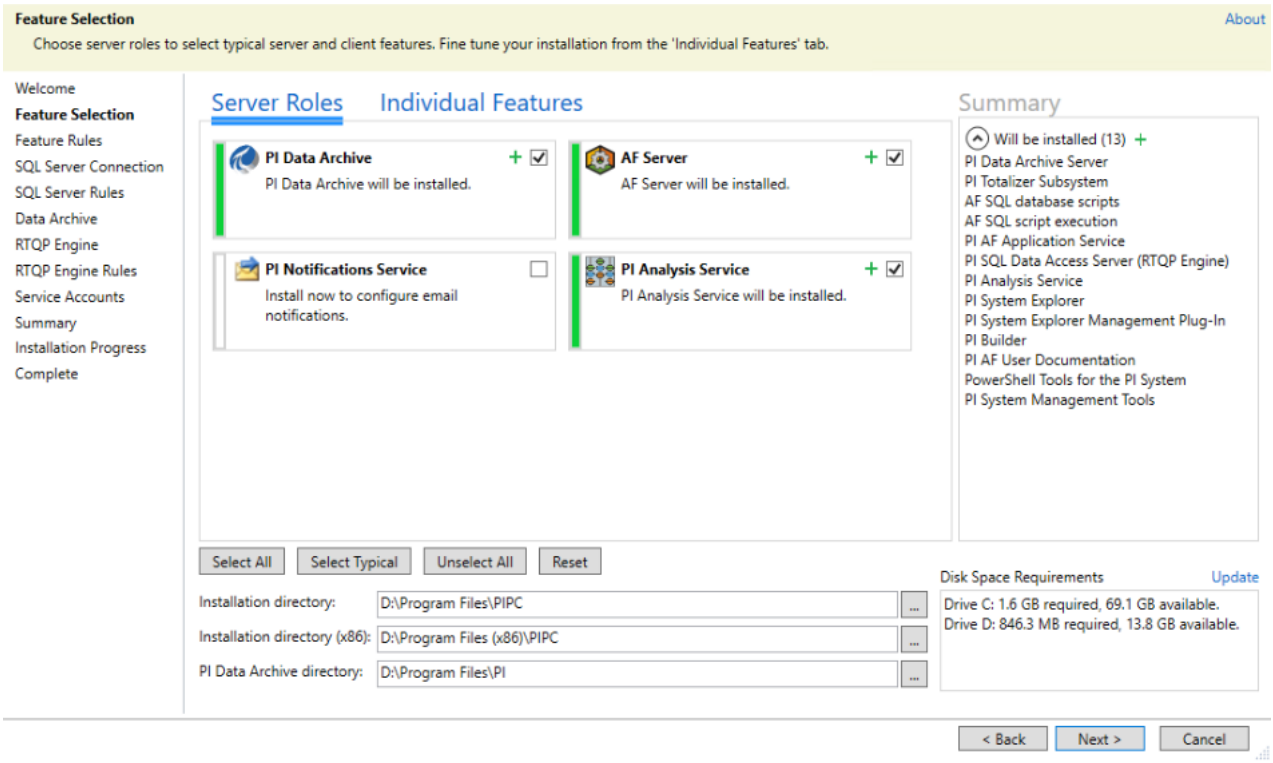
### Description

You are ready to begin the PI System installation.

You should have validated the prerequisites and Microsoft SQL Server, have the install kit and license file, and performed all the computer checks (clock, etc.) You did, didn't you?

### Approach

1. In the installation folder, you will find the PI Server installation kit.
2. Right-click and **"Run as Administrator"**. After the Welcome screen, where you can deselect the participation in PI System Customer Experience Improvement Program there is a feature selection screen where we check the components we would like to install. Select all except PI Notifications Service.



3. Select the installation directories for 64-bit and 32-bit PIPC folders and PI Data Archive Directory. In our case it is:
   - **D:\Program Files\PIPC**
   - **D:\Program Files (x86)\PIPC**
   - **D:\Program Files\PI**

**Important Note:** Since unified installation kit PI Server 2018, the fresh installation of PI Data Archive by default no longer installs these subsystems:

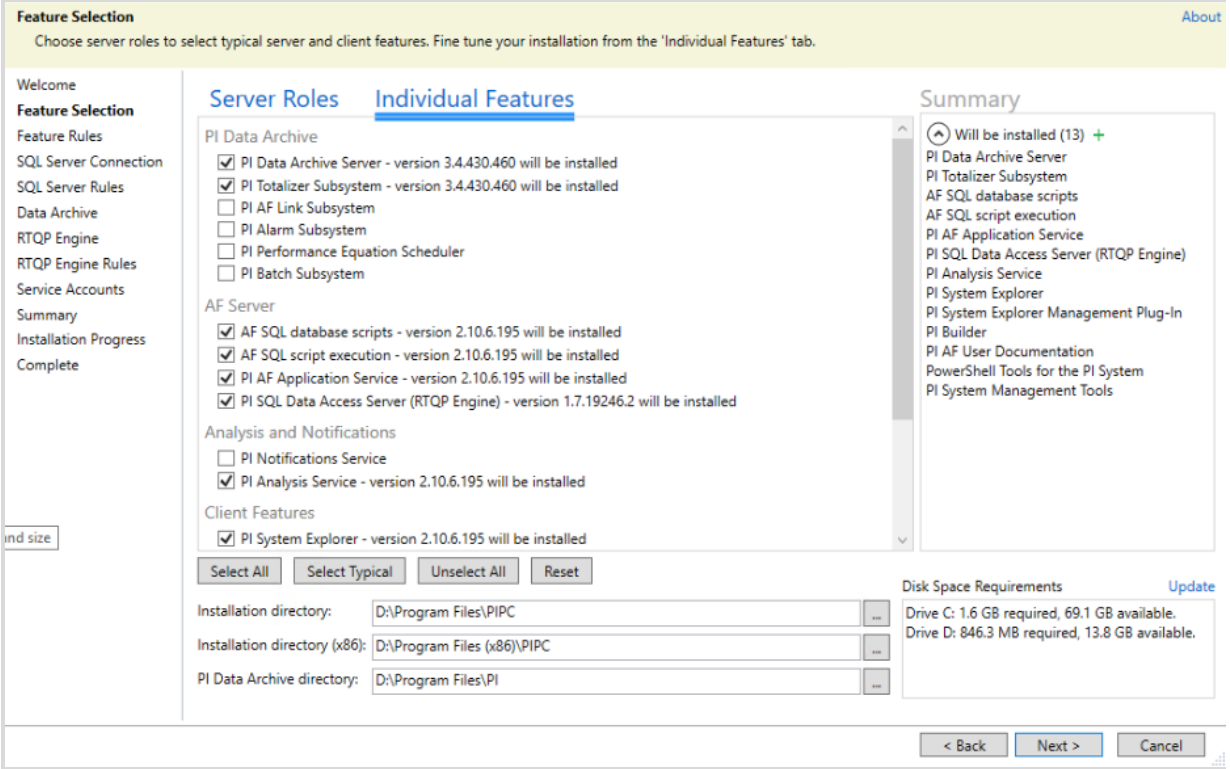
*- PI AF Link Subsystem*

*- PI Alarm Subsystem*

*- PI Performance Equation Scheduler*

*- PI Batch Subsystem*

If you wish to install those components, you switch to Individual Features section and tick the box. For upgrade from previous versions of PI Data Archive those components remain.



Since PI Server 2018 SP3, these PI Interfaces are no longer part of the installation kit:

*- PI Interface for Performance Monitor*

*- PI Interface for Ping*

*- PI Interface for SNMP*

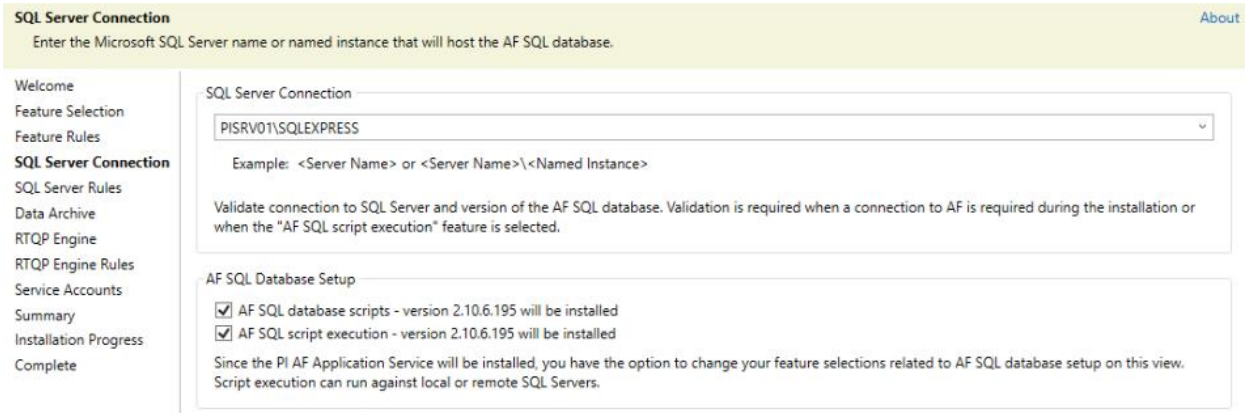
*- PI Interface for TCP Response*

*- PI Interface for Random Data Simulator Data*

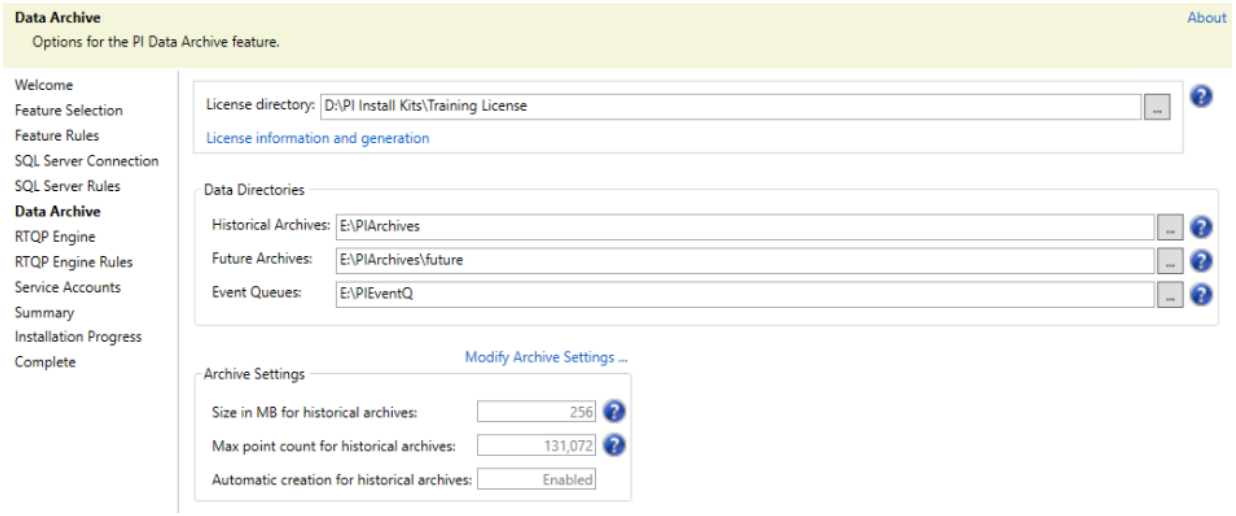
*- PI Interface for RampSoak Simulator Data*

Removing the PI Interfaces for Random and RampSoak Simulator Data from PI Server installation kit, eliminates default PI Points (*SINUSOID, SINUSOIDU, CDT158, CDM158, CDEP158, BA:LEVEL.1, BA:TEMP.1, BA:CONC.1, BA:ACTIVE.1* and *BA:PHASE.1*) from being installed, therefore **NO PI POINTS ARE PRESENT** on PI Data Archive after the installation.

4. Select the SQL server provided. In our case **PISRV01\SQLEXPRESS.** Keep the checks that AF SQL Database scripts will be installed and executed. Ensure the MS SQL Server service is started before installation.



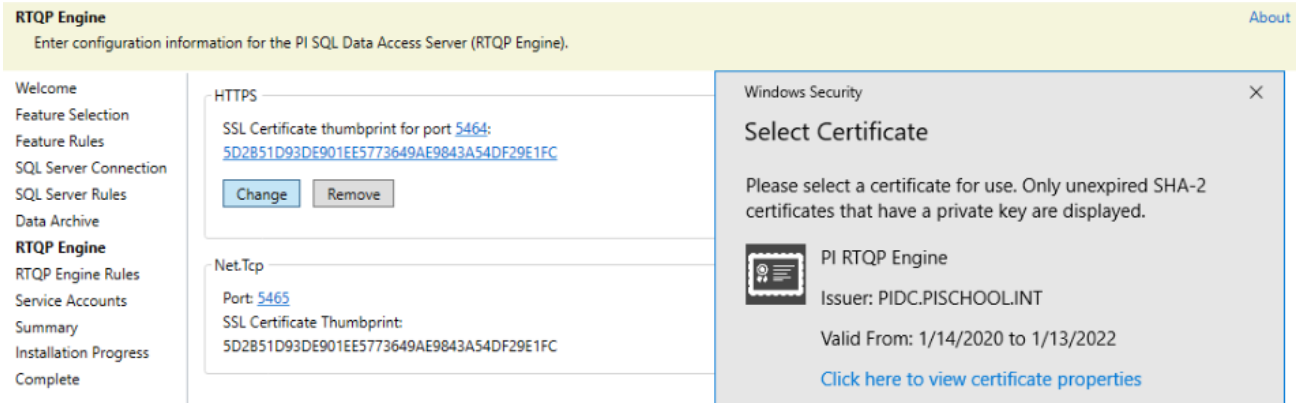

5. For PI Data Archive section select the License Directory and Data Directories



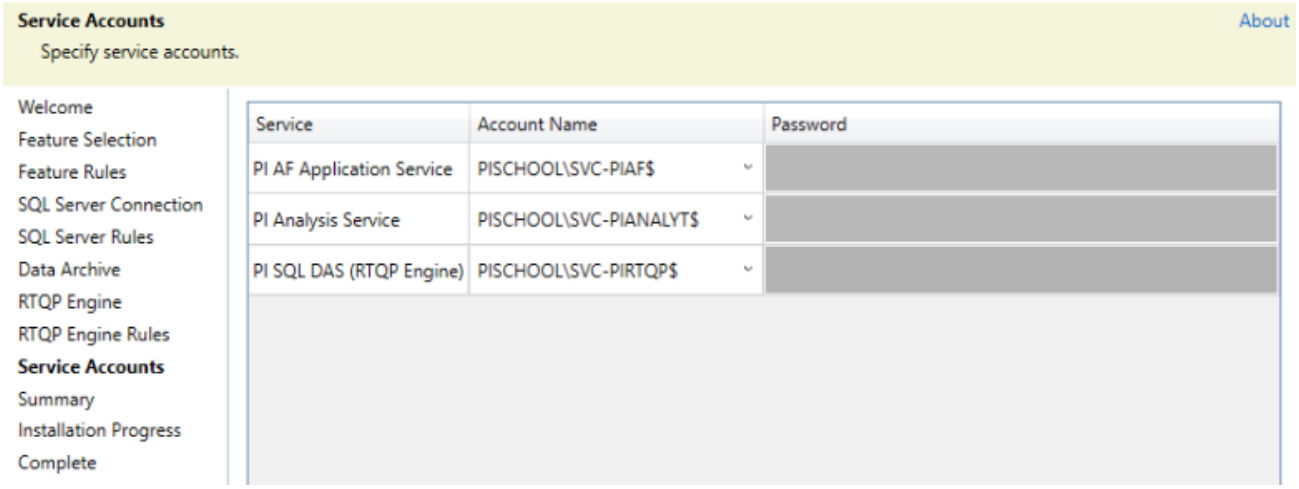
- License Directory to **D:\PI Install Kits\Training License**
- Historical Archives to **E:\PIArchives**
- Future Archive to **E:\PIArchives\future**
- Event Queues to **E:\PIEventQ**

In Archive settings you can modify the size for historical archives to lesser size than precalculated size (viz. chapter Archive Sizing)

6. At the RTQP Engine page, keep the default port 5465. For SSL certificate select the only available certificate **PI RTQP Engine**




7. Select the gMSA accounts:
   - PI AF Application Service: **PISCHOOL\SVC-PIAF$**
   - PI Analysis Service: **PISCHOOL\SVC-PIANALYT$**
   - PI SQL DAS (RTQP Engine): **PISCHOOL\SVC-PIRTQP$**



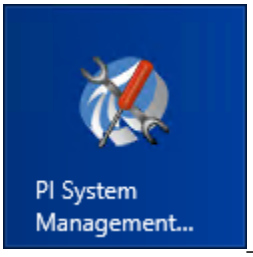

8. Click Next to the Summary page and start the installation.
9. If MS Excel page pops up, click on Install button to install PI Builder plug-in.

10. After the installation is complete manually stop the PI Data Archive by executing the script **%PISERVER%\adm\pisrvstop.bat**.
11. Open Services Manager and switch *NT Service\pinetmgr* account for gMSA **PISCHOOL\SVC-PIDA$** for PI Network Manager service
12. How do you start the PI Data Archive? _____

The installation created PI System folder in Windows Start Menu with a list of selected applications. Find out what they are used for…


_____


_____




_____


_____

## 5.7 Directed Activity – Validate the Installation

Now you are going to see if it works.

**Activity Objectives**

- Become familiar with the PI SMT console
- Understand PI Time Format

**Approach**

Prior the version PI Server 2018 SP3, when the PI Data Archive was installed, it came with a sample interfaces called the *PI Interface for Random* and *PI Interface for RampSoak Simulator Data*. The interfaces do just that. They produce configurable continuous streams of random data – sine waves, sawtooth or random noise. Those interfaces are used for trainings and simulations.

Since **PI Server 2018 SP3** they are not installed automatically, thus no default simulation PI Points.

We will create those PI Points a little bit later in the course.

In this activity connect to your newly installed PI Data Archive and determine whether all subsystems have started and there are no errors in PI Message Log.

Run the PISDKUtility.


Click on **Connections** in the **PI-SDK** branch.

The dialogue box invoked displays all PI Data Archive systems configured, in our case, one. Click in the checkbox next to your PI Data Archive to connect and validate your connection by validating:

- Connected user _____
- PI Version _____
- Operating system _____

**Note:** The first time the PI SDK is installed anywhere, a default PI Data Archive is needed. Therefore, even if you have never configured a PI Data Archive server, at least one server should appear in your **PI Connection Manager**.

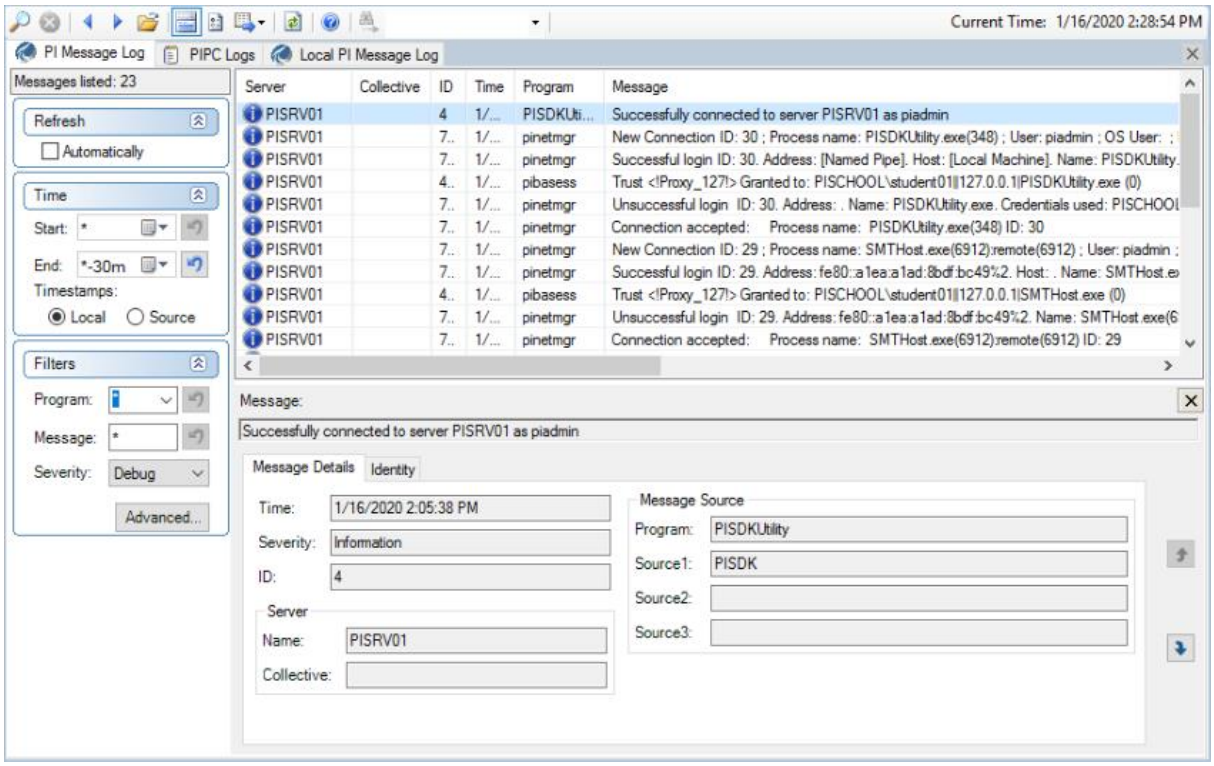
### 5.7.1 Validation using PI System Management Tools (PI SMT).

The PI System Management Tools (PI SMT) is the tool most used by PI System Administrators. It allows a person with the correct credentials to perform any action.

**Tip** Using the PI SMT on the same computer as the PI Data Archive server will most likely give you unlimited access. Use with caution.

Once connected there are several ways to determine the status of the Data Archive system.

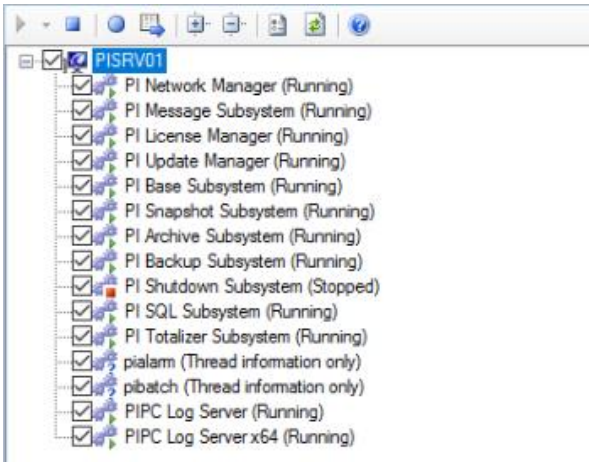
- Search the message log files for any errors (under **Operations → Message Logs**)



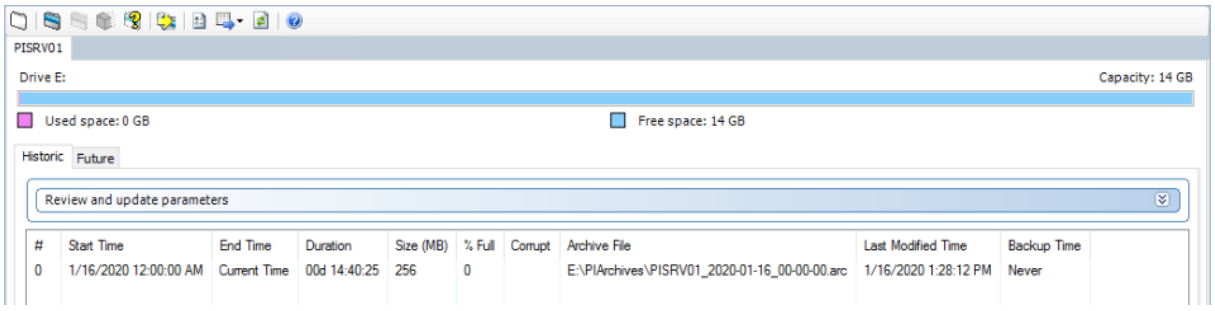

- View applications currently connected to the server in **Operations → Network Manager Statistics**

| Server | ID | PIPath | Name | PID | RegAppName | RegAppType | Identity | Trust | ProtocolVersion | ElapsedTime | BytesSent | ConType |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PISRV01 | 0 | D:\Program Files\PI\ | pimsgss | 5768 | pimsgss | PIService | | | 3.5 | 74,844 | 262,486 | Local connection |
| PISRV01 | 1 | D:\Program Files\PI\ | pilicmgr | 4212 | pilicmgr | PIService | | | 3.5 | 74,843 | 3,018,024 | Local connection |
| PISRV01 | 6 | D:\Program Files\PI\ | pisnapss | 2944 | pisnapss | PIService | | | 3.5 | 74,839 | 8,014,187 | Local connection |
| PISRV01 | 5 | D:\Program Files\PI\ | pibasess | 4848 | pibasess | PIService | | | 3.5 | 74,840 | 15,750,826 | Local connection |
| PISRV01 | 4 | D:\Program Files\PI\ | piupdmgr | 3992 | piupdmgr | PIService | | | 3.5 | 74,841 | 1,374,432 | Local connection |
| PISRV01 | 7 | D:\Program Files\PI\ | piarchss | 5592 | piarchss | PIService | | | 3.5 | 74,837 | 13,255,082 | Local connection |
| PISRV01 | 17 | D:\Program Files\PI\ | pisqlss | 6916 | pisqlss | PIService | | | 3.5 | 74,803 | 147,196 | Local connection |
| PISRV01 | 24 | D:\Program Files\PI\ | pilogsrv.exe | 7096 | PIPC Log Server | OSIAPIApp | | | 3.5 | 74,786 | 44,718 | Local connection |
| PISRV01 | 23 | D:\Program Files\PI\ | pilogsrv.exe | 4508 | PIPC Log Server | OSIAPIApp | | | 3.5 | 74,788 | 87,598 | Local connection |
| PISRV01 | 13 | D:\Program Files\PI\ | pibackup | 4116 | pibackup | PIService | | | 3.5 | 74,817 | 353,839 | Local connection |
| PISRV01 | 19 | D:\Program Files\PI\ | pitotal | 6400 | pitotal | ServerApp | | | 3.5 | 74,798 | 263,414 | Local connection |
| PISRV01 | 26 | D:\Program Files\PI\ | SMTHost.exe | 6912 | SMTHost | OSISDKApp | piadmin | !Proxy_127! | 3.5 | 74,202 | 368,465 | Local connection |
| PISRV01 | 29 | D:\Program Files\PI\ | SMTHost.exe(6912):remote | 6912 | SMTHost | OSISDKApp | piadmin | !Proxy_127! | 3.5 | 74,124 | 1,724,309 | Remote resolver |
| PISRV01 | 28 | D:\Program Files\PI\ | SMTHost.exe | 6912 | SMTHost | OSISDKApp | | | 3.5 | 74,124 | 44,506 | Local connection |
| PISRV01 | | D:\Program Files\PI\ | PINetMgr | | pinetmgr | PIService | | | 3.5 | | 44,709,082 | |

- Check if all PI Services are running (beside PI Shutdown Subsystem) in **Operation** → **PI Services** (or you can use Services Manager)




- Check if primary archive is listed in **Operation** → **Archives**



## 5.8 Overview of Management Tools for PI

PI System Management Tools (PI SMT) are tools used to administer the PI Data Archive servers from client connections. The PI System Manager Tools install kit is included with every PI Data Archive server and is available as a separate download. The PI System Management Tool kit includes the following programs. Running the main setup kit will automatically install and run all the individual setup kits.

- *PI System Management Tools (PI SMT), with many plug-ins*.
  - is a set of easy-to-use tools that allow you to perform all the basic PI Server administration tasks
- *PI Collective Manager*
  - This tool allows an administrator to create a new PI Server collective, look at the status of existing collectives and their member nodes, and edit the collective and member node properties
- *PI System Tray*
  - PI System Tray monitors the default PI Server (or PI Server collective) and the AF application service associated with the default AF server.
- *PowerShell Tools for the PI System*
  - PowerShell Tools for the PI System is a set of cmdlets for Windows PowerShell, which allows you to manage a PI System. This enables PI System administrators to create reusable scripts for commonly needed or bulk system management operations.
- *PI Builder*
  - Installed as part of AF Services, it is also included as part of the Data Archive installation kit

For further information on any of these tools access the HELP in each tool or visit http://livelibrary.osisoft.com

## 5.9 Time and the PI Data Archive

The PI Data Archive stores data in the form of events. Each event has a value and a timestamp that indicates the collection time of the value.

### 5.9.1 Daylight Savings Time

The PI Data Archive stores all values with a time that is converted to UNIX time i.e. PI System stores timestamps as the number of seconds expired since January 1, 1970 GMT. This means that each day of the year has exactly 24 hours. Any adjustments for time, such as time zone or Daylight-Saving Time (DST), are made by the local machine clock of the user looking at the data. Display sequencing and math operations are performed on the UTC basis. The displayed time is interpreted by looking at the time zone on the client or server and re-converting this data time into a local time.


PI servers, data collection computers, and client computers should all have their time zones and times set correctly and synchronized. As the PI clients and PI Data Archive know what time zone, they are in, so data can be viewed in either **Server Time** or **Client Time**. This is determined by a setting in the client tool.

**Note:** It is important that all the computers involved in collecting data (PI Data Archive, data collection computers, etc.) have their operating system clocks set correctly.

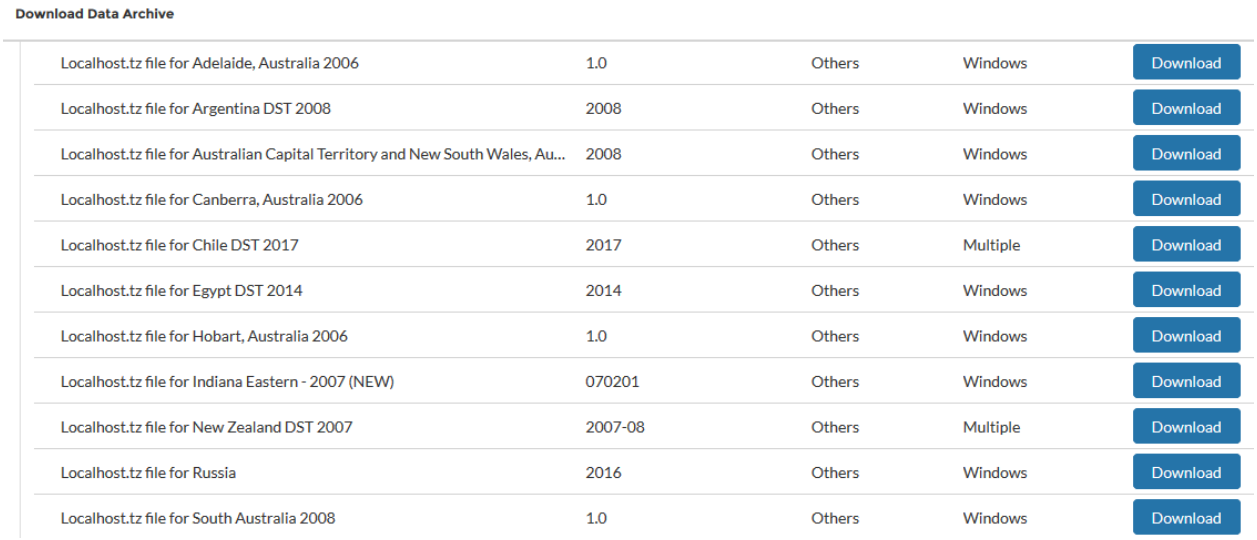
For most current PI Interfaces, events are sent to the server with UTC timestamps. All PI Connectors and PI Connector Relay are sending the events with UTC timestamp too. If the PI Interface time is *more than ten minutes ahead* of the PI Data Archive, the PI Data Archive cannot handle the time difference and discards the data; it is considered a future event.

Automatic DST changes will not cause a problem when all computers observe the same rules. That is, either all computers change their clocks twice a year at the same time or they do not.

**Note:** Situations where some computers change their clocks when others do not can cause data loss.

The PI Data Archive automatically adjusts data according to daylight savings time transitions. By default, only the current DST transitions are known. They are provided by the operating system. To make sure that past transitions between daylight savings times are correct, you need to update the localhost.tz file on your PI Data Archive.

To verify that the DST rules, run **pidiag -tz** command in the \PI\adm folder in Command Prompt to check the time zone and DST transitions table of your PI Data Archive. If you anticipate backfilling data into the archives from years in the past, you will need to get a replacement localhost.tz for your time zone file from the OSIsoft's Customer Portal site by going to PI Server downloads and expanding the Other section.



**Download Data Archive**

| | | | | |
|---|---|---|---|---|
| Localhost.tz file for Adelaide, Australia 2006 | 1.0 | Others | Windows | Download |
| Localhost.tz file for Argentina DST 2008 | 2008 | Others | Windows | Download |
| Localhost.tz file for Australian Capital Territory and New South Wales, Au... | 2008 | Others | Windows | Download |
| Localhost.tz file for Canberra, Australia 2006 | 1.0 | Others | Windows | Download |
| Localhost.tz file for Chile DST 2017 | 2017 | Others | Multiple | Download |
| Localhost.tz file for Egypt DST 2014 | 2014 | Others | Windows | Download |
| Localhost.tz file for Hobart, Australia 2006 | 1.0 | Others | Windows | Download |
| Localhost.tz file for Indiana Eastern - 2007 (NEW) | 070201 | Others | Windows | Download |
| Localhost.tz file for New Zealand DST 2007 | 2007-08 | Others | Multiple | Download |
| Localhost.tz file for Russia | 2016 | Others | Windows | Download |
| Localhost.tz file for South Australia 2008 | 1.0 | Others | Windows | Download |

**Note:** To check all the DST changes from year 1970 up to year 2038 run the command: ***pidiag –tz –sys -dump***. This will list all dates when DST occurs (if DST is applied).

# 6. PI Interfaces

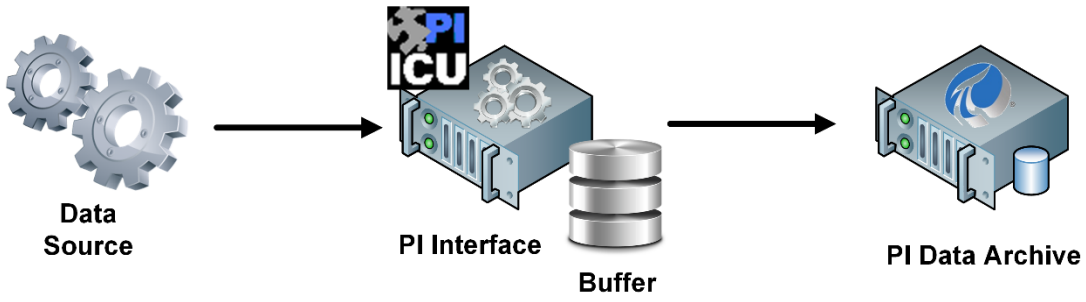
**Objectives:**

- Define a PI Interface
- Identify the basic components of the PI Interface installation
- Discuss the variety of architecture possibilities
- Describe how to connect PI Interfaces in a secure way
- Install and configure PI Interface
- Configure the basic PI Interface parameters
- Create and validate the default PI points for PI Random and RampSoak Interfaces

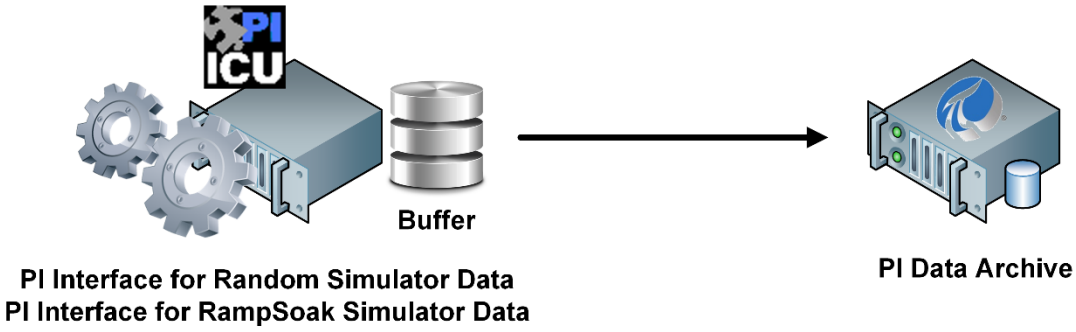
## 6.1 The PI Interface Defined

A PI Interface plays a critical role in the PI System. It performs the following tasks:

1. Connects to the data source
2. Timestamps the data (or ensures the data received is stamped at the source)
3. Formats data correctly
4. Sends data to PI Data Archive




PI Interface for Random and RampSoak Simulator Data do not connect to any data source, they are the data source themselves.



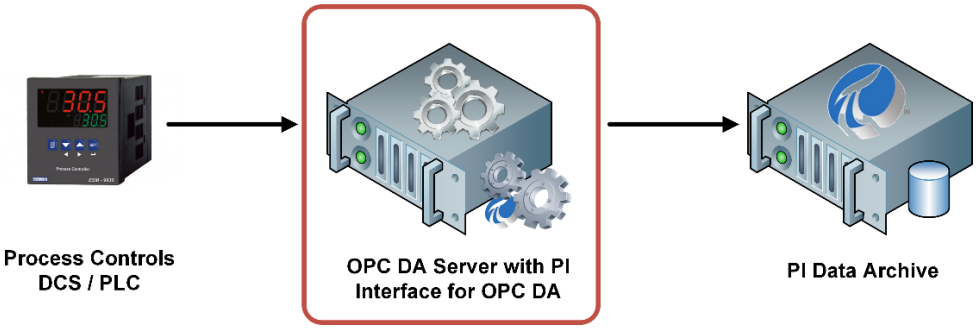
## 6.2 Directed Activity – Preparing to Install PI Interfaces

In this part of the class, you will perform a learning activity to explore the different concepts presented in this chapter or section.

**Activity Objectives**

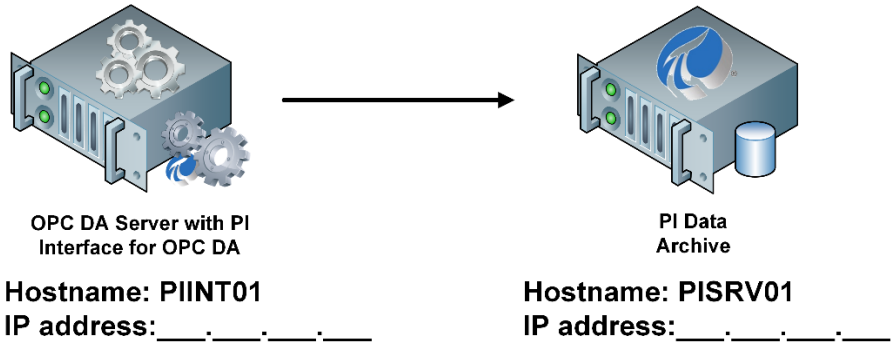
Normally before you begin PI Interface installation, you would identify your data source. Since we are in a training environment, we will use an OPC DA Simulator. It has been configured to generate a set of sample data from simulated pumps.

You will install the PI Interface for OPC DA on the same computer as the **data source** – in this case the OPC DA Server Simulator, along with simulation interfaces Random and RampSoak.




The OPC server we use is made available by the OPC Foundation as the download "OPC Data Access 3.00 Binaries". These files are available at http://www.opcfoundation.org/. Search for "OPC DA Sample Binaries." Consult the OPC Foundation web site for details if you wish to obtain a copy.

Fill in the following architecture diagram with the IP Addresses of your machines:



**Tip** It is a good idea to stick with the same configuration credentials when setting up the PI Interface and PI Buffering. If you use an IP Address in the PI Interface hostname then make sure you use the IP Address in the PI Buffering configuration, and vice versa.

## 6.3 Directed Activity – Validating communication between the PI Data Archive Server and the PI Interface

Install the PI ICU and PI API for Windows Integrated Security software and validate the connection.

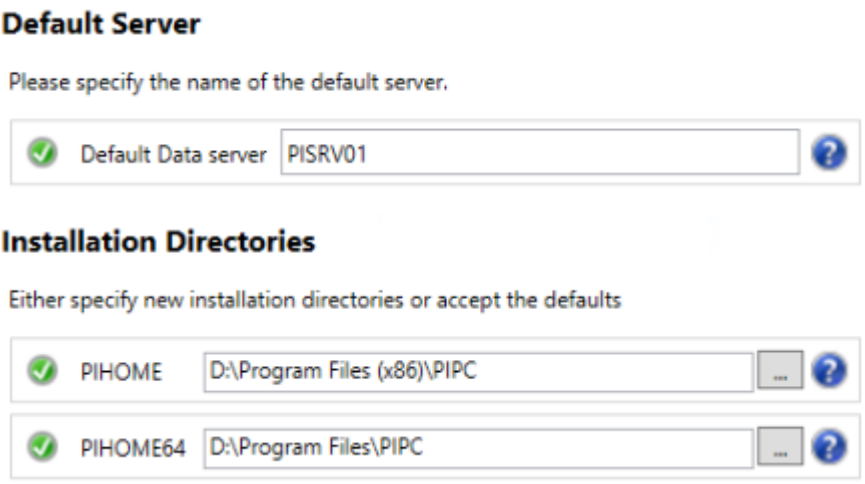
### Activity Objectives

- Validate the connection between PI Interface and PI Data Archive server.

### Approach

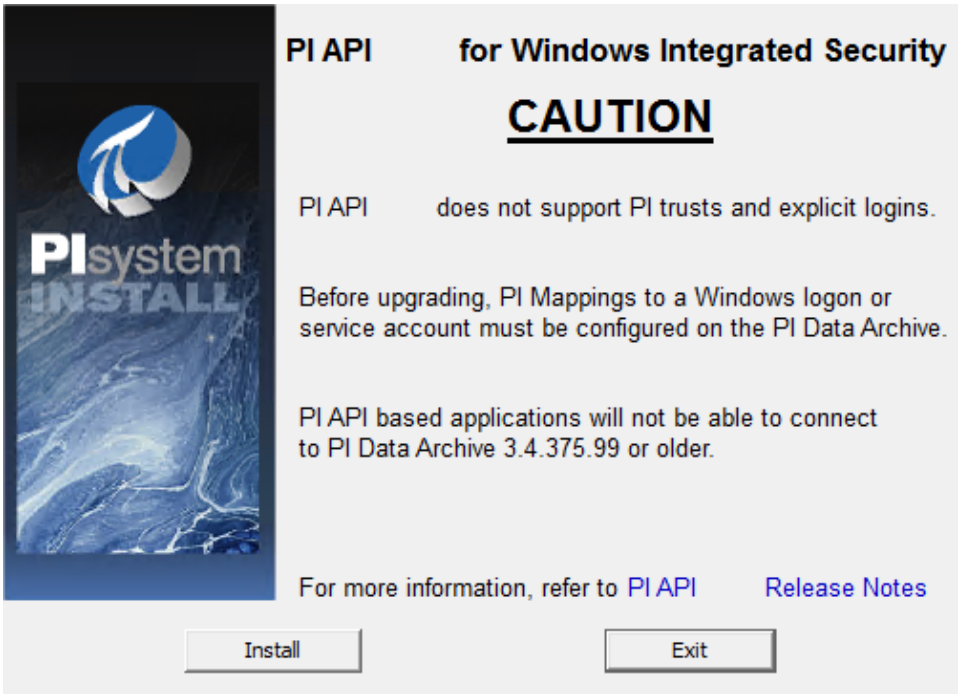
What does ICU stand for?

1. Install the PI ICU software on PIINT01. This will give you the required software to test the connection. You will be asked to fill the default PI Data Archive name and for the directories for the application system files.




2. Then install the **PI API for Windows Integrated Security**.

You will be prompted with a warning that PI API for WIS does not support PI Trusts and explicit logins.



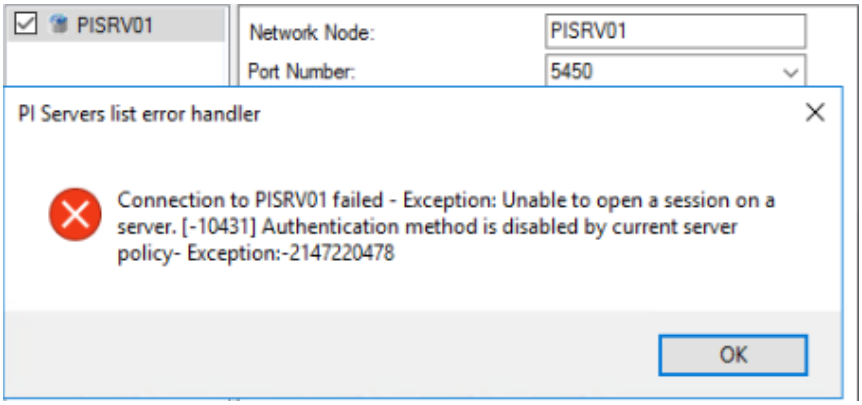

**Note:** PI API for Windows Integrated Security should replace the previous PI API versions that rely upon PI Trusts or explicit logins for authentications. With PI API for WIS Windows authentication becomes the only supported authentication model for PI API-based applications, such as PI Interfaces on the node where it was installed.

Windows Integrated Security is a more secure authentication model than PI Trusts for authenticating users!

3. After installation, verify the connection to the PI Data Archive using the command prompt **apisnap** utility (located at %PIHOME%\bin) that validates the **PI API** connection to the PI Data Archive.

```
D:\Program Files (x86)\PIPC\bin>apisnap PISRV01

APISNAP version 2.0.2.5
PI-API version 2.0.2.5
 Attempting connection to PISRV01
 Error connecting to PI Server node PISRV01: [-1] Unable to connect to server
D:\Program Files (x86)\PIPC\bin>_
```

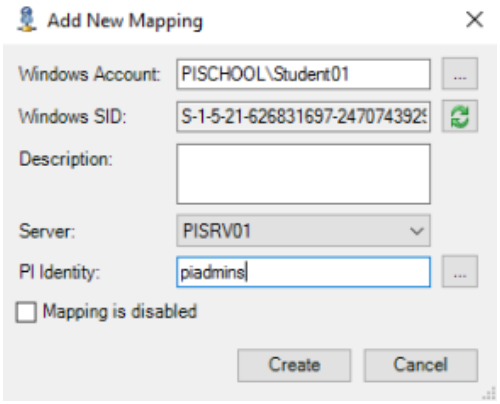
4. The PI Connection Manager validates the **PI SDK** connection to PI Data Archive. It is accessible with the **PISDKUtility** as seen previously.




**These tests fail. What do you think you will need, based on earlier discussions, to connect? – think security!**

Yes, it is because our user PISCHOOL\Student01 cannot authenticate against PI Data Archive as it is not defined. As we are PI Administrators, we assign our account administrative privileges.

5. On PISRV01, open PI SMT and navigate to **Security** → **Mappings & Trusts** plug-in and select Mappings tab.
6. Right-click on blank canvas and select New Mapping or click on icon.
7. Insert **PISCHOOL\Student01** as Windows Account. Click on to resolve Windows SID. This is a verification, that the account exists.
8. Insert **piadmins** as PI Identity and click on Create.

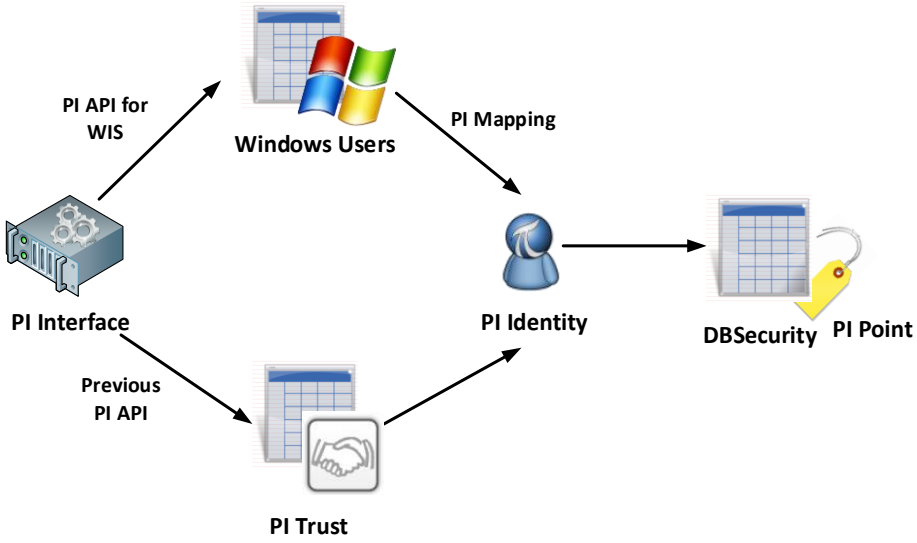



9. Run the apisnap utility again for PI API connection and PI Connection Manager for PI SDK connection to verify connection is possible.

## 6.4 PI Security for connecting PI Interfaces

For obvious reasons, the PI Interface needs to connect to the PI Data Archive. With PI API for WIS, a PI Interface can authenticate using the Windows account the application service is running under if **PI Mapping** to a proper **PI Identity** exists. We will be talking about PI Mappings and PI Identities later in the Security chapter.

With previous PI API version, which is part of the PI ICU or PI Interface install kit, the application cannot log in so it must rely on a mechanism called a **PI Trust** to gain access. PI Trust simply assigns a connecting program to the PI Identity. The PI Data Archive checks all incoming connections for a PI Trust if not disabled.




A good tool to view connections to the PI System is the Network Manager Statistics plug in in the PI SMT.

Previous PI API versions used a protocol that only sends the IP Address and an encoded application name in its communication. PI Trusts should be based on those parameters.

The application name used in the connection for PI Interfaces is an encoded phrase that takes *the first four letters* of the interface name and the capital letter “E”. The best way to verify this is to start the interface and watch the denied connection in the PI Data Archive using the Message Log plug-in of the PI SMT.

For example, the **PI Interface for Random Simulator Data** will use the phrase **RandE**.

PI API for Windows Integrated Security is using the *Windows authentication protocol*.

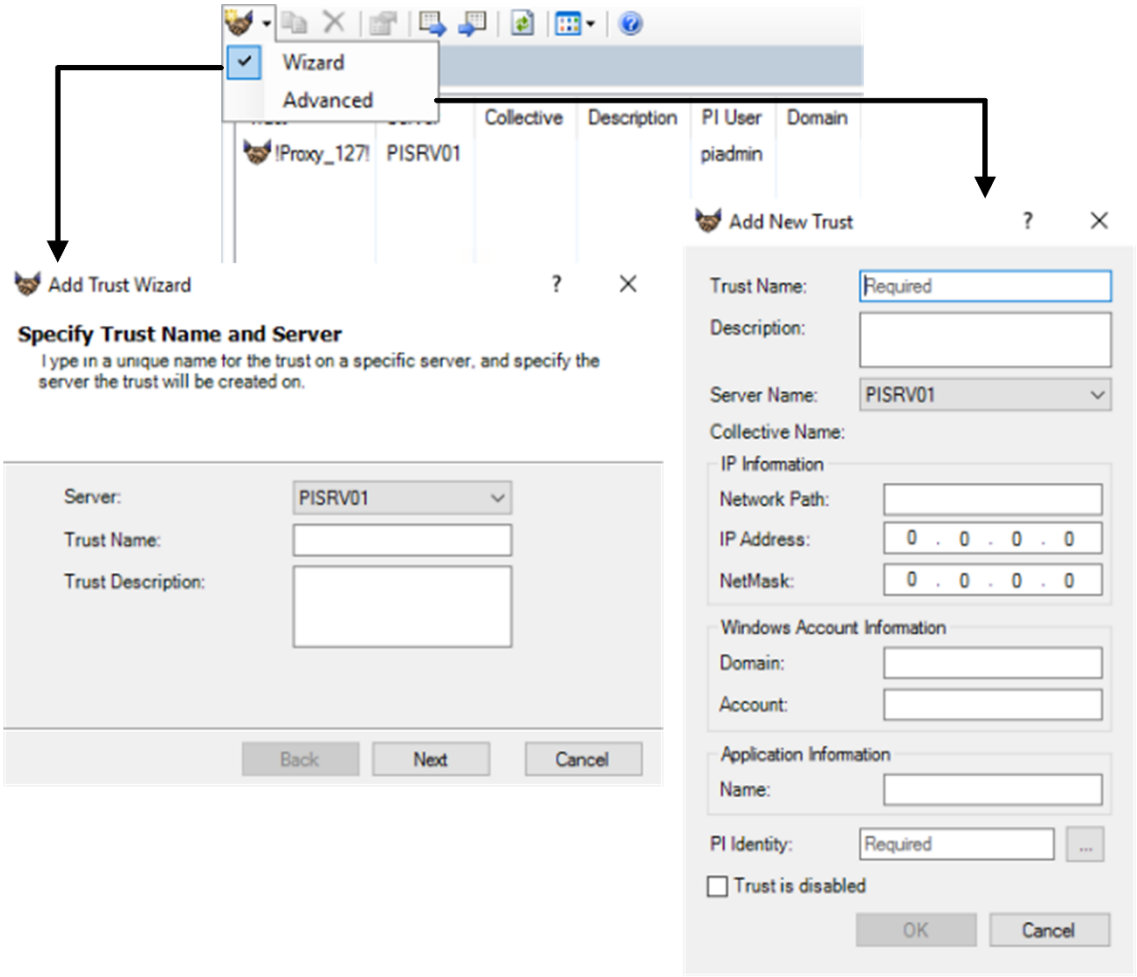
## 6.5 About PI Trusts

To create PI Trusts, use the PI SMT Mappings and Trusts plug in.

You can use the Wizard or the Advanced selection to create a PI Trust – they perform the exact same function.

**Tip**

Trusts should follow the OSIsoft Field Service Technical Standard **"2+ convention".** That is, a PI Trust to the PI Interface based on IP Address ***AND*** the executable name. You want to create Trusts SPECIFIC to the executable that is connecting to PI. It is not safe practice to map trusts to the **piadmin** user.




Some may find the Wizard confusing because understanding when to use a "PI API Application" or "PI SDK application on a Windows NT based OS" might not be intuitive.

Select Advanced instead of Wizard. This presents all the options in one dialog box.

## 6.6 Directed Activity – IP Addressing in Trusts

In this part of the class, you will perform a learning activity to explore the different concepts presented in this chapter or section.

### Activity Objectives

- Explain IP Address Filter for a PI Trust

### Approach

In the PI Trust mechanism, if you use IP Address you must specify a Netmask. However, the relationship between **IP Address** and **NetMask** in the PI Trust database is the same as the relationship between network destination and net mask in a TCP/IP routing table. Any valid subnet and IP address combination is supported. If you use this mechanism to allow access to all addresses in a subnet, you must set the bits corresponding to your subnet to zero.

| Row | Trust IP Address | Trust NetMask | Machine IP Address | Result of AND | Match |
|---|---|---|---|---|---|
| 1 | None (0.0.0.0) | None (0.0.0.0) | 192.168.168.121 | 0.0.0.0 | Yes |
| 2 | 192.168.168.0 | 255.255.255.0 | 192.168.168.121 | 192.168.168.0 | Yes |
| 3 | 192.168.168.0 | 255.255.255.0 | 192.168.175.004 | 192.168.175.0 | No |
| 4 | 192.168.168.22 | 255.255.255.255 | 192.168.168.22 | 192.168.168.22 | Yes |
| 5 | 192.168.168.22 | 255.255.255.255 | 192.168.168.20 | 192.168.168.20 | No |

Answer the following questions:

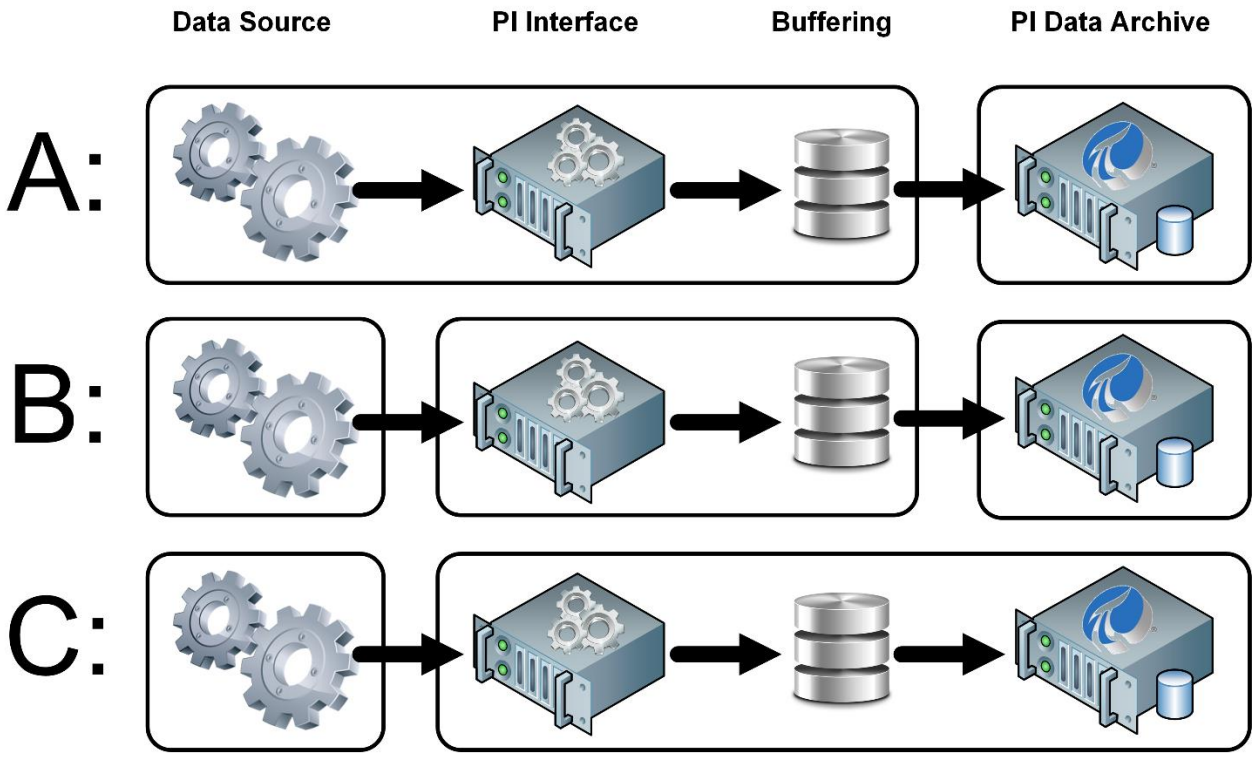
1. What computers would the following combination apply to:
   192.168.1.0 / 255.255.255.0

2. How would you construct a trust specifically for the machine with IP Address 192.168.10.54?

## 6.7 PI Interface Installation Considerations

### 6.7.1 Where do you install the PI Interface?

Several architectures are possible with PI Interfaces. Examples are shown below:




- **Architecture A**: The Data Source and PI Interface are installed on the same machine with stand-alone PI Data Archive.
- **Architecture B**: The Data Source, PI Interface and PI Data Archive are all installed on separate machines.
- **Architecture C**: PI Interface is installed on the PI Data Archive server

Having seen the above recommendations, what would you say was the least difficult configuration?

As a group, discuss the advantages, disadvantages and example application for each architecture:

| Architecture | Advantages | Disadvantages | Example application |
|---|---|---|---|
| A | | | |
| B | | | |
| C | | | |

### 6.7.2 Data Collection Node Clock

Before you install and configure the PI Interface, make sure that the clocks on the Data Collection machine and PI Data Archive machine are relatively close. They do not have to be exact.

Tip

PI Interface that is more than 30 minutes “off” the server clock will not run, and data for Real-time data points with a timestamp more than 10 minutes into the future will be discarded!

### 6.7.3 DST Rules

It does not matter so much that your computers obey DST, if the servers and your PI Interfaces are configured the same way and observe the same rules!

## 6.8 Directed Activity – PI Interfaces - What is installed?

Your instructor will have directions.

**Activity Objectives**

- Install the PI Interface for OPC DA, PI Interface for Random and PI Interface for RampSoak Simulator Data on PIINT01 and identify the PI Interface components.

**Approach**

Almost every PI Interface has at least four components that are installed on the data collection computer (*many PI Interfaces have additional files – refer to the manual for details*).

The location is always:

**%PIHOME%\Interfaces** – for 32-bit PI Interfaces

**%PIHOME64%\Interfaces** – for 64-bit PI Interfaces

The images below illustrate the basic components for the PI Interface for OPC DA. Write the functions next to the icons.


_____


_____


_____


_____

## 6.9 Directed Activity – PI Interface Parameters Dissected

In this part of the class, you will learn the common PI Interface switches

**Activity Objectives**

- Learn the PI Interface startup file switch syntax.

**Approach**

Navigate to the unconfigured startup batch file "OPCInt.bat_new" for the PI Interface for OPC DA. Edit the file to open it in Notepad. The elements in the startup file are parameters or "switches."

Open PI Interface manual (any manual).

Answer the following questions concerning the startup parameters:

1. What does the **/HOST** parameter mean? What formats are acceptable?

1. What does the **/ID** parameter do?

2. What does the **/PS** parameter refer to?

3. What is the link between **/ID** and **/PS**?

4. What is the **/F** parameter used for? What formats are acceptable? What is an "offset" and why is it used?

Notes on scan frequencies.

| | |
|---|---|
| **/f=***SS.nn*<br>or<br>**/f=***SS.nn,ss.nn*<br>or<br>**/f=***HH:MM:SS.nn*<br>or<br>**/f=***HH:MM:SS.nn, hh:mm:ss.nn*<br><br>Required for reading scan-based inputs | The **/f** parameter defines the time period between scans in terms of hours (*HH*), minutes (*MM*), seconds (*SS*) and sub-seconds (*nn*). The scans can be scheduled to occur at discrete moments in time with an optional time offset specified in terms of hours (*hh*), minutes (*mm*), seconds (*ss*), and sub-seconds (*nn*). If *HH* and *MM* are omitted, then the time period that is specified is assumed to be in seconds.<br>Each instance of the **/f** parameter on the command-line defines a scan class for the interface. There is no limit to the number of scan classes that can be defined. The first occurrence of the **/f** parameter on the command-line defines the first scan class of the interface and so on. PI Points are associated with a particular scan class via the Location4 PI Point attribute.<br><br>Two scan classes are defined in the following example:<br>**/f=00:01:00,00:00:05 /f=00:00:07**<br>or, equivalently:<br>**/f=60,5 /f=7**<br>The first scan class has a scanning frequency of 1 minute with an offset of 5 seconds, and the second scan class has a scanning frequency of 7 seconds. When an offset is specified, the scans occur at discrete moments in time according to the formula:<br>scan times = (reference time) + n(frequency) + offset<br>where n is an integer and the reference time is midnight on the day that the interface was started. In the above example, frequency is 60 seconds and offset is 5 seconds for the first scan class. This means that if the interface was started at 05:06:06, the first scan would be at **05:07:05**, the second scan would be at **05:08:05**, and so on. Since no offset is specified for the second scan class, the absolute scan times are undefined.<br>Wall Clock Scheduling<br>Scan classes that strictly adhere to wall clock scheduling are possible. This feature is available for interfaces that run on Windows and/or UNIX. For example, **/f=24:00:00,08:00:00** corresponds to 1 scan a day starting at 8 AM. However, after a Daylight Saving Time change, the scan would occur either at 7 AM or 9 AM, depending upon the direction of the time shift. To schedule a scan once a day at 8 AM (even across daylight saving time), use **/f=24:00:00,00:08:00,L.** The ,L at the end of the scan class tells UniInt to use the wall clock scheduling algorithm. |

## 6.10 Group Recap Questions

The following questions are intended to reinforce key information, or to discover a new insight. Your instructor may choose to have you try to answer the questions on your own or have the group answer them together aloud.

**Questions**

Which of the following scan classes *may* collect data at a different time to the others?

/f=5,10
/f=00:00:05, 00:03:05
/f=5,0
/f=00:00:05

Define scan classes for:

A scan every hour which scans at 10 minutes past each hour.

A scan every minute, on the minute.

A scan every 15 seconds with no preference on when the first scan is done.

A scan every 12 hours commencing at 7:00pm

What is the default scan time?

## 6.11 Directed Activity – PI Interface Documentation

Learn how to read a manual.

**Activity Objectives**

- Describe the reasons why the PI Interface Documentation is ***critical*** to the proper installation and configuration of PI Interface.

**Approach**

In computing, many applications can be installed without the documentation getting in your way. ***This does not apply to PI Interfaces.***

Navigate to and open the PI Interface for OPC DA Documentation. Go through the document sections and describe where key information can be found. Answer the following questions:

1. Will the interface run on a Windows 7 platform?

2. What PI Point types are supported?

3. Can this PI Interface write data back to the data source?

4. What is the maximum point count for this PI Interface?

5. What methods of Failover are supported?

6. Is vendor software required?

7. Is Disconnected start-up supported?

Tip

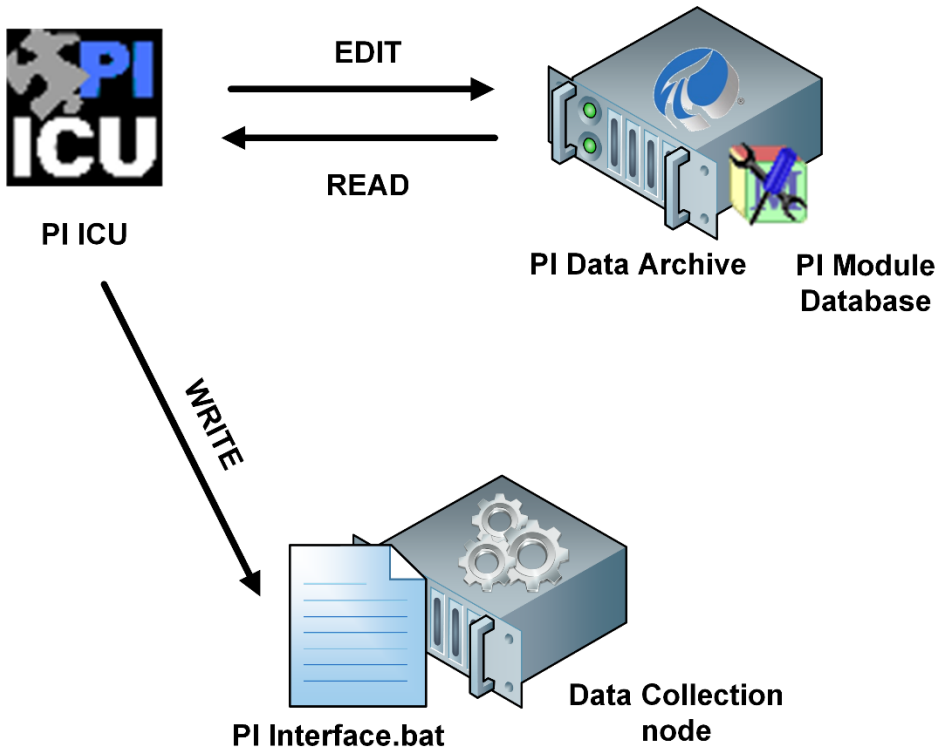
The PI Interface documentation will help you with the configuration of the PI Interface.

## 6.12 PI Interface Configuration Utility

PI ICU allows system managers to easily configure and maintain PI Interfaces. It automatically stores needed information on the server, and in the PI Interface start-up file (.bat). The PI ICU can only configure PI Interfaces that run on the *same* computer as the PI ICU.

The PI Interface Configuration Utility (PI ICU) is simply a GUI for editing the startup batch file.

One thing to keep in mind is that the PI ICU ***READS*** from the PI Module Database but also ***WRITES*** to the PI Interface startup batch file, as well as ***updating*** the MDB entry.




**Tip**

If you use the PI ICU to edit an interface startup file, do not edit the startup file manually afterwards – PI ICU will report MDB and start up batch file are not in sync.

This also means you will need to run the PI ICU under the user that has ***write*** access to the PI Module Database on PI Data Archive, which **piadmins** PI Identity by default has.

## 6.13 Exercise – PI Interface for OPC DA Configuration with PI ICU

Follow the instructor

**Exercise Objectives**

- Configure the PI Interface for OPC DA with PI ICU on PIINT01
- Start the interface as a Windows service.

**Problem Description**

You need to set up the PI Interface for OPC DA to collect data. An OPC Data Simulator (*OPCSample.OpcDa20Server.1*) is installed on your computer.

**Approach**

- Use the PI OPC Client Tool to find your OPC Server.
- Configure the PI Interface for OPC DA with the OPC server name and test the setup. The following walk through shows you what needs to be done.

***Before you start:***

Set the security for PI ICU and PI Interface on the PI Data Archive, use PI System Management Tools (PI SMT) to:

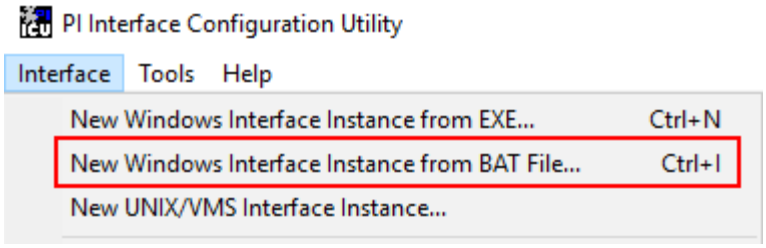
1. Create new PI Identity **PI Interfaces** and map it to the **PISCHOOL\SVC-PIINT$** gMSA.
2. Create new PI Identity **PI Buffer** and map it to the **PISCHOOL\SVC-PIBUFFER$** gMSA (Important for PI Buffering and PI Security chapters).

You cannot search for gMSA in PI SMT in order to map it to PI Identity. Service Accounts object is missing from the Object Types list. You must type the account name with "$" directly to Windows Account field.

3. Modify the Database Security in PI-SMT and add the PI Interfaces and PI Buffer identities to the following tables: **PIPOINT** with Read & Write access. (***Note****: The PI Security concept will be discussed in a separate chapter*)

***Use PI Interface Configuration Utility (PI ICU) to configure the PI Interface for OPC DA:***

4. Open PI ICU.
5. Open the PI Interface for OPC DA start-up file with the PI ICU. You are looking for the **OPCINT.BAT_NEW** file located at the *%PIHOME%\Interfaces\OPCInt* folder.



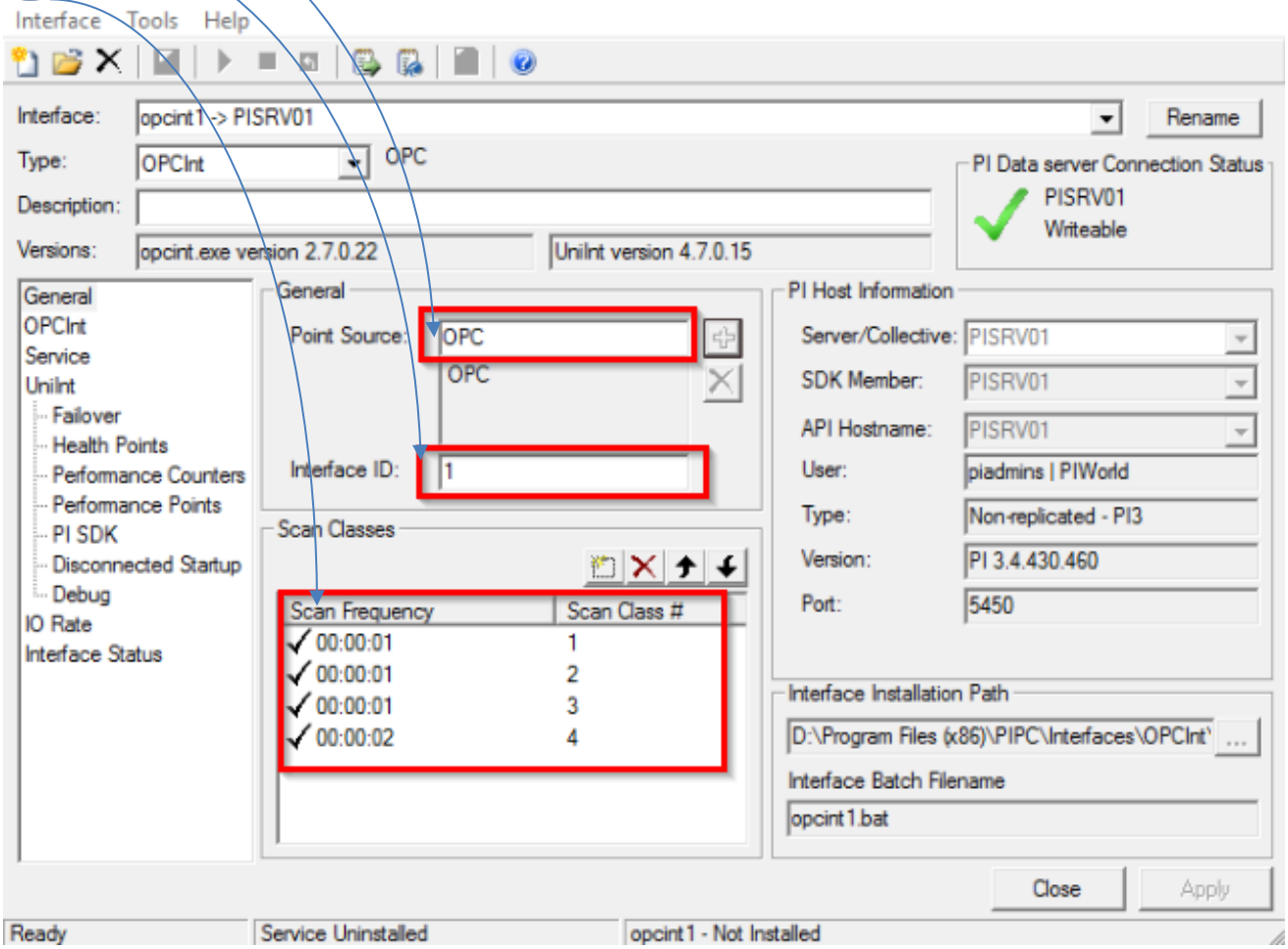

6. Select the PISRV01 from the drop-down list. Now you are ready to configure the PI Interface

### 6.13.1 The PI ICU General Section

/PS = Point Source (which interface?)

/ID = Interface ID (which instance of the interface?)

/F = Frequency/Scan Class (how often to read a value?)



### 6.13.2 The PI ICU Interface Specific Section

Specific to each interface. In this case OPCint.

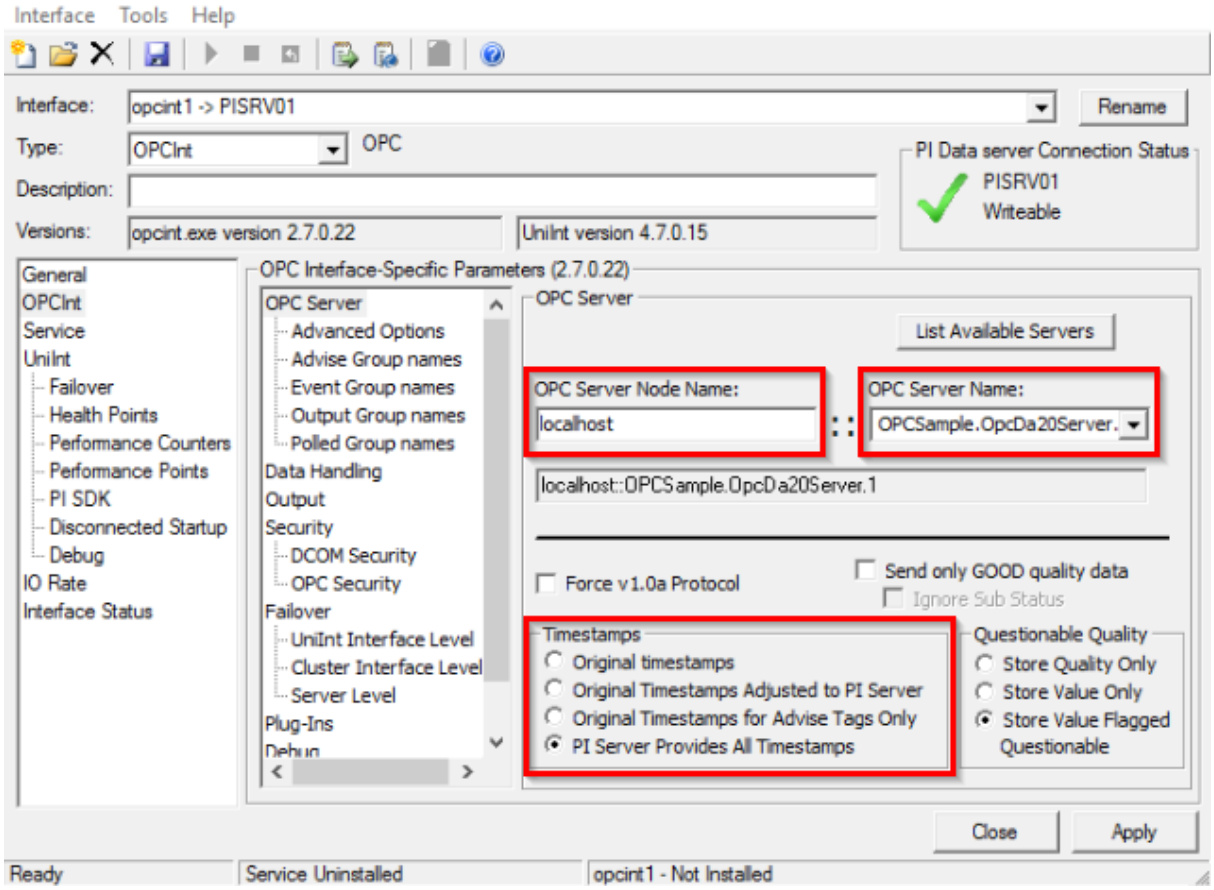
**/OPCSTOPSTAT** = IntfShut (write a value on shutdown)

**/SERVER**=<node>::<name>  (the machine and OPC Server name)

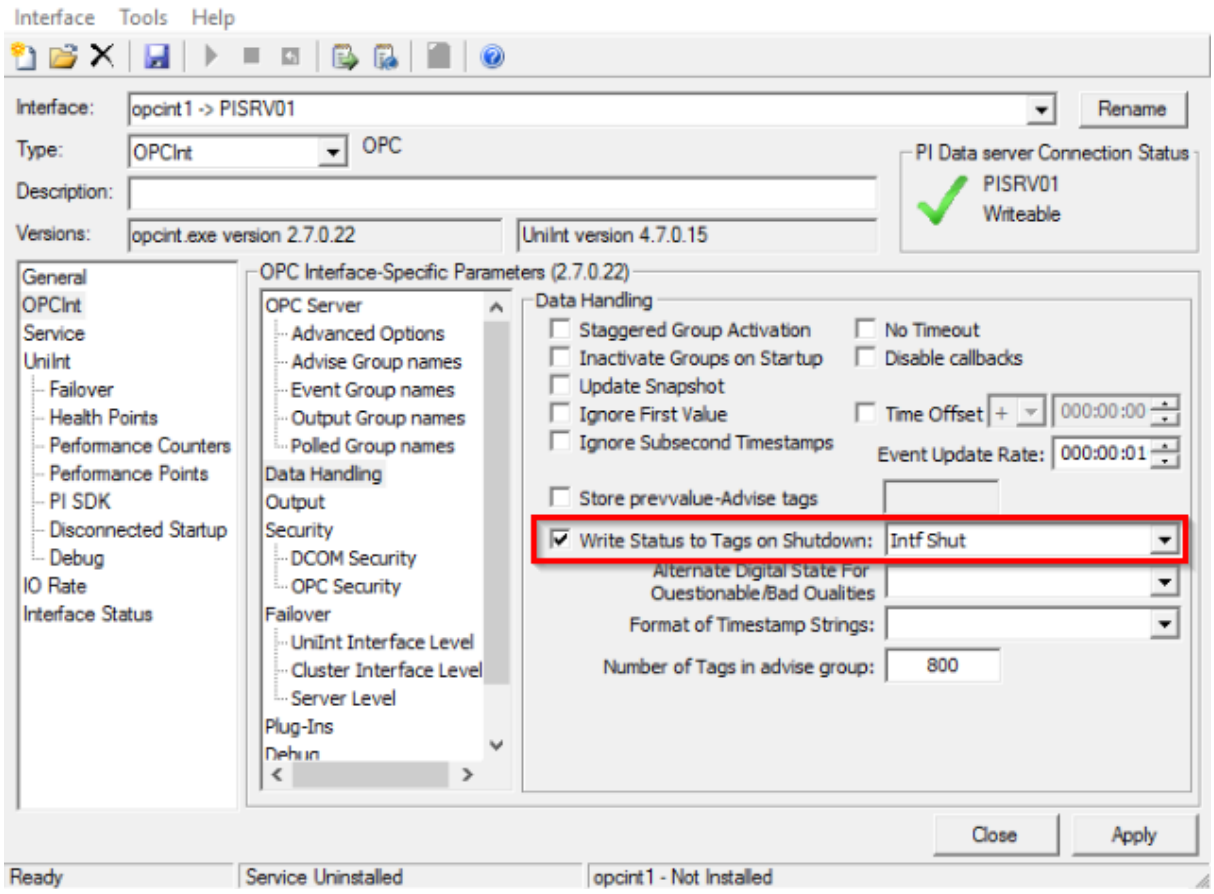
**/TS** = Timestamp Source

7. Set the OPC server name to the OPC DA server simulator ***OPCSample.OpcDa20Server.1***.

   **Hint**: Obtain the name of the OPC Server Name by using the PI OPC Client Tool. The PI OPC Client Tool is installed with the OPC Interface and is found in Start Menu under **PI System → PI OPC Client Tool.** Or click on List Available Servers button, however this may not work all the time based on DCOM settings.



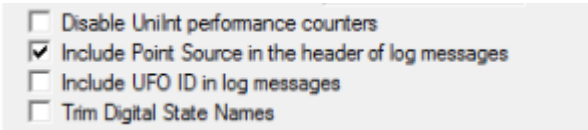

8. In OPCInt – Data Handling section check *Write Status to Tags on Shutdown*




### 6.13.3 The PI ICU UniInt Section

**UniInt** is short for Universal Interface. It is reusable code integrated in many of OSIsoft's interfaces to include generic functions such as establishing a connection to the PI Data Archive computer and monitoring the PI Point database for changes. This section is applicable to most, though not all interfaces. For most of the PI Interfaces the option Write Status to Tags on Shutdown is selected here, but for PI Interface for OPC DA it is ignored, and it is set in the OPCInt plug-in -> Data Handling section.

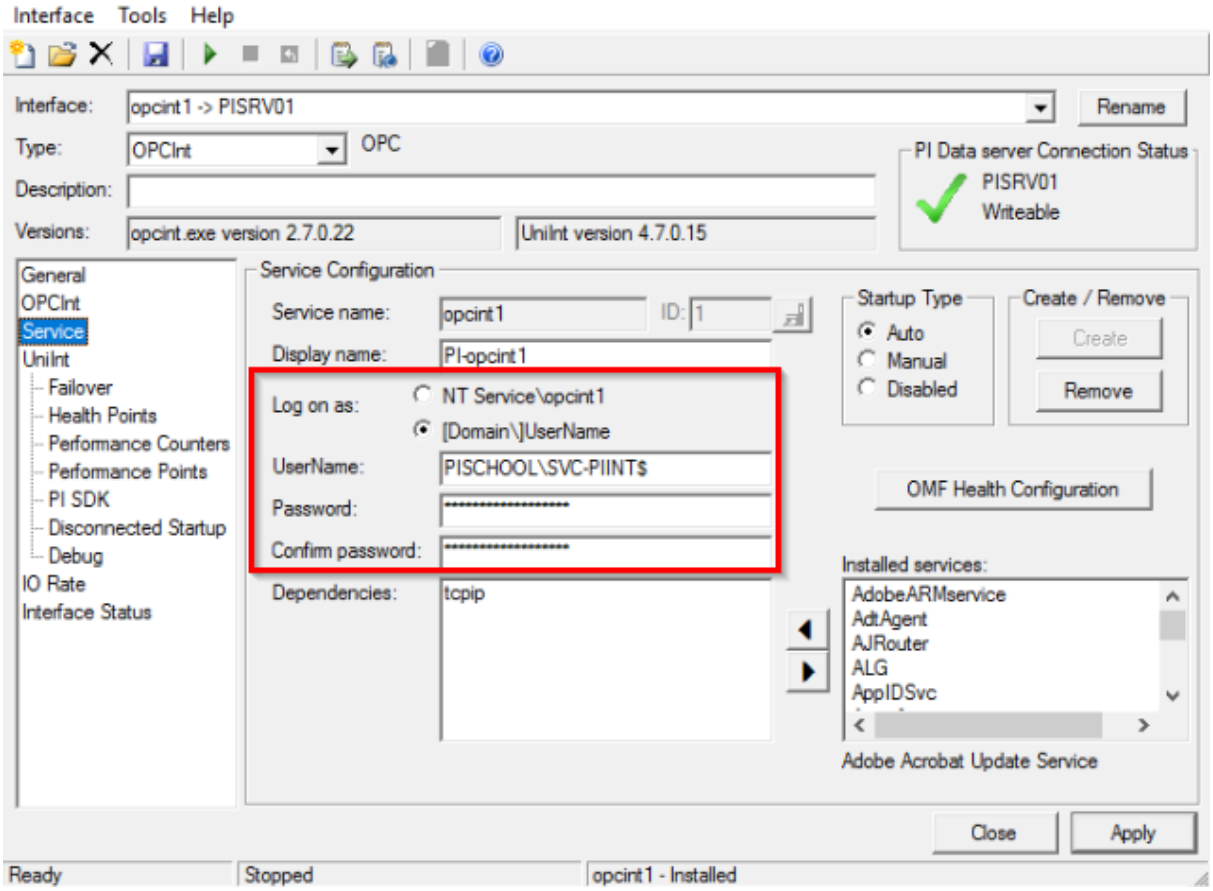
9. Recommended is also check the option *Include Point Source in the header of log messages.*




### 6.13.4 The PI ICU Service Section

10. Use PI ICU to create the interface Windows service. PI ICU is not capable to create PI Interface Windows Service under gMSA. Therefore, keep the default *NT Service\opcint1* account and hit **Create** button. Close PI ICU.

11. Open Services Manager find *PI-opcint1* service and switch the *NT Service\opcint1* account for **PISCHOOL\SVC-PIINT$** gMSA.
12. Open PI ICU again, select *opcint1* from drop-down list and select Service section. The gMSA is now applied.

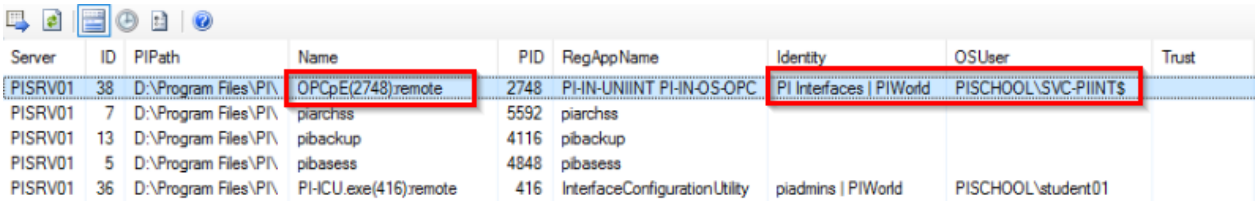



13. Start the PI Interface

There are two ways to start the interface:

- Interactive *(run the interface under your user account) Used only for testing!* In PI ICU menu *Interface → Start Interactive* or *CTRL+T*
- Non-Interactive *(start the Windows Service using services.msc from Start Menu)* or from within the PI ICU
- Start the interface interactively to confirm operation. Look for connections to both the PI Data Archive and the OPC server.
- Start the PI Interface as a Windows service and confirm proper start-up by using the PISDKUtility to monitor the message log.

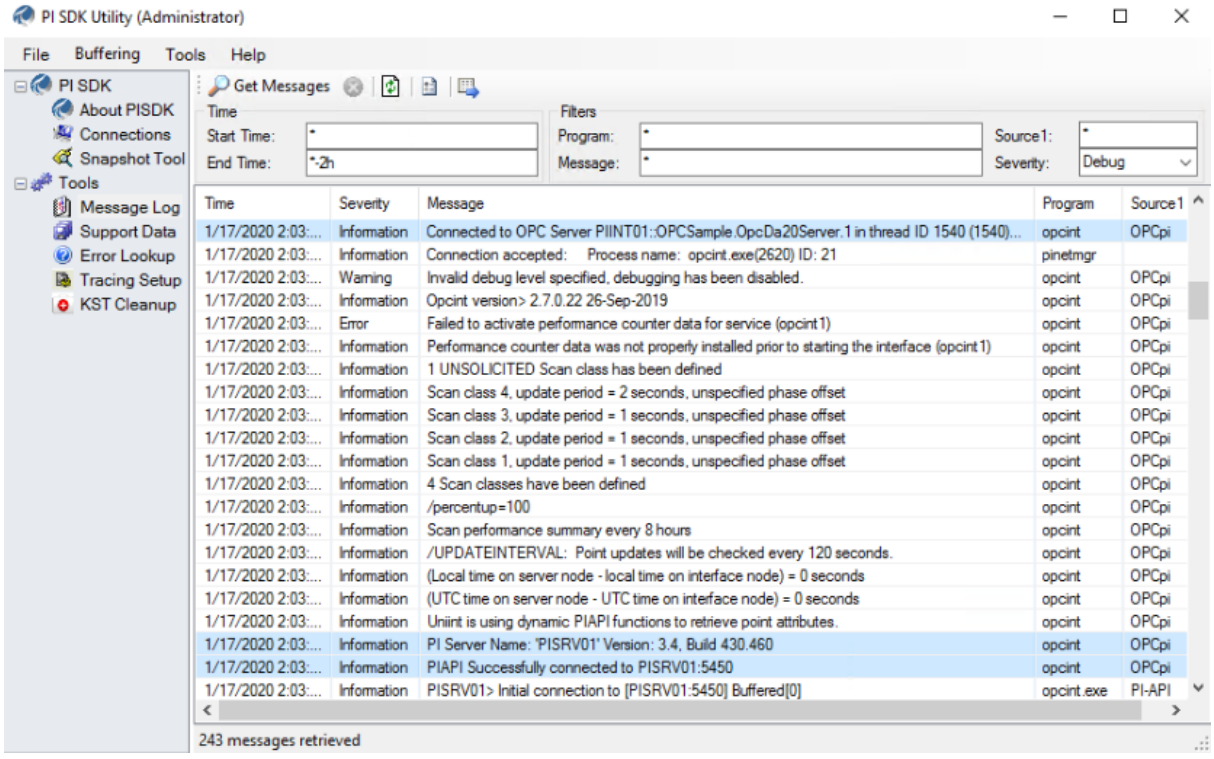
| | |
|---|---|
| **Tip** | Validate the interface can be started by the Windows service before you consider your configuration complete. Starting the interface interactively uses your account permissions (piadmins); however, the Windows service may not have the correct permissions and fail. |



| Server | ID | PIPath | Name | PID | RegAppName | Identity | OSUser | Trust |
|---|---|---|---|---|---|---|---|---|
| PISRV01 | 38 | D:\Program Files\PI\ | OPCpE(2748):remote | 2748 | PI-IN-UNIINT PI-IN-OS-OPC | PI Interfaces \| PIWorld | PISCHOOL\SVC-PIINT$ | |
| PISRV01 | 7 | D:\Program Files\PI\ | piarchss | 5592 | piarchss | | | |
| PISRV01 | 13 | D:\Program Files\PI\ | pibackup | 4116 | pibackup | | | |
| PISRV01 | 5 | D:\Program Files\PI\ | pibasess | 4848 | pibasess | | | |
| PISRV01 | 36 | D:\Program Files\PI\ | PI-ICU.exe(416):remote | 416 | InterfaceConfigurationUtility | piadmins \| PIWorld | PISCHOOL\student01 | |

When starting the interface, check for errors in the log file.

14. View the log file with the PISDKUtility




On buffered data collectors, you will want the PI Interface to be dependent on the PI Buffer Subsystem (PIBuffss). This will assure that the buffer starts before the PI Interface, because if PI Interface starts before the PI Buffer Subsystem it will start sending values to the PI Data Archive directly, thus bypassing the buffer.

***You will create the points collected by this PI Interface in the next chapter.***

## 6.14 Exercise – PI Interface for Random and RampSoak Simulator Data Configuration with PI ICU

Follow the instructions precisely!!!

**Exercise Objectives**

- Configure the PI Interface for Random Simulator Data and PI Interface for RampSoak Simulator Data.
- Create the predefined default PI Points using PI Builder
- Verify the values are updating.

**Problem Description**

You utilize the knowledge gained in the previous exercise to set up the PI Interfaces for Random and RampSoak Simulator Data on PIINT01. Then you create the predefined PI Points for those interfaces and verify the values are updating.

**Approach**

1. Add new interface from BAT file in PI ICU. Locate the **RANDOM.BAT_NEW** file at **%PIHOME64%\Interfaces\Random** directory as PI Interface for Random Simulator Data is 64-bit application, unlike 32-bit PI Interface for OPC DA.
2. In General section keep the **"*R*"** point source and Interface ID blank. Edit first scan class from **00:00:30** to **00:00:05**
3. In random interface section check *Fill time gaps with data.*
4. In UniInt section check *Include Point Source in the header of log messages*.
5. In Services section create service under default *NT Service\random1* account.
6. User Services Manager to switch it for gMSA **PISCHOOL\SVC-PIINT$**.
7. Re-open PI ICU to see the gMSA was applied and start the service.
8. Check the local PI Message Log and **Operations → Network Manager Statistics** in PI SMT for **RandE** application.

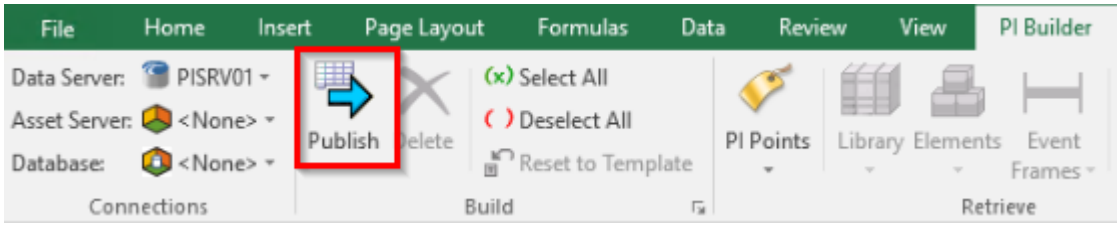
```
PISRV01   43   D:\Program Files\PI\   RandE(5420)remote        5420   PI-IN-UNIINT PI-IN-OS-R...   PI Interfaces | PIWorld   PISCHOOL\SVC-PIINT$
```

9. Repeat the steps 1 to 8 for PI Interface for RampSoak Simulator Data.
    - File **RMP_SK.BAT_NEW** is located at **%PIHOME64%\Interfaces\Rmp_sk** directory.
    - Keep the point source **"*9*"** and Interface ID blank.
    - In Network Manager Statistics look for **RmpSE** application

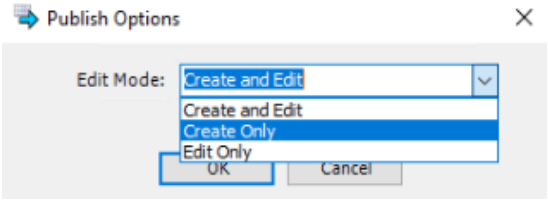
```
PISRV01   45   D:\Program Files\PI\   RmpSE(5580)remote        5580   PI-IN-UNIINT PI-IN-OS-R...   PI Interfaces | PIWorld   PISCHOOL\SVC-PIINT$
```

10. On PISRV01 open Exercise Files folder from the desktop shortcut and open MS Excel file **Default PI Points for Random and RampSoak.xlsx.**

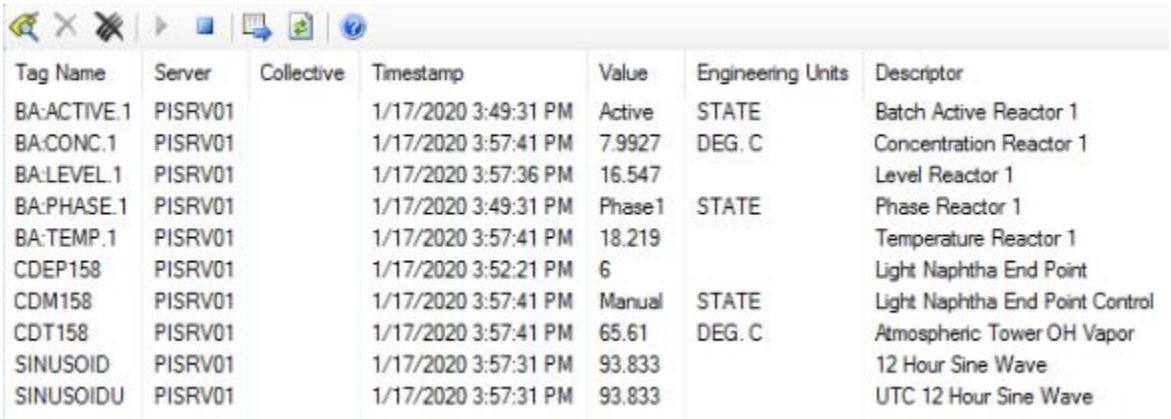
11. What all those columns mean will be explained in the next chapter PI Points. Right now, we only need to create them quickly. Switch to PI Builder tab and define Data Server to be PISRV01.
12. This enables the Publish button:




13. Hit it and from Publish Options dialog select Create Only and click OK




14. Open PI SMT and navigate to **Data** → **Current Values** plug-in.
15. Click on the Tag Search  icon to open Tag Search dialog.
16. Keep the fields as default (“*”) and click on Search button to search for all available tags. Then click on Select All and OK.
17. All tags should have values. Click on play  button for automatic value updates.



| Tag Name | Server | Collective | Timestamp | Value | Engineering Units | Descriptor |
|---|---|---|---|---|---|---|
| BA:ACTIVE.1 | PISRV01 | | 1/17/2020 3:49:31 PM | Active | STATE | Batch Active Reactor 1 |
| BA:CONC.1 | PISRV01 | | 1/17/2020 3:57:41 PM | 7.9927 | DEG. C | Concentration Reactor 1 |
| BA:LEVEL.1 | PISRV01 | | 1/17/2020 3:57:36 PM | 16.547 | | Level Reactor 1 |
| BA:PHASE.1 | PISRV01 | | 1/17/2020 3:49:31 PM | Phase1 | STATE | Phase Reactor 1 |
| BA:TEMP.1 | PISRV01 | | 1/17/2020 3:57:41 PM | 18.219 | | Temperature Reactor 1 |
| CDEP158 | PISRV01 | | 1/17/2020 3:52:21 PM | 6 | | Light Naphtha End Point |
| CDM158 | PISRV01 | | 1/17/2020 3:57:41 PM | Manual | STATE | Light Naphtha End Point Control |
| CDT158 | PISRV01 | | 1/17/2020 3:57:41 PM | 65.61 | DEG. C | Atmospheric Tower OH Vapor |
| SINUSOID | PISRV01 | | 1/17/2020 3:57:31 PM | 93.833 | | 12 Hour Sine Wave |
| SINUSOIDU | PISRV01 | | 1/17/2020 3:57:31 PM | 93.833 | | UTC 12 Hour Sine Wave |

# 7. PI Points

**Objectives**

- Define a PI Point
- Describe the different point types
- Describe the basic point attributes
- Build and edit points with Point Builder
- Describe a digital state set
- Create a digital state set
- Create digital state points
- Build and edit points with the PI Builder add-in to Excel.
- Connect the OPC data to PI points

## 7.1 What is a PI Point?

A PI point is a unique storage point for data in the PI Server.

For more information see "PI Point Classes and Attributes" in *PI Data Archive System Management Guide*.

Some examples are:

- A flow rate from a flow meter (would use floating point [also known as float, real] data)
- A DCS controller's mode of operation (may use digital or discrete data)
- The batch number of a product (can use one of float, integer, or string data)
- Text comments from an operator (using string [character] data)
- The result of a calculation (float or integer data)
- Memory % usage in a server (uses floating point data)

**Note**: Some industries and customers use the term "tag." In the PI system, *point, tag* and *data stream* are synonymous.

### 7.1.1 Point Class

The Point Class is simply the name for a defined set of point *attributes*. The PI Data Archive is pre-configured with the point classes you will need. The typical PI System has no need for additional point classes.

All points are based on the **Base** point class. However, these points do not have the complete set of attributes required to collect data via an interface. Although points created by PI Connectors are of Base point class

The **Classic** point class contains all the Base point class ***plus all the attributes required by the interface to connect to the data source and collect data.***

Tip

There are a handful of predefined “point classes” in the PI System and you can create your own, but these are special situations. Most of the points you will create will use the **classic** point class.

### 7.1.2 Point Type

The PI Data Archive, designed to collect and store **time-series** data, can store almost any data type.

There is no absolute when selecting point type but matching the PI point type with the data type on the source is usually a good start. For example, if the data source indicates that the data collected is a REAL32 then you would most likely use Float32 (a 32-bit floating-point value).

Tip

The PI Float16 is not a real data type – it was made up for the PI Server. It is actually a floating-point value scaled to a 16-bit integer. Developed decades ago, when disk space was scarce and expensive, you should not use this point type unless you have a specific reason to do so.

## 7.2 Directed Exercise – Data Types

You are invited to interact with the instructor to explore the different concepts presented in this section.

**Problem Description**

Identify the measurement that can be associated with each data type.

**Example:** Float32: ___pressure in bars UOM___

The table below lists the common PI Data Types. List another few example to the point type in the second column.

| | |
|---|---|
| Digital | |
| Int16 | |
| Int32 | |
| Float32 | |
| Float64 | |
| String | |

Now do the reverse and think about what data types would be appropriate for the following measurements:

| | |
|---|---|
| switch position | |
| Batch ID | |
| operator comments | |
| calculation results | |
| % remaining server disk space | |
| current reaction phase | |
| conveyor load | |

## 7.3 Directed Activity – Point Attributes

Research point attributes.

**Activity Objectives**

- Describe the Point Attributes for a **Classic** point

**Approach**

You may need to use the documentation for this exercise.

Use the table on the next page.

Match the attribute description to the attribute name for selected Point Attributes for the **CLASSIC** point class.

Each person will research attributes and match the proper description.

When you are done with the exercise, the instructor will go through each attribute to review them.

To complete the activity, you can use "Base Class Point Attributes" and "Classic Class Point Attributes" in *PI Data Archive System Management Guide*.

| N. | Attribute |
|---|---|
| 1 | CompDev |
| 2 | Span |
| 3 | Zero |
| 4 | ExcDev |
| 5 | Shutdown |
| 6 | Archiving |
| 7 | Location1 |
| 8 | Future |
| 9 | ExcMax |
| 10 | DigitalSet |
| 11 | ExcMin |
| 12 | EngUnits |
| 13 | Compressing |
| 14 | PointSource |
| 15 | CompMax |
| 16 | Location4 |
| 17 | InstrumentTag |
| 18 | Step |
| 19 | ExDesc |
| 20 | DisplayDigits |

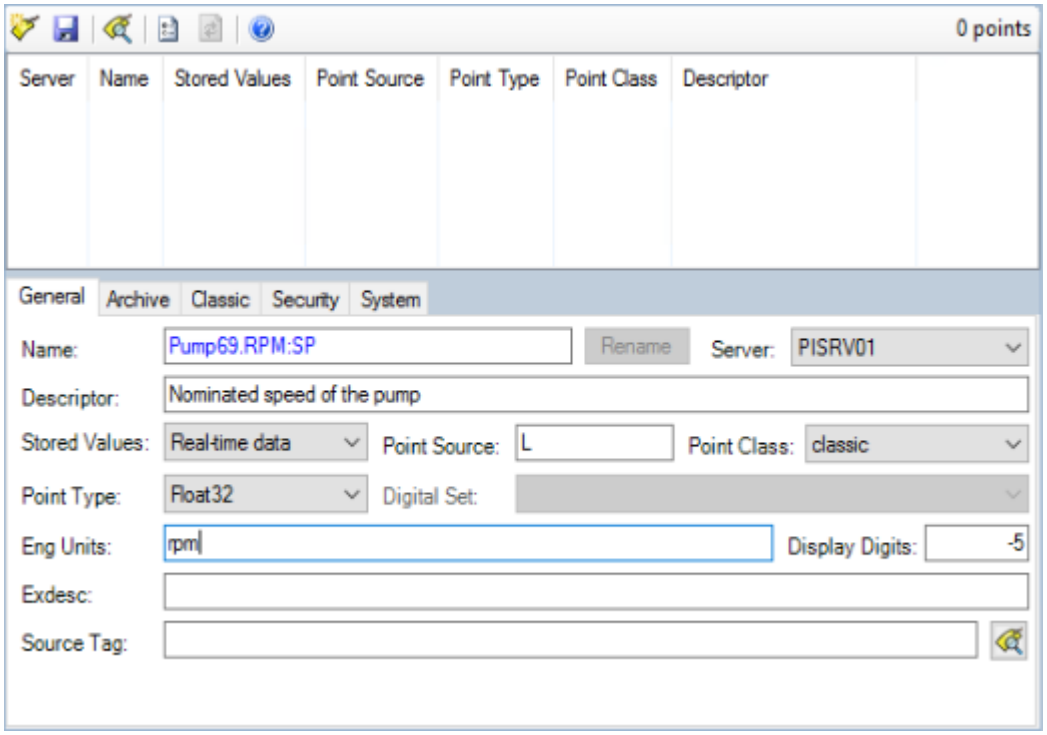
| N. | Description |
|---|---|
| A | Flag set to ON (1) for a point to be archived. Set to OFF (0) to stop archiving of a point. |
| B | Flag se to ON (1) to apply compression algorithm to a point. Set to OFF (0) to record every values in the archive. |
| C | Deviation for compression algorithm in engineering units. |
| D | Specifies the name of the digital state set associated with the tag. |
| E | Controls the format of numeric values on screens and in reports. |
| F | Describes the units of the measurement. |
| G | Provides additional information. Some interfaces use it to encode additional configuration information. |
| H | Specification of a deviation for exception-reporting. |
| I | Maximum time for compression. Duplicate values are archived if the elapsed time exceeds the value. |
| J | Associates a tag with an interface or PI application |
| K | Records a time-stamped event to a tag when the archive was shut down. |
| L | Minimum time for exception-reporting. For interface points the value should be 0. |
| M | The difference between the top of the range and the bottom of the range. Required for all numeric data type points. |
| N | Defines how numeric values are interpolated. Flag se to OFF (0) for values as continuous signal. Set to ON (1) for discrete values. |
| O | Used by some interfaces as the tag in the external system. |
| P | Maximum time for exception-reporting. Duplicate values are archived if the elapsed time exceeds the value. |
| Q | Specifies the interface ID for most of the interfaces. |
| R | Flag set to ON (1) for a point to receive values with future timestamps. Set to OFF (0) for a point to receive historical values. |
| S | Specifies the scan class number for most of the interfaces. |
| T | Indicates the lowest values possible. Required for all numeric data type points. |

## 7.4 Creating and Managing PI Points

There are many ways to create points in the PI Server. Throughout the course, we will show you the most common.

One tool that may be used to buld and edit points is **Point Builder** in PI SMT.

The Point Builder plug-in for PI SMT is a graphical tool that allows the user to create and edit PI points. This tool allows the system manager to set the attributes for each point individually during PI point creation and allows you to edit them afterward. Some attributes are system assigned and cannot be changed.




It is possible to rename a point while preserving the historical data associated with it. It is also possible to delete a point; in which case the associated archived data is ***no longer accessible***!

## 7.5 PI Point Attributes and PI Interfaces

Remember the statement, “always reading the documentation manual”? Each interface can use point attributes in a different manner. That is why each interface documentation specifies what point attributes are used and how.

Listed below are the common point attributes and how they are ***commonly*** used.

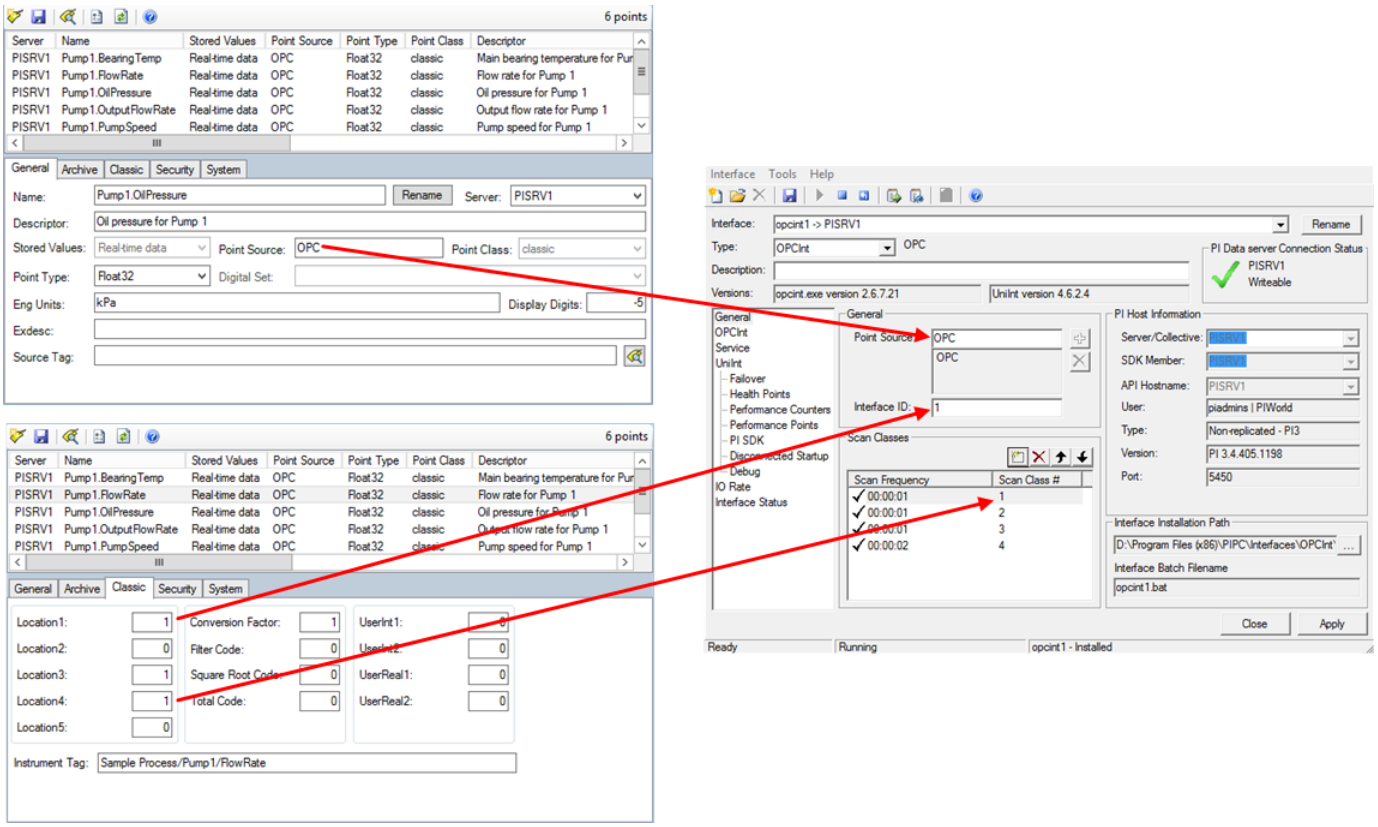
***ALWAYS consult the interface manual!***

| | |
|---|---|
| **Instrument Tag** | Name of the point/location in the source data system. *Often it must match the data source exactly!* |
| **Extended Descriptor** | Place for detailed query instructions. |
| **Future Data** | If defined as ‘*Allow*’ it means that events with time stamps in the future may be stored. |
| **Exception Specifications** | Defines a significant change in value. |
| **Point Source** | Must match the value set in the interface configuration. See the /PS parameter in the interface start-up file. |
| **Location1** | *Typically*, the Location1 field is used for the interface instance number (/ID) |
| **Location4** | *Typically,* the field is the scan class number. (/f) |
| **Scan** | Include the PI point in the list of points to scan (always set to ON) |

.

Tip

*Instrument Tags are often case sensitive*. Copy and paste this information directly into PI SMT Point Builder or MS Excel PI Builder from the PI OPC Client Tool when possible.




*The most common cause of malfunctioning new points is incorrect mapping of PI point attributes to the data source or the interface configuration.*

### 7.5.1 A Word about Future Data.

The PI Data Archive accepts data with timestamps up until January 2038. January 1970 is the past limit for all points.

In order to store future data, a point must be created with the "**Stored Values**" point attribute set to 1, or '**Future data'**. By default, the "Stored Values" attribute is set to 0 (**Real-time data**). This cannot be modified after a point has been created. Points with the attribute set to 0 will reject data with timestamps more than 10 minutes into the future.

Generally speaking. Future points should be used when storing data that is not collected sequentially in chronological order. For example, process or operational data should be kept in historical points because it is measured and collected in real time. On the other hand, forecasts and predictive data over an arbitrary time range are suited for future points.

Data for all future points is stored in separate archive files. Future data never moves to the historical archive files even after the future data ages into the past. The archive files for future points will be created automatically as data is written to the points and are resized dynamically. Each future archive has an initial size of 1 MB and grows dynamically, and the duration of the archive is always one calendar month. If neither the initial size nor the duration is desirable, archives can be manually created and registered.

## 7.6 Polled, Advised or Output Points

Many interfaces have different methods of retrieving data. All three ***read*** types of points are read asynchronously by the interface and the same data routines process all updates. The method is often set in a **Location Code**.

These are described below:

### *Polled*

For polled points, the interface sends an asynchronous refresh call for the group (points of the same scan class). This is often the most common method for reading data and is supported by virtually every interface.

### *Advise*

Advise points "listen" for new events. For Advise points (referred to as read on change in the OPC Standard), the data source sends data whenever a new value is read into the server's cache. Not all interfaces support the Advise method.


Often the Advise method of reading data is the most efficient and best performing.

**Caution:** Do not mix advised points and polled points in the same group (i.e. scan class). Some interfaces do not tolerate this and will not function correctly.

### *Output Points*

Output points read a separate PI Point and ***write*** the value out to the data source. These often reference calculation points which values are calculated by PI Analysis Service and are performing a calculation that cannot be performed on the data source. Only several interfaces support bi-directional data.

PI Interfaces capable writing value to data source, such as PI Interface for OPC DA or PI Interface for Modbus Ethernet, are published in two editions: **Read Write** a **Read Only**.

With Read Only edition, the function of writing values to data source is disabled due to security reasons.

## 7.7 Exercise – OPC Points

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

- Build PI OPC Points.
- Validate PI OPC points are collecting data from an OPC server.

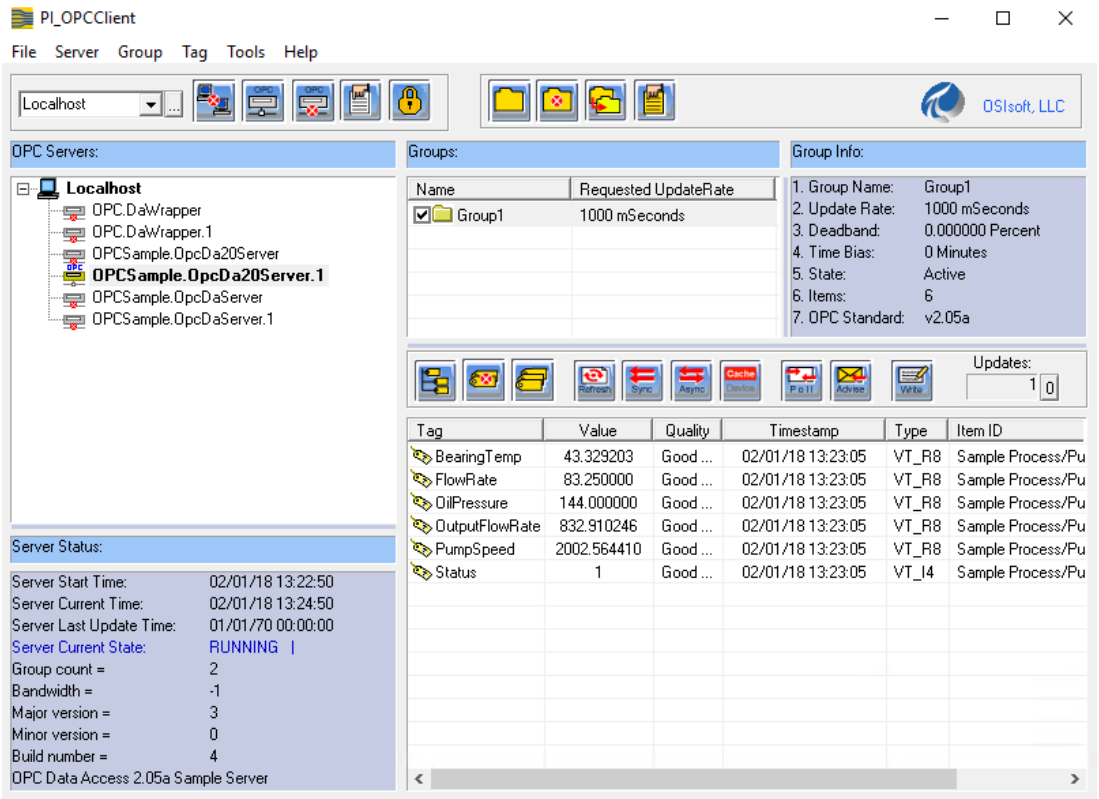
### Problem Description

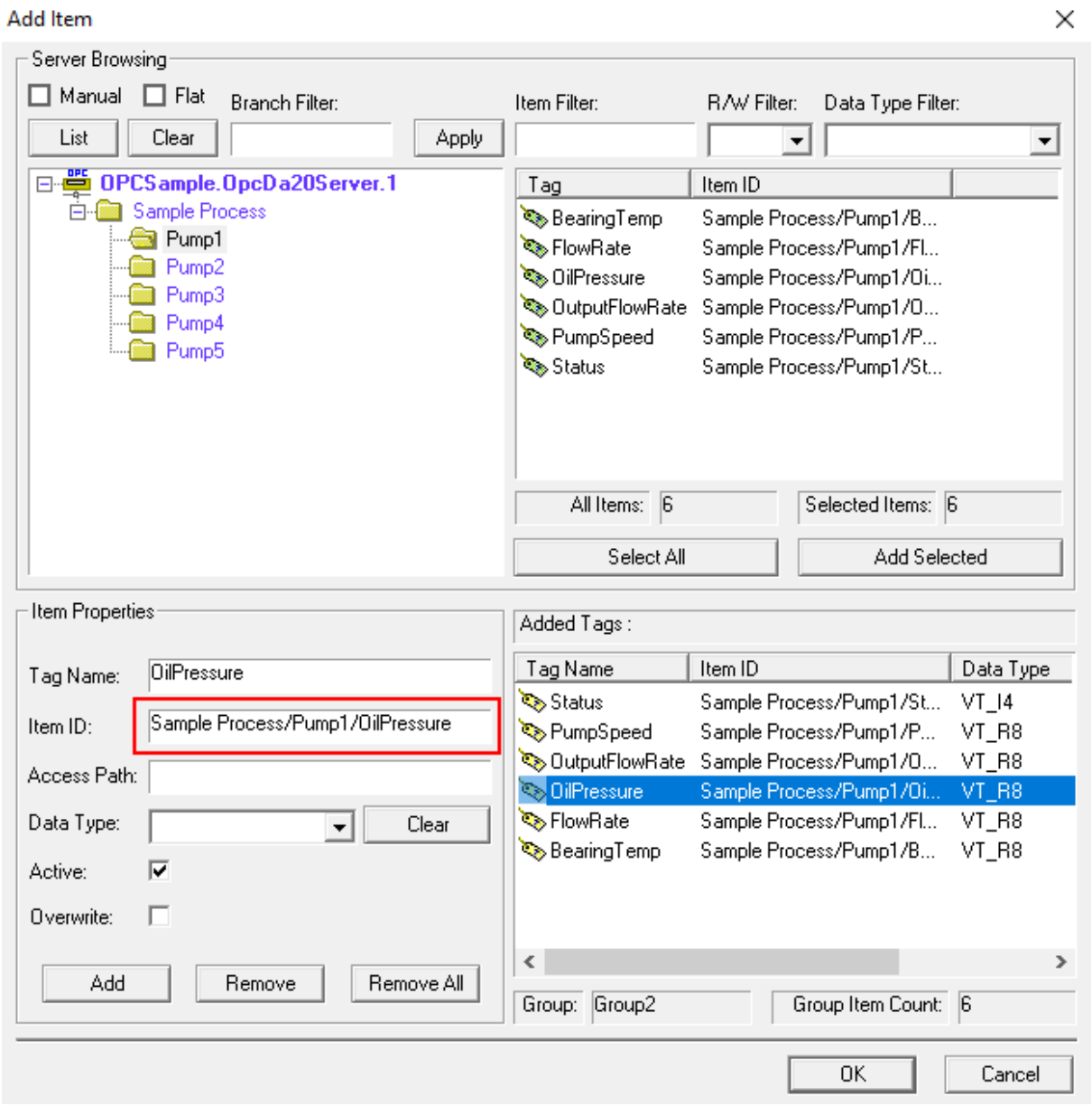
You have a set of Pumps connected to your OPC Server generating data. (***You have already configured your PI Interface for OPC DA***).

You need to create points in the PI Data Archive.

### Approach

Use the **PI OPC Client Tool** (installed with the PI Interface for OPC DA) on PIINT01 to examine your OPC Server. You should notice that there is a series of pumps. Each pump has points associated with it, collecting simulated data.






Use the *PI Interface for OPC DA User Guide* and the information contained in the PI ICU to configure the points for the first pump only. **DO NOT** build the **Status** point just yet. Pay attention to the Location codes (1,3,4); and the relationship between the PI point attribute **Instrumenttag** and the OPC **ItemID.**

1. Use PI OPC Client Tool Add Item dialog window to add Items of **Pump1** to the Added Tags list as in the screenshot above. Useful to copy the Item ID.
2. On PISRV01 in PI SMT **Points** → **Point Builder** define the necessary point attributes. The naming syntax you use will be: **<Pump#>.<Tag Name>**

a. General Tab:
   - Name: **Pump1.OilPressure**
   - Descriptor: **Oil Pressure for Pump 1**
   - Point Source: **OPC**
   - Point Type: **Float32**
   - Eng Units: **kPa**

b. Archive Tab:
   - Zero: **0**
   - Span: **1000**

c. Classic Tab:
   - Location 1: **1**
   - Location 3: **1**
   - Location 4: **1**
   - Instrument Tag: **Sample Process/Pump1/OilPressure**

3. Create the point by clicking on **Save** icon.

4. Repeat the procedure for the other tags except Status. Keep the same Point Source, Point Type, Location 1, Location 3 and Location 4 as for Pump1.OilPressure point. For other attributes use the following table:

| Name | Description | Eng Units | Zero | Span | Instrument Tag |
|---|---|---|---|---|---|
| Pump1.BearingTemp | Main bearing temperature for Pump 1 | °C | 0 | 100 | Sample Process/Pump1/BearingTemp |
| Pump1.FlowRate | Flow rate for Pump 1 | m3/h | 0 | 600 | Sample Process/Pump1/FlowRate |
| Pump1.OutputFlowRate | Output flow rate for Pump 1 | m3/h | 0 | 1000 | Sample Process/Pump1/OutputFlowRate |
| Pump1.PumpSpeed | Pump speed for Pump 1 | rpm | 0 | 3000 | Sample Process/Pump1/PumpSpeed |

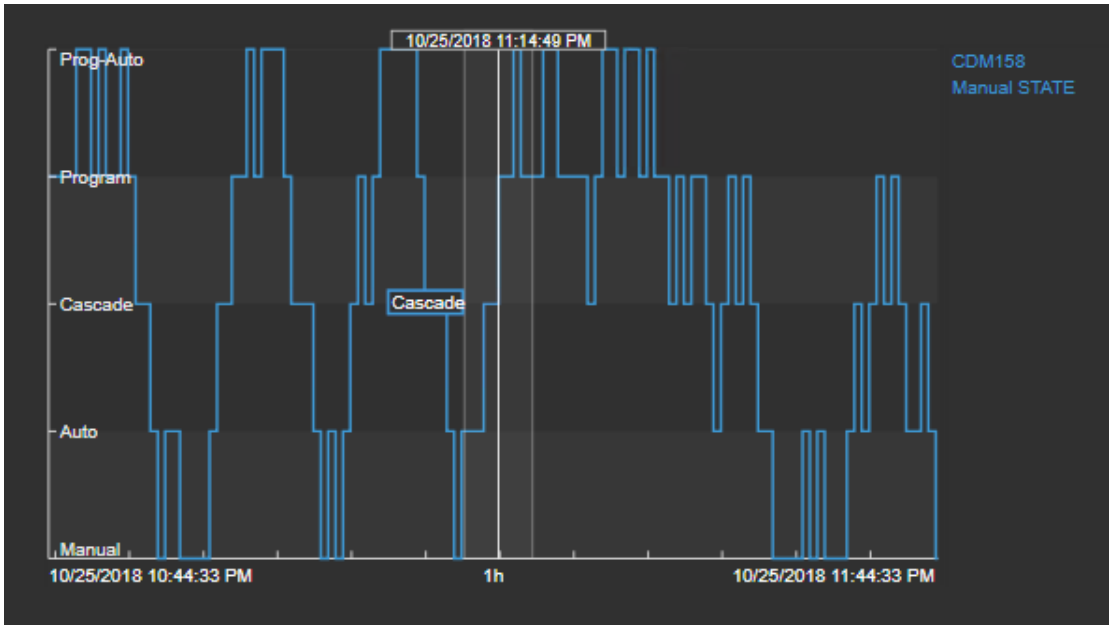
5. Go to **Data** → **Current Values**. Add created points to the list. Use Point Source: OPC as the filter. Verify all the points are updating.

**Tip**

Use the PISDKUtility message facility to see the points as they are recognized by the PI Interface for OPC DA – or the 'View PI Message Log Continuously' icon in the PI ICU. It will show if the point was picked up properly by the interface.

## 7.8 Digital State Sets

### 7.8.1 About Digital State Sets



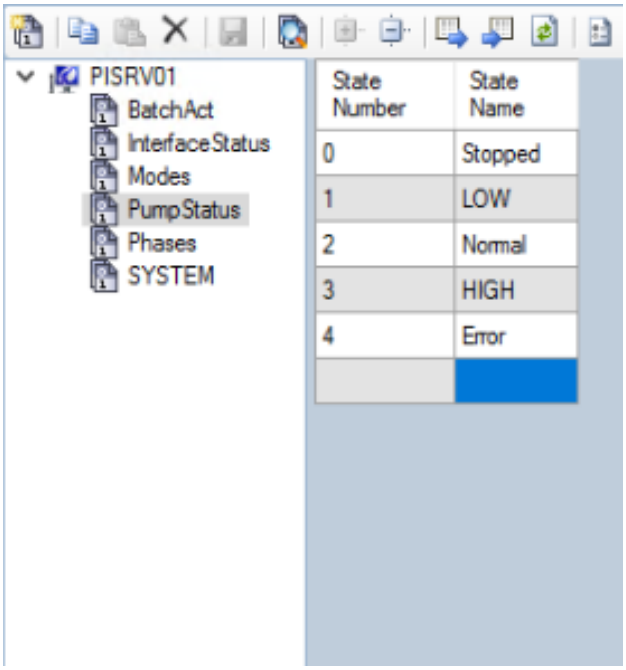

PI Points of type **Digital** are used to store data values that have discrete states exemplified above. Example of typical states are Open/Closed for a valve or On/Off for a switch. While the user is interested in the actual state, PI Data Archive stores this information as an integer. This integer is then associated with a **Digital State Set**; a grouping of these states. Whenever the value is requested, PI Data Archive retrieves the integer value, does a lookup in the Digital State Set, and then presents the associated text value.

Digital Sets are kept in a common table for all points of type Digital to access. The Digital State Set must exist prior to the creation of the digital point. When defining a point use the **DigitalSet** attribute to store the associated Digital Set name. The digital states are case preserving, but not case sensitive, so you can refer to them with upper and lower case, but any display will show the configured upper or lower case. There is a large default set called **System** that contains system error messages and other information. All points, including non-digital points, can receive and archive a state from the System digital states set (Shutdown, PtCreated, Over Range, Under Range, I/O Timeout, etc.).

**Do not edit the SYSTEM set on your PI System!**

### 7.8.2 Creating digital state sets

Create **Digital States** with the Digital States plug-in under the **Points** section in PI SMT. This allows you to create new Digital States, copy and paste existing Digital states, and edit or delete existing Digital States. Digital Sets are local to the PI Data Archive, so if you require the same Digital Sets on multiple PI Data Archives, you need to export and import the list of sets and states as a .csv file or copy / paste the state sets from one server to another by opening multiple PI Data Archives inside the plug-in.




To create a new Digital Set, select the Server then the New icon in the menu bar. You can also right click and select Add Set from the pop-up menu. A new table will appear with two columns. These columns are State Number and State Name. The State Name field corresponds to the states of the data source and is manually entered, although is a facility for importing digital states. The State Number field corresponds to the integer that PI System will store in the archive for that digital state. This field is populated automatically, starting at a value of 0 and increasing by 1 from there. Once you have fully entered all your digital states, remember to save the new digital states by clicking on the Save icon in the toolbar.

**Note**: Additions and edits to Digital Sets are immediately available on the PI Data Archive. However, most client applications, including PI SMT, cache the Digital Sets. *This means that you may have to exit and restart the client application for any changes to be visible.*

## 7.9 Exercise – Create a Digital State Set and Point

This activity is designed to maximize learning in a specific topic area.

### Exercise Objectives

- To create a digital state set and point.

### Problem Description

A new controller has been installed in your plant, and the OPC Data server, for which you configured PI Interface for OPC DA in an earlier exercise, is collecting its data. It has digital points.

You have decided to create a point to archive this controller's mode of operation. However, in order to build the point, you will first need a digital state set.

Use the PI OPC Client Tool (or other appropriate OPC tool in your own environment) to examine the digital states. Can you see the state changes?

The table on the right shows the correlation between state names and numbers.

| State Name | State Number |
|---|---|
| **Stopped** | **0** |
| **LOW** | **1** |
| **Normal** | **2** |
| **HIGH** | **3** |
| **Error** | **4** |

### Approach

1. On PISRV01 open the PI SMT and navigate to **Points** → **Digital States** plug-in.
2. Create new Digital State Set called **PumpStatus**.
3. Fill the State Names according the table above.
4. Save it and restart PI SMT. Why? _____
5. Navigate to **Points** → **Point Builder** plug-in.
6. Create a new point with the following attributes:
    a. General Tab: Name: **Pump1.Status**; Description: **Status of Pump 1**; Point Source: **OPC**; Point Type: **Digital**; Digital Set: **PumpStatus**
    b. Classic Tab: Location 1: **1**; Location 3: **1**; Location 4: **1**; Instrument Tag: **Sample Process/Pump1/Status**
    c. Archive Tab: Keep defaults. Step attribute greyed out? Why? _____
7. Save the configured PI Point.
8. Wait for the PI Interface to start collecting data for the new point.
9. Check the updates in **Data** → **Current Values** plug-in.

## 7.10 PI Builder

The PI Builder is an add-in to Microsoft Excel. The spreadsheet format is convenient when viewing and editing in bulk, with a row for each point or element and a column for each attribute.

The tool best suited to bulk build and edit points is **PI Builder**

PI Builder requires the spreadsheet to have the following layout:

- The attribute names are listed in the top row.
- The point names are listed in the second column.
- Each point has its attributes listed under the headings in the top row, one point per row.
- Select a point row by putting **X** in the first column. Import or export operations are performed on these selected points only.

Tip

Like every powerful tool, PI Builder can save you a lot of time, or do a lot of damage if not used carefully. Be careful!

### 7.10.1 Enabling Delete

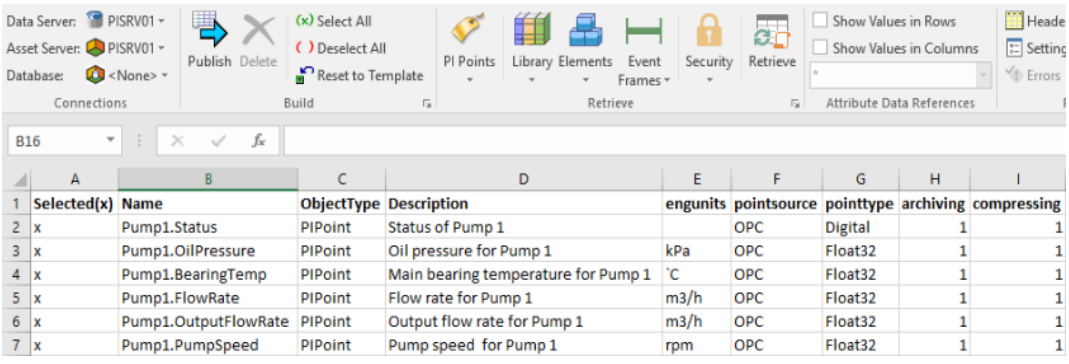
Notice that the Delete action is not enabled by default. There is no "undelete" for a point. ***If you delete a point by accident, you cannot get their history back***. The history is inaccessible or lost. Entire systems have been deleted by accident.

It is recommended that you only enable Delete when you need to; be ***extremely*** careful when using it. Turning the point attribute SCAN to OFF is a better alternative.

### 7.10.2 Export only What Matters

OSIsoft recommends that you export only attributes that have been changed*; that is, remove any unchanged columns*. If you don't do this, then all the attributes are exported, not only those that have changed.

Moreover, only export the points (rows) that are new or that you have changed.




| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Selected(x) | Name | ObjectType | Description | engunits | pointsource | pointtype | archiving | compressing |
| 2 | x | Pump1.Status | PIPoint | Status of Pump 1 | | OPC | Digital | 1 | 1 |
| 3 | x | Pump1.OilPressure | PIPoint | Oil pressure for Pump 1 | kPa | OPC | Float32 | 1 | 1 |
| 4 | x | Pump1.BearingTemp | PIPoint | Main bearing temperature for Pump 1 | `C | OPC | Float32 | 1 | 1 |
| 5 | x | Pump1.FlowRate | PIPoint | Flow rate for Pump 1 | m3/h | OPC | Float32 | 1 | 1 |
| 6 | x | Pump1.OutputFlowRate | PIPoint | Output flow rate for Pump 1 | m3/h | OPC | Float32 | 1 | 1 |
| 7 | x | Pump1.PumpSpeed | PIPoint | Pump speed for Pump 1 | rpm | OPC | Float32 | 1 | 1 |

## 7.11 Exercise – PI Builder

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

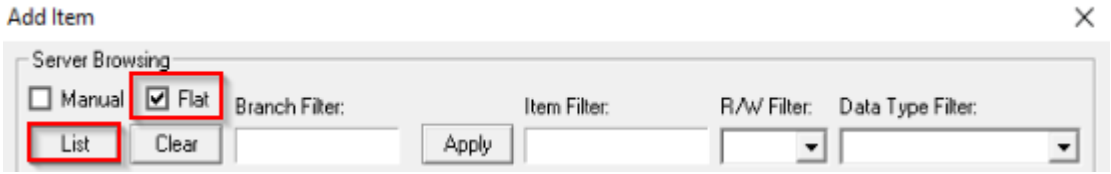
- Create points in bulk using the PI Builder
- Modify points in bulk using the PI Builder

### Problem Description

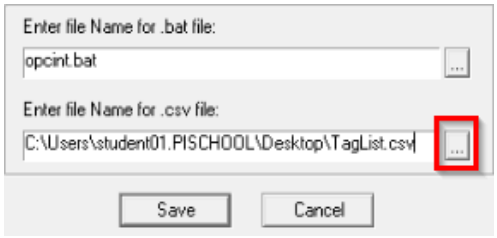
Create the remaining points of Pump2 to Pump5 for the PI Interface for OPC DA without tedious necessity of using the PI SMT Point Builder plug-in.

### Approach

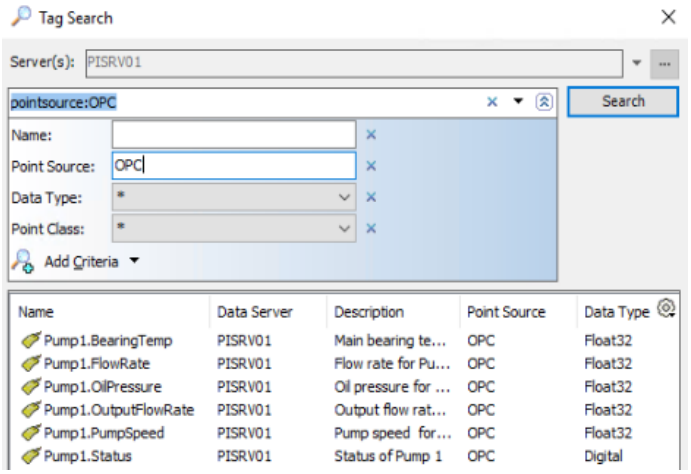
1. On PIINT01 open PI OPC Client Tool. Connect to *OPCSample.OpcDa20Server.1*
2. In Add Item window in Server Browsing section check **Flat** and hit **List**.




3. In total 30 items are listed. Select all items for Pump2 to Pump5 (Selected Items: 24) and click **Add Selected** button and OK.
4. On main menu toolbar go to **File** → **Save As…** and select the destination folder for **TagList.csv** file (for example Desktop). We do not care about the opcint.bat file.

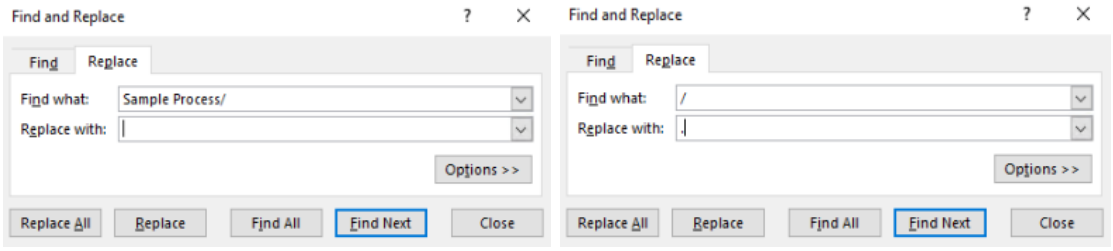


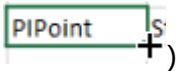

5. Copy the file from PIINT01 to PISRV01
6. On PISRV01 open MS Excel with PI Builder and with **PI Points** button load all PI points for Pump1 using the Point Source: OPC filter.
7. Click on Name column header to sort the points alphabetically.
8. Select all points using SHIFT, or CTRL+A and hit OK.




9. From Select Object Types and Column Headers check only the following attributes:
    a. *General*: Description, digitalset, engunits, pointsource, pointtype
    b. *Archive*: scan, span, zero
    c. *Classic*: instrumenttag, location1, location2, location3, location4, location5
10. All other attributes are not needed and should be unchecked. Except *Required Columns* as expected cannot be unchecked.
11. Open a new sheet in MS Excel, select Data tab and from section Get External Data click on **From Text** button.


12. Load the **TagList.csv** file you copied from PIINT01.
13. In Text Import Wizard on Step 2 select Comma as delimiter and finish the wizards with default settings.
14. Copy the instrumenttag column from Sheet2 to Sheet1 and append it to the instrumenttag column there to rows loaded for Pump1.
15. Using the replace all function (CTRL+H) replace "**Sample Process/**" with "**NULL**" and then replace "***/***" for "**.**" in Tag column to have point names according the naming syntax *<Pump#>.<Tag Name>* and again copy to Sheet1 Name column and append.

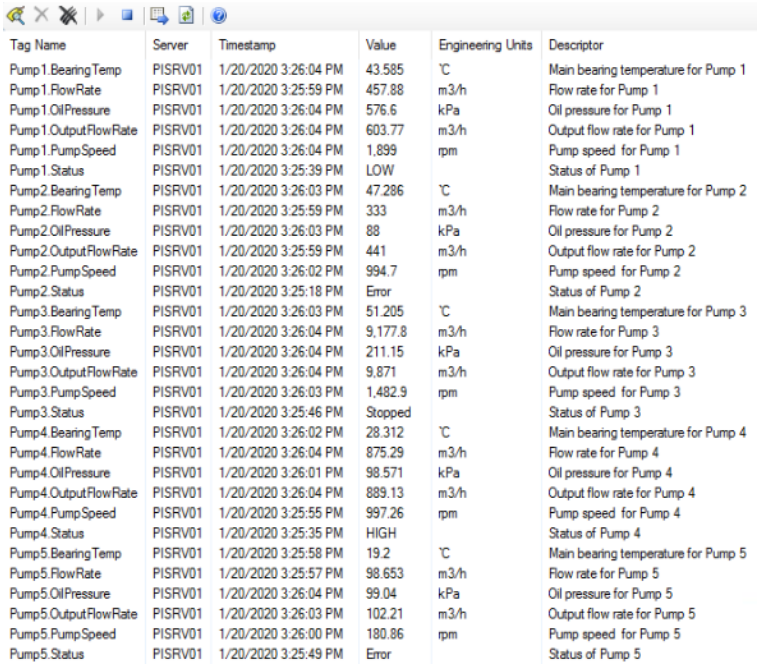



16. In Sheet1 replicate the values from columns ObjectType, pointsource, scan, location1, location2, location3, location4 and location5 to the rest of the rows. (You can use the double-click on the lower right corner of the cell 

)

17. For engunits, span and zero the values are repeating for each pump. Select the all rows for Pump1 in those columns and copy and paste below 4 times.
18. The same goes for Description column only there you must modify the pump number (CTRL+H).
19. Copy PumpStatus in digitalset to Status tag rows for Pump2 to Pump5. You PI points configuration is now complete.
20. Click on **(x) Select All** button 

to select all rows, but then deselect rows for Pump1 as those tags already exist.
21. Hit **Publish** and select Create Only option. Save the file.
22. In PI SMT **Data** → **Current Values** load the created tags and verify the values are updating.



| Tag Name | Server | Timestamp | Value | Engineering Units | Descriptor |
|---|---|---|---|---|---|
| Pump1.BearingTemp | PISRV01 | 1/20/2020 3:26:04 PM | 43.585 | ℃ | Main bearing temperature for Pump 1 |
| Pump1.FlowRate | PISRV01 | 1/20/2020 3:25:59 PM | 457.88 | m3/h | Flow rate for Pump 1 |
| Pump1.OilPressure | PISRV01 | 1/20/2020 3:26:04 PM | 576.6 | kPa | Oil pressure for Pump 1 |
| Pump1.OutputFlowRate | PISRV01 | 1/20/2020 3:26:04 PM | 603.77 | m3/h | Output flow rate for Pump 1 |
| Pump1.PumpSpeed | PISRV01 | 1/20/2020 3:26:04 PM | 1,899 | rpm | Pump speed for Pump 1 |
| Pump1.Status | PISRV01 | 1/20/2020 3:25:39 PM | LOW | | Status of Pump 1 |
| Pump2.BearingTemp | PISRV01 | 1/20/2020 3:26:03 PM | 47.286 | ℃ | Main bearing temperature for Pump 2 |
| Pump2.FlowRate | PISRV01 | 1/20/2020 3:25:59 PM | 333 | m3/h | Flow rate for Pump 2 |
| Pump2.OilPressure | PISRV01 | 1/20/2020 3:26:03 PM | 88 | kPa | Oil pressure for Pump 2 |
| Pump2.OutputFlowRate | PISRV01 | 1/20/2020 3:25:59 PM | 441 | m3/h | Output flow rate for Pump 2 |
| Pump2.PumpSpeed | PISRV01 | 1/20/2020 3:26:02 PM | 994.7 | rpm | Pump speed for Pump 2 |
| Pump2.Status | PISRV01 | 1/20/2020 3:25:18 PM | Error | | Status of Pump 2 |
| Pump3.BearingTemp | PISRV01 | 1/20/2020 3:26:03 PM | 51.205 | ℃ | Main bearing temperature for Pump 3 |
| Pump3.FlowRate | PISRV01 | 1/20/2020 3:26:04 PM | 9,177.8 | m3/h | Flow rate for Pump 3 |
| Pump3.OilPressure | PISRV01 | 1/20/2020 3:26:04 PM | 211.15 | kPa | Oil pressure for Pump 3 |
| Pump3.OutputFlowRate | PISRV01 | 1/20/2020 3:26:04 PM | 9,871 | m3/h | Output flow rate for Pump 3 |
| Pump3.PumpSpeed | PISRV01 | 1/20/2020 3:26:03 PM | 1,482.9 | rpm | Pump speed for Pump 3 |
| Pump3.Status | PISRV01 | 1/20/2020 3:25:46 PM | Stopped | | Status of Pump 3 |
| Pump4.BearingTemp | PISRV01 | 1/20/2020 3:26:02 PM | 28.312 | ℃ | Main bearing temperature for Pump 4 |
| Pump4.FlowRate | PISRV01 | 1/20/2020 3:26:04 PM | 875.29 | m3/h | Flow rate for Pump 4 |
| Pump4.OilPressure | PISRV01 | 1/20/2020 3:26:01 PM | 98.571 | kPa | Oil pressure for Pump 4 |
| Pump4.OutputFlowRate | PISRV01 | 1/20/2020 3:26:04 PM | 889.13 | m3/h | Output flow rate for Pump 4 |
| Pump4.PumpSpeed | PISRV01 | 1/20/2020 3:25:55 PM | 997.26 | rpm | Pump speed for Pump 4 |
| Pump4.Status | PISRV01 | 1/20/2020 3:25:35 PM | HIGH | | Status of Pump 4 |
| Pump5.BearingTemp | PISRV01 | 1/20/2020 3:25:58 PM | 19.2 | ℃ | Main bearing temperature for Pump 5 |
| Pump5.FlowRate | PISRV01 | 1/20/2020 3:25:57 PM | 98.653 | m3/h | Flow rate for Pump 5 |
| Pump5.OilPressure | PISRV01 | 1/20/2020 3:26:04 PM | 99.04 | kPa | Oil pressure for Pump 5 |
| Pump5.OutputFlowRate | PISRV01 | 1/20/2020 3:26:03 PM | 102.21 | m3/h | Output flow rate for Pump 5 |
| Pump5.PumpSpeed | PISRV01 | 1/20/2020 3:26:00 PM | 180.86 | rpm | Pump speed for Pump 5 |
| Pump5.Status | PISRV01 | 1/20/2020 3:25:49 PM | Error | | Status of Pump 5 |

You may choose to use the **PI Points for OPC DA Int – Pumps.xlsx** spreadsheet provided in the Exercise Files directory, to create the PI Points if falling behind or due to time constrains.

**DO NOT PROCEED** beyond this point until all points are collecting data. Many of the remaining course exercises depend on this exercise.

Information on building and managing points is found in *PI Data Archive System Management Guide.*

See *PI Builder User Guide* for information on the PI Builder for Excel.

What are common causes for no data being collected?

# 8. PI Vision

**Objectives**

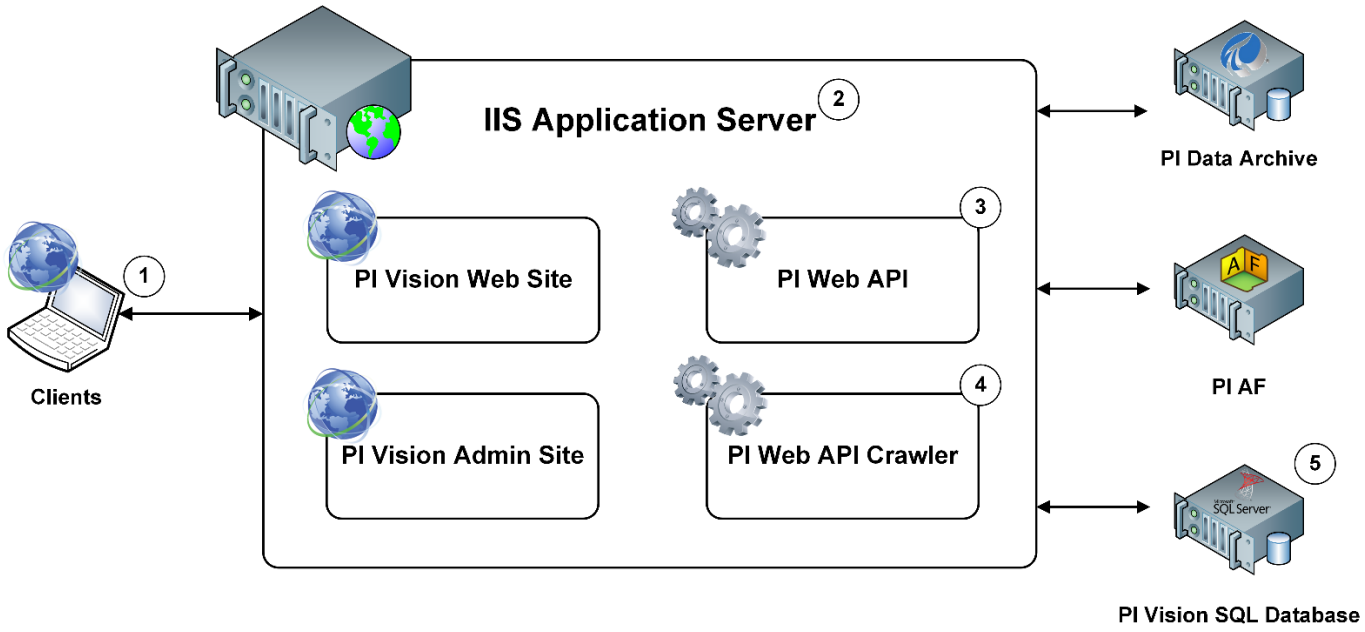
- Describe PI Vision architecture
- Introduce Kerberos Delegation
- Install and configure PI Vision
- Install PI ProcessBook to PI Vision Migration Utility
- Create a simple display in PI Vision

Detailed installation and configuration of PI Vision is described in *PI Vision Installation and Administration Guide*.

## 8.1 PI Vision Architecture

PI Vision is a web-browser based application that enables you to easily retrieve, monitor, and analyze process engineering information.

The main components of PI Vision installation are:




1. **Clients** are individual PI Vision users accessing PI System data. PI Vision is supported by most modern browsers on a wide variety of devices, including tables and phones running iOS or Android operating system.

2. **PI Vision IIS Application Server** provides the execution environment for PI Vision. The application server handles all application operations between users (clients) and PI Data Archive servers, PI AF servers, MS SQL Server and PI Web API.
3. **PI Web API** provides cross-platform, multi-user programmatic access to PI System data
4. **PI Web API Crawler** enables the search functionality of PI Vision by periodically crawling the registered servers for metadata and creating a search index.
5. **PI Vision SQL Database** on MS SQL Server stores user display settings and definitions. Display definitions include such data as a display name and display owner, symbols on the display, user permissions etc.

OSIsoft recommends that PI Vision uses the **SAME** MS SQL Server that PI AF uses. Otherwise, you can either install MS SQL Server on the same computer as PI Vision IIS application server or you can use a dedicated MS SQL Server installation.

OSIsoft strongly recommends that the PI Data Archive server(s) and PI AF server(s) be in the **SAME DOMAIN** as PI Vision IIS Application Server and MS SQL Server hosting the PI Vision SQL Database

## 8.2 PI Vision Data Flow

Example of a typical data flow that occurs as different architecture elements interact to provide the client with PI System data.

**When searching for data in a new display:**

1. When a user performs a search for a data item (asset, attribute or PI point), the client sends a request for that data item to the PI Vision IIS Application Server. The request is relayed to the PI Web API which populates the search results retrieved from the local index created by the PI Web API Crawler.
2. Based on the search results, the user can create a symbol for the data item by dragging it onto the display. The symbol on the display does not yet contain any PI System data values.
3. Symbol creation triggers a request for PI System data, which PI Vision IIS Application Server relays to PI Data Archive or PI AF. When PI System data returns to the client, the symbols on the display are populated with data values.
4. When a user saves the display, the display definition (display settings) is send to the MS SQL Server and stored in the PI Vision SQL database.

**When opening and existing display:**

1. When a user opens an existing display, the client side sends a request for a display definition to the PI Vision IIS Application Server, which is relayed to the MS SQL Server. The MS SQL Server returns the display definition from which the client generates a display with symbols. The symbols do not yet contain any data values.

2. Symbol creation triggers a request for PI System data, which PI Vision IIS Application Server relays to PI Data Archive or PI AF. When PI System data returns to the client, the symbols are populated with data values.

## 8.3 Addresses, Application Pools, Services and Local Groups

### 8.3.1 Web Addresses

PI Vision uses three websites:

- The administration website used for initial configuration and maintenance:
  - *https://<WebServer>/PIVision/Admin*
- The main application website for creating and visualizing displays:
  - *https://<WebServer>/PIVision*
- PI Web API Indexed Databases page for checking the status or recreating indexes for PI Vision search dialog:
  - *https://<WebServer>/piwebapi/admin/search/database.html*

<*WebServer*> is the hostname or FQDN of the PI Vision IIS Application Server where Microsoft Internet Information Services (**IIS**) server role is installed.

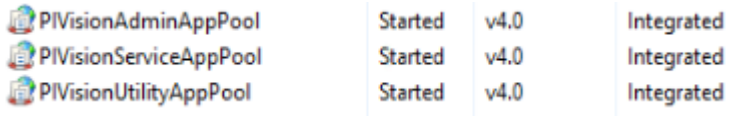
### 8.3.2 Application Pools

An IIS application pool is a pool - i.e. a collection - that houses applications on IIS. Each application pool consists of a process called **w3wp.exe** that runs on the server machine.

PI Vision installation creates three application pools:

- *PIVisionAdminAppPool* runs the Administration website
- *PIVisionServiceAppPool* runs the main PI Vision application website
- *PIVisionUtilityAppPool* runs the PI ProcessBook to PI Vision Migration Utility website.

The applications pools run under Windows service accounts that have appropriate access permissions across the PI System.




### 8.3.3 Windows Services

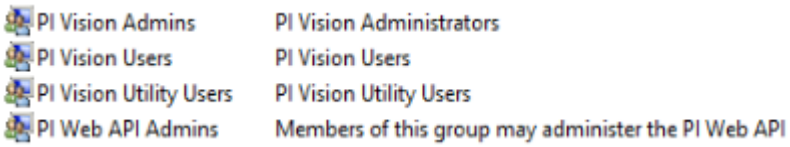
PI Vision installation installs PI Web API that creates two services:

- *PI Web API*: A RESTful API to the PI System.
- *PI Web API Crawler*: crawls PI System data sources to provide metadata to indexed search.

### 8.3.4 Local Groups

PI Vision installation creates four local groups on PI Vision IIS Application Server:

- *PI Vision Admins*: Members of this group are granted access to the PI Vision Administration website *https://<WebServer>/PIVision/Admin.*
- *PI Vision Users*: Members of this group are granted access to the main PI Vision application website *https://<WebServer>/PIVision*.
- *PI Vision Utility Users*: Members of this group are granted permission to use PI ProcessBook to PI Vision Migration Utility to migrate PI ProcessBook displays to native PI Vision displays.
- *PI Web API Admin*: Members of this group are granted permissions to access PI Web API search indexes website *https://<WebServer>/piwebapi/admin/search/database.html*.




## 8.4 Hardware and Software Requirements

### 8.4.1 PI Vision IIS Application Server requirements

The requirements for the software on the machine the PI Vision IIS Application Sever are:

- Microsoft Windows Server 2012 64-bit and later, incl. Server Core versions
- Microsoft Information Services (IIS) 8.0 and later
- Microsoft .NET Framework 4

The following table show the recommend hardware sizing based on the number of users (clients) connecting to the PI Vision IIS Application Server at the same time:

| # of Users (Clients) | 1 to 50 | 50 to 250 | 250 to 500 |
|---|---|---|---|
| CPU: # of Cores | 4 | 4 | 8 |
| CPU: Speed (GHz) | 2 | 2.5 | 3 |
| RAM (GB) | 6 | 12 | 24 |

### 8.4.2 MS SQL Server requirements

PI Vision requires MS SQL Server 2014 or above. All editions (Express, Standard, Enterprise) are supported.

### 8.4.3 PI System requirements

- PI Data Archive version 3.4.380 and later
- PI AF version 2018 (2.10) and later

### 8.4.4 Client requirements

PI Vision clients need to use web browsers that are HTML5 compatible:

- Microsoft Internet Explorer 11
- Microsoft Edge 44 and later
- Google Chrome 76 and later
- Mozilla Firefox 69 and later
- Safari 11 and later

## 8.5 Kerberos Delegation

Kerberos delegation is a network authentication protocol that allows users in a distributed application environment to securely access remote data sources. Kerberos delegation is designed to provide strong authentication for client/server applications by using secret key cryptography. Clients obtain tickets from the Kerberos Key Distribution Center on Domain Controller and provide these tickets to servers when connections are established.

### 8.5.1 Kerberos Double Hop

Simply speaking, in order to view data from PI Data Archive and PI AF in PI Vision, client's (user's) account must authenticate not only to PI Vision IIS Application Server, but also to PI Data Archive and PI AF. PI Vision IIS Application server needs to forward (delegate) client's credentials to PI Data Archive and PI AF for authentication. This is called a **Double Hop** scenario.

**Kerberos Double Hop** is a term used to describe our method of maintaining the client's Kerberos authentication credentials over two or more connections. In this fashion we can retain the user's credentials and act on behalf of the user in further connections to other servers.

It works in 5 steps:

*Step 1* – Client provides credentials to domain controller and it returns a Kerberos ticket to the client.

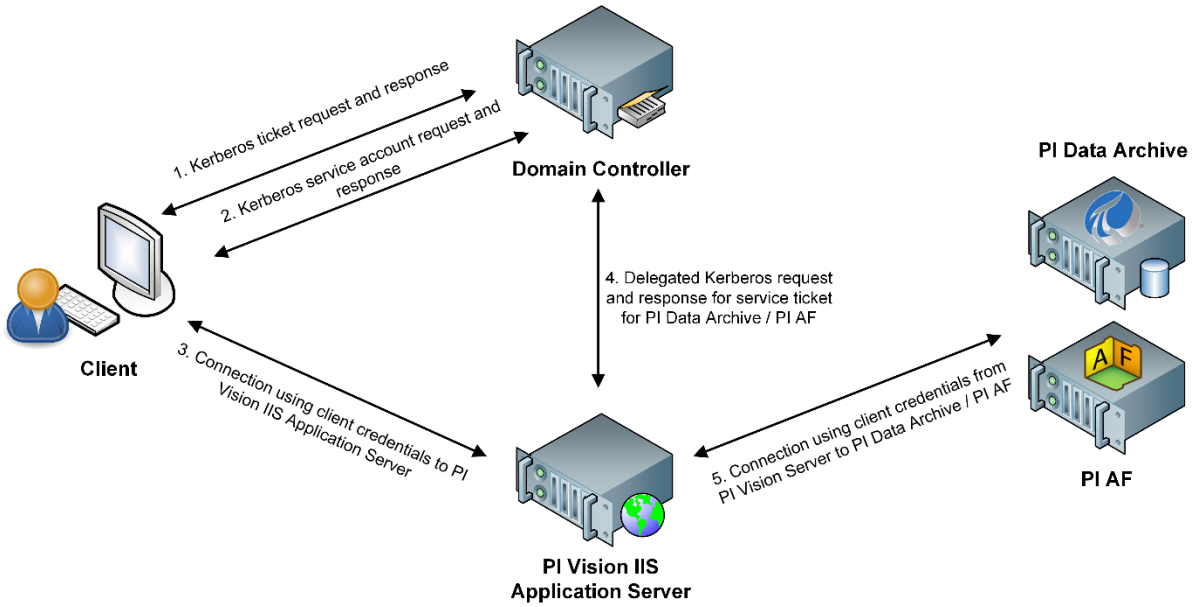
Kerberos ticket confirms the client credentials are validated and client is authorized to receive a service ticket. Service ticket provides access to application services.

*Step 2* – Client uses the Kerberos ticket to request a service ticket from domain controller to connect to PI Vision IIS Application Server.

*Step 3* – Client connects to PI Vision IIS Application Server and provides both Kerberos and service tickets.

*Step 4* – PI Vision IIS Application Server uses the client's Kerberos ticket to request a service ticket, so PI Vision IIS Application Server can connect to PI Data Archive \ PI AF.

*Step 5* – PI Vision IIS Application Server connects to PI Data Archive / PI AF using the client's credentials.




### 8.5.2 Kerberos Delegation Types

There are four types of Kerberos delegation:

- Unconstrained Delegation (not recommended)
- Constrained Delegation – Kerberos Only
- Constrained Delegation – Any Authentication Protocol
- Resource-Based Constrained Delegation

**Constrained** delegation is more secure because it limits delegation to a specified list of services, rather than allowing delegation to any service as in **unconstrained delegation**. It requires additional configuration compared with unconstrained delegation. **Service Principal Names** must be setup for accounts the services are running under.

**Kerberos only** means there is no protocol transition to a non-Kerberos authentication method.

**Any Authentication Protocol** allows protocol transition if Kerberos authentication fails.

These types of delegation are configured for the service that delegates to other services. In our case, PI Vision service (*PIVisionServiceAppPool*) delegates to *PI Network Manager* service of PI Data Archive and *PI AF Application Service*.

Setting up the delegation requires access to the Active Directory Users and Computers on the Domain Controller and Domain Admin privileges.

Setting up Kerberos delegation is described in *PI Vision Installation and Administration Guide* in chapter "Enable Kerberos delegation using a custom PI Vision service account" or in OSIsoft Live Library.

If standard **Domain User Accounts** are used for the services all four types of delegation can be configured.

If **Group Managed Service Accounts** (gMSA) are used for the services only Resource-Based Constrained Delegation is possible. This is case of PISCHOOL environment.

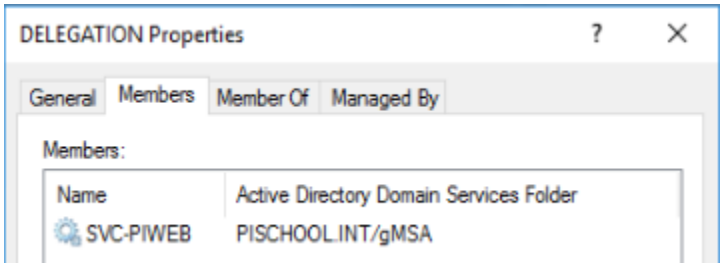
**Resource-based Constrained Delegation (RBCD)** introduced with Windows Server 2012 on the other hand has several benefits from the previous types:

- Is configured for the account of the resource service (*PI Network Manager*, *PI AF Application Service*) instead of the account of the service that delegates (*PIVisionServiceAppPool*)
- Domain Admin privileges are not required.
- Functions across domain and forest boundaries.

RBCD configuration can only be done via *PowerShell*. To setup the delegation for the account of the service we need to delegate to ***-PrincipalsAllowedToDelegateToAccount*** parameter is used. In our case for the PISCHOOL\SVC-PIDA$ and PISCHOOL\SVC-PIAF$ gMSA.

```
Set-ADServiceAccount -Identity SVC-PIDA -PrincipalsAllowedToDelegateToAccount
DELEGATION
Set-ADServiceAccount -Identity SVC-PIAF -PrincipalsAllowedToDelegateToAccount
DELEGATION
```

DELEGATION is the Domain Group type Security where **PISCHOOL\SVC-PIWEB$** gMSA is member of.




Setting up RBCD is described in *PI Vision Installation and Administration Guide* in chapter "Configure resource-based constrained delegation" or in OSIsoft Live Library.

### 8.5.3 Service Principal Name

A service principal name (**SPN**) is a **unique identifier** of a service instance. SPNs are used by Kerberos authentication to associate a service instance with a service logon account. This allows a client application to request that the service authenticate an account even if the client does not have the account name. Without the SPN the client cannot receive the Service ticket and Kerberos delegation is not possible.

To configure delegation for PI Vision the SPNs must be registered for these services:

- *PI Network Manager* for PI Data Archive running under **PISCHOOL\SVC-PIDA$**
- *PI AF Application Service* for PI AF running under **PISCHOOL\SVC-PIAF$**
- *PI Web API and PI Web API Crawler* services and IIS *PIVisionServiceAppPool*, *PIVisionAdminAppPool*, *PIVisionUtilityAppPool* application pools for PI Vision that will be running under **PISCHOOL\SVC-PIWEB$**

The SPN name format for host-based service is composed of two elements:

***<service class>/<host>***

<*service class*>: A string that identifies the general class of the service. For example, “SqlServer”. In general, this can be any string that is unique to the service class. Be aware that the SPN syntax uses a forward slash (/).

- **PIServer** is service class of PI Network Manager service of PI Data Archive
- **AFServer** is service class of PI AF Application Service
- **HTTP** is service class for web applications using HTTP or HTTPS protocol such as PI Vision, PI Web API and PI Web API Crawler.

<*host*>: The name of the computer on which the service is running. It can be FQDN or hostname.

SPN can be registered from any machine which is a member of the domain by user with appropriate permissions (such as Domain Admin or any user who was granted permissions)

To register an SNP use command line tool SETSPN in elevated command prompt and the following syntax:

***setspn -s <service class>/<host> <domain>\<service account>***

It is a good practice to register SPN for both <*host*> identifiers: hostname and FQDN.

Applicated to this course environment, where PI Data Archive is installed on PISRV01 an PI SRV02, PI AF is installed on PISRV01 and PI Vision will be installed on PISRV02; the registration commands are:

*setspn -s PIServer/PISRV01 PISCHOOL\SVC-PIDA$*
*setspn -s PIServer/PISRV01.pischool.int PISCHOOL\SVC-PIDA$*
*setspn -s PIServer/PISRV02 PISCHOOL\SVC-PIDA$*
*setspn -s PIServer/PISRV02.pischool.int PISCHOOL\SVC-PIDA$*
*setspn -s AFServer/PISRV01 PISCHOOL\SVC-PIAF$*
*setspn -s AFServer/PISRV01.pischool.int PISCHOOL\SVC-PIAF$*
*setspn -s HTTP/PISRV02 PISCHOOL\SVC-PIWEB$*
*setspn -s HTTP/PISRV02.pischool.int PISCHOOL\SVC-PIWEB$*

To verify the SPNs are registered correctly use *SETSPN -L <domain>\<service account>.*

*setspn -L PISCHOOL\SVC-PIDA$*
*setspn -L PISCHOOL\SVC-PIAF$*
*setspn -L PISCHOOL\SVC-PIWEB$*

```
C:\Users\student01.PISCHOOL>setspn -l PISCHOOL\SVC-PIDA$
Registered ServicePrincipalNames for CN=SVC-PIDA,OU=gMSA,DC=PISCHOOL,DC=INT:
        PIServer/PISRV01
        PIServer/PISRV01.PISCHOOL.INT
        PIServer/PISRV02
        PIServer/PISRV02.PISCHOOL.INT

C:\Users\student01.PISCHOOL>setspn -l PISCHOOL\SVC-PIAF$
Registered ServicePrincipalNames for CN=SVC-PIAF,OU=gMSA,DC=PISCHOOL,DC=INT:
        AFServer/PISRV01.pischool.int
        AFServer/PISRV01

C:\Users\student01.PISCHOOL>setspn -l PISCHOOL\SVC-PIWEB$
Registered ServicePrincipalNames for CN=SVC-PIWEB,OU=gMSA,DC=PISCHOOL,DC=INT:
        HTTP/PISRV02.pischool.int
        HTTP/PISRV02
```

PI Data Archive and PI AF are capable to register the SPNs by themselves if their service account is granted the permissions using this PowerShell command:

```
$gMSA = Get-ADServiceAccount -Identity SVC-PIDA
dsacls $gMSA.DistinguishedName /G "SELF:RPWP;servicePrincipalName"
$gMSA = Get-ADServiceAccount -Identity SVC-PIAF
dsacls $gMSA.DistinguishedName /G "SELF:RPWP;servicePrincipalName"
```

For more detailed information about Service Principal Names beyond the scope necessary for this course visit Microsoft Docs.

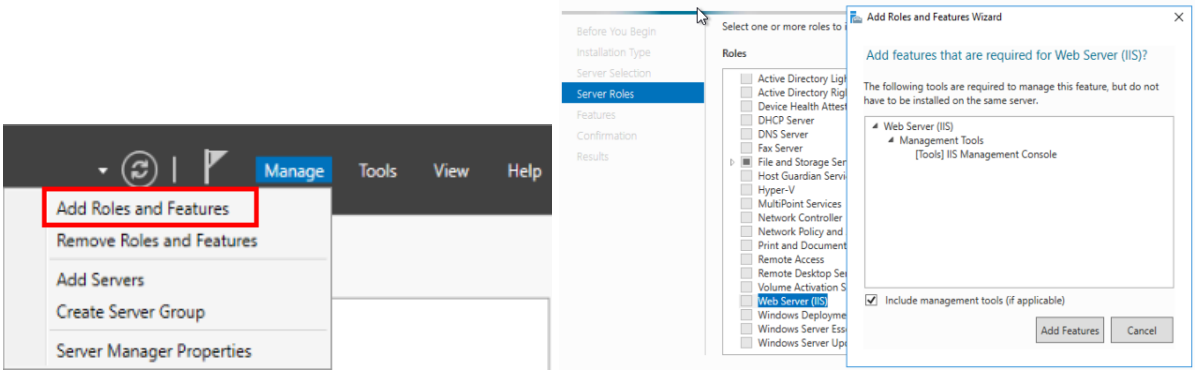
## 8.6 IIS Application Server preparation

PI Vision install kit requires **Web Server (IIS) role** to be installed prior the installation.

Adding the server role is done via Server Manager application. **Manage → Add Roles and Features**.

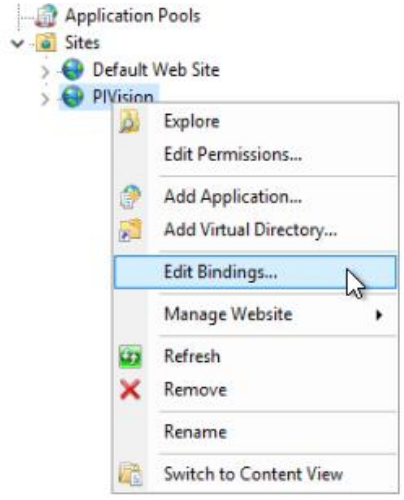
It is not necessary to add more features beside the default selection that includes management tools (IIS Management Console). PI Vision installation procedure triggers a script that installs all necessary features and role services for PI Vision.

The least minimum role services are listed in *PI Vision Installation and Administration Guide* in section “Prepare the PI Vision application server computer – Add server roles and features”. Or in OSIsoft Live Library.

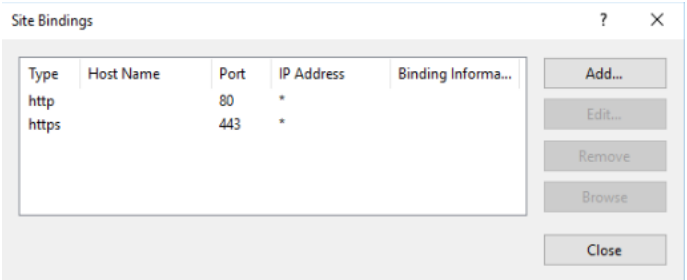



After Web Server (IIS) role is installed Internet Information Services (IIS) Manager is in **Start Menu → Windows Administrative Tools**.

If the Web Server is dedicated only to PI Vision application, then it is possible to install it to the Default Web Site, but if the Web Server is shared with other web applications, third-party or OSIsoft’s PI Manual Logger Web and RtReports Generator, it is recommended to install PI Vision to the dedicated website.




For new website it is necessary to assign bindings and their ports. HTTP protocol has default port 80 and HTTPS default port 443. If those ports are taken by another website, they can’t be used, and different port number must be assigned. Any other number must be specified in URL address.



## 8.7 Directed Activity – PI Vision Installation

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

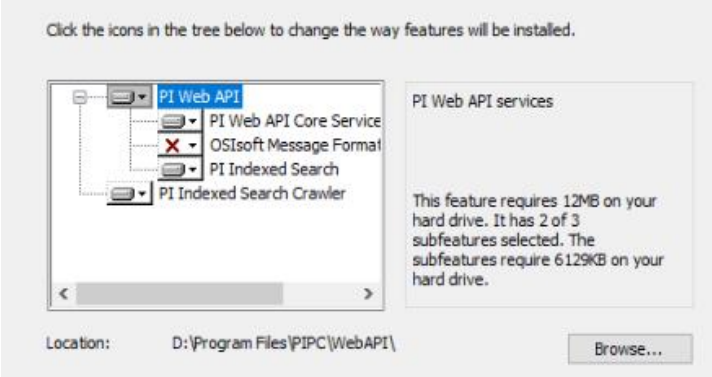
- Install PI Vision and PI Web API
- Install PI ProcessBook to PI Vision Migration Utility

### Problem Description

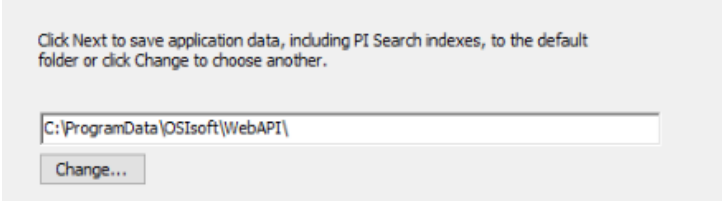
Kerberos delegation is set up and IIS application server is prepared. Install and configure PI Vision on PISRV02, so you will be able to visualize data from PI points that you created previously.

### Approach

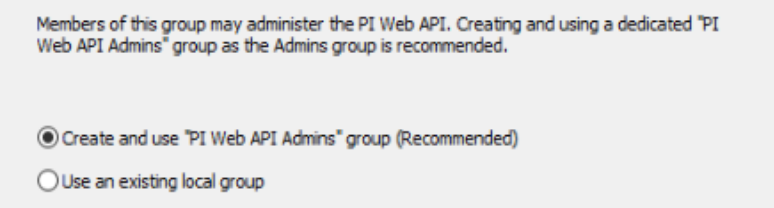
1. On PISRV01 open PI SMT and navigate to **Security** → **Identities, Users & Groups** and create new PI Identity and call it **PI Vision**. Map it to the **PISCHOOL\SVC-PIWEB$** gMSA.
2. Navigate to **Security** → **Database Security** and assign *Read Access* to PI Vision identity in following tables: **PIDBSEC**, **PIMAPPING**, **PIPOINT** and **PIUSER**.
3. On PISRV02 run PI Vision install kit located in PI Install Kits folder.
4. In PI Web API install section keep the default selection and location and click **Next**.



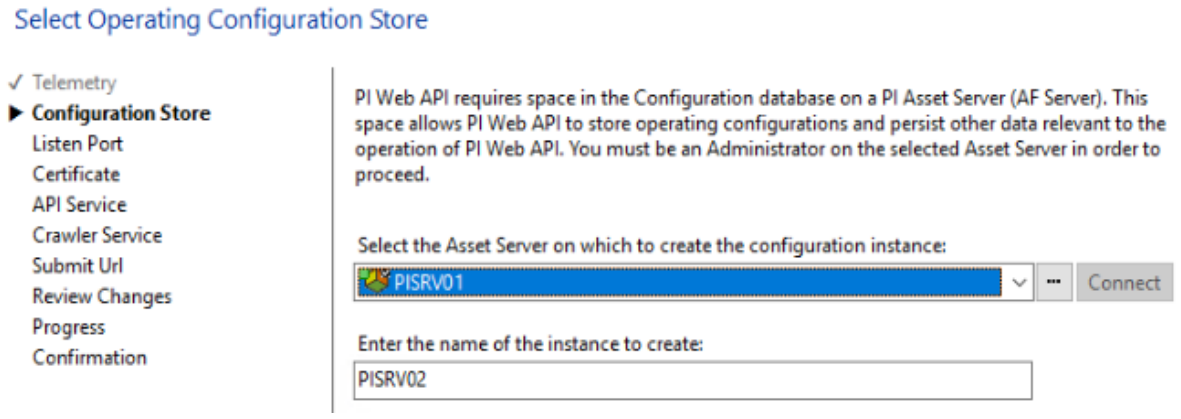

5. On the next screen where you can choose the location of search indexes keep the default path *%ProgramData%\OSIsoft\WebAPI* and click **Next**.



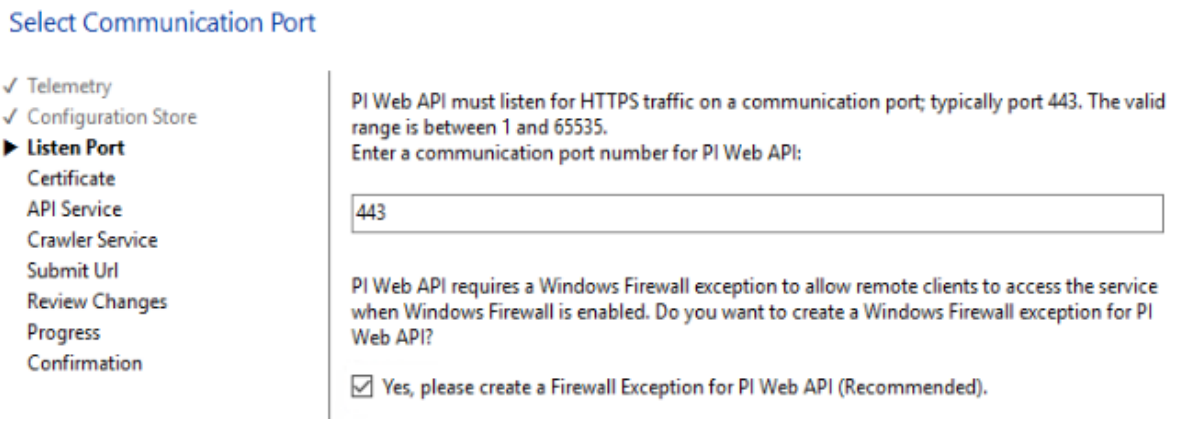
6. On the next screen confirm the option Create and use “PI Web API Admins” group click **Next** and **Install**.



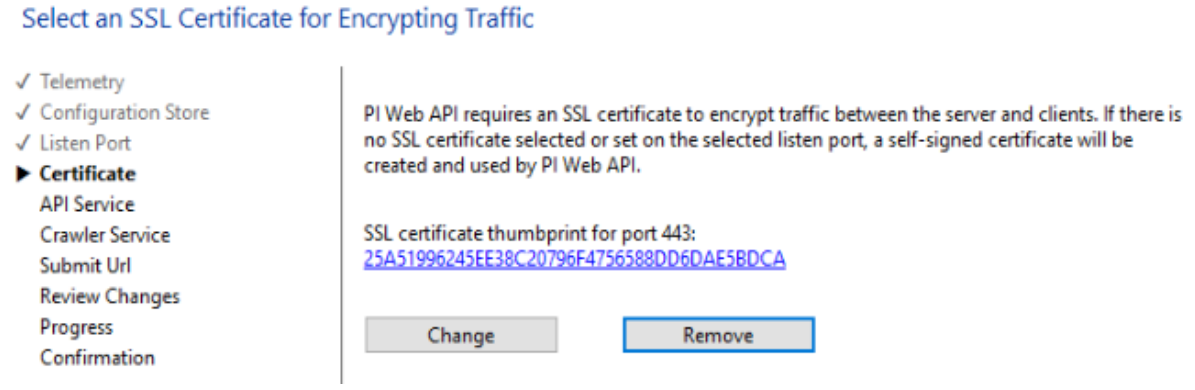

7. Installation of PI Web API is complete and click **Finish**. Do not continue with PI Vision installation, because this will open **PI Web API Admin Utility** to finish the PI Web API configuration.
8. Uncheck the participation in PI System Customer Experience Improvement Program. Click **Next**.
9. The PI Web API configuration is stored in PI AF Configuration database. Click on **Connect** button to test connection to PISRV01 AF server. Connection to PI AF must be verified to enable **Next** button.



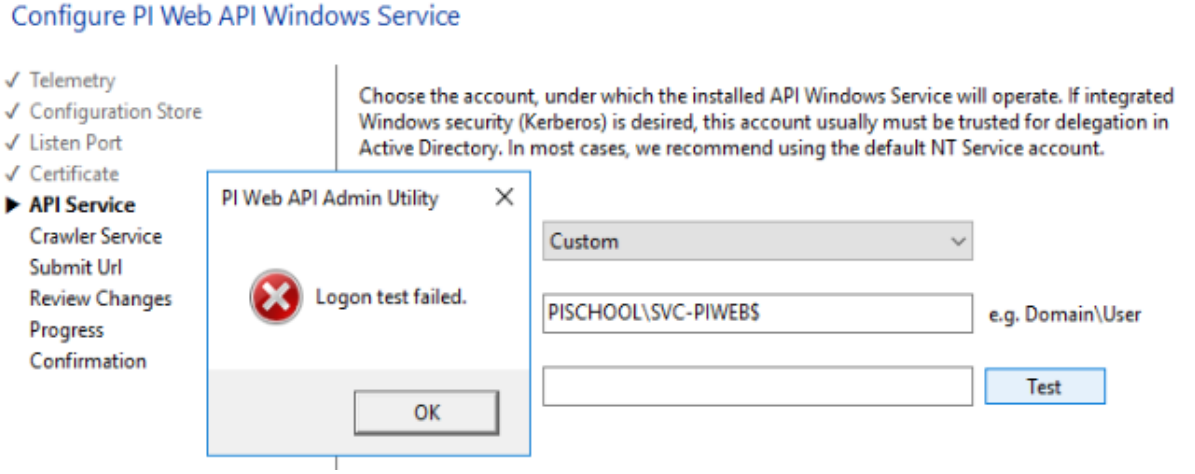

10. PI Web API listen for HTTPS traffic on communication port. Keep the default port 443 and adding a firewall exception. Click **Next**.



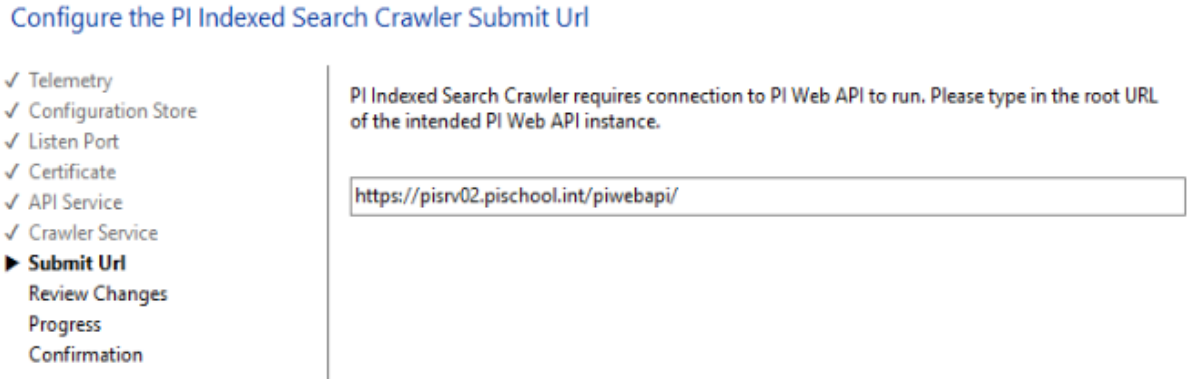
11. Because the certificate was already assigned to HTTPS binding in PIVision website configuration in IIS Manager it is now pre-selected. Otherwise the selection is done here. If no certificate is available, Self-signed certificate will be created. Click **Next**.



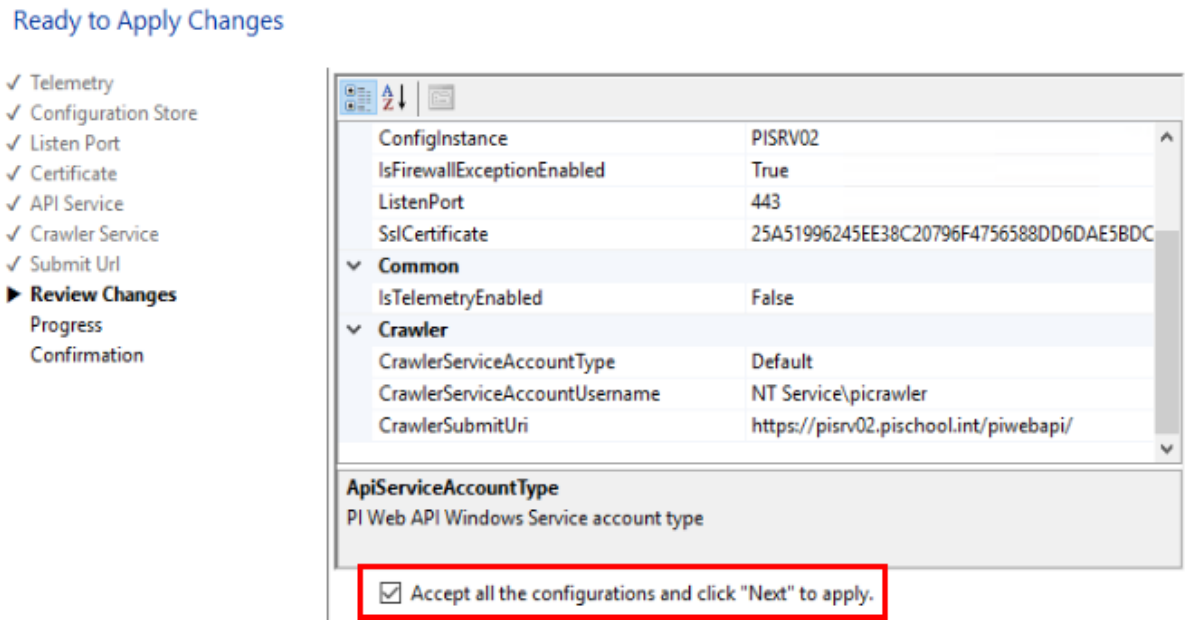

12. Next screen is the PI Web API and PI Web API Crawler services account selection. PI Web API Admin Utility is not capable to accept gMSA. Logon test will fail. Keep the selection of default *NT Service\piwebapi* and *NT Service\picrawler* accounts.




13. For PI Web API Crawler keep the default URL *https://pisrv02.pischool.int/piwebapi*. Click **Next**.

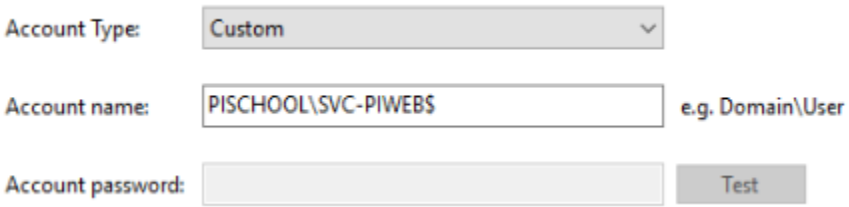


14. The final screen is the summary before the changes are applied. **Check** Accept all the configurations and click **Next**.

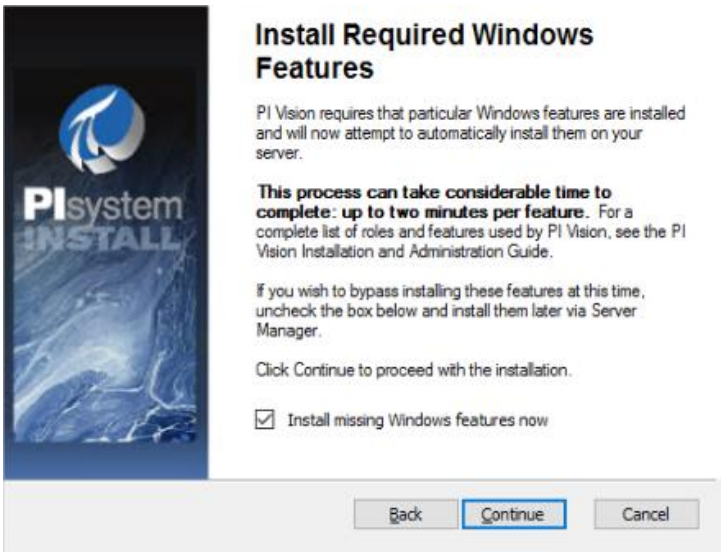



15. Configuration is now stored with custom accounts. To apply gMSA click **Finish** to close the PI Web API Admin Utility.
16. Open Services Manager and locate PI Web API and PI Web API Crawler services and switch default accounts for gMSA **PISCHOOL\SVC-PIWEB$**. Do no restart them!

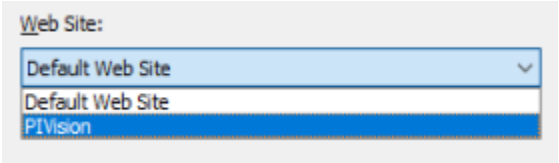

17. Changed the account in Services Manager is not enough for PI Web API. The change must be applied to the PI AF Configuration database. In order to do this, run PI Web API Admin Utility located in **Start Menu** → **PI System** folder again.
18. Keep clicking on **Next**. In account selection for PI Web API and PI Web API Crawler services gMSA are now selected.



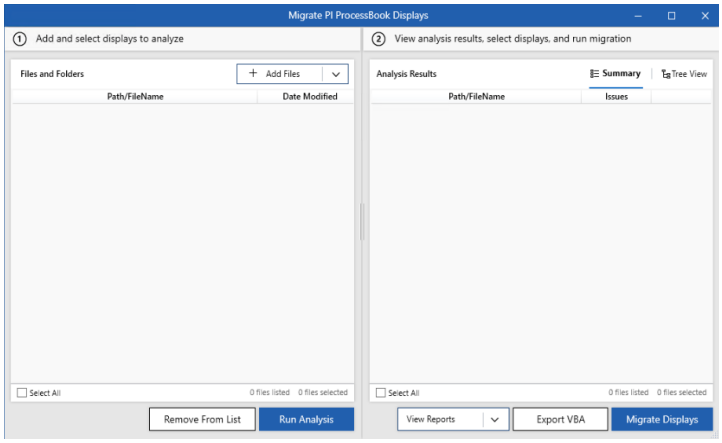

19. On summary page again check to accept all the configurations and click **Next** to apply. After all jobs complete successfully click **Finish.**
20. Now continue with PI Vision installation. Click **Next** and acknowledge compliance.
21. PI Vision install kit installs missing Windows features if necessary. Keep the “Install missing Windows features now” checked and click **Continue** and **OK**.




22. Select **No** to participation in User Experience Initiative and click **Next**.
23. Because another PI System application is already installed on PISRV02, PI Vision application directory selection is locked to %PIHOME64%. Click **Next**.
24. For IIS site selection select **PIVision** site from drop-down menu instead of Default Web Site. Click **Next** and **Install**.




25. Once the installation is complete. Click **Finish** and then agree with computer reboot.
26. After logging back close the installation dialog and run **PI ProcessBook to PI Vision Migration Utility** install kit from PI Install Kits folder.
27. Keep the default location %PIHOME% (D:\Program Files (x86)\PIPC). Click **Next** → **Install** → **Finish**.
28. Verify the installation by running the utility from newly created desktop shortcut.



## 8.8 Directed Activity – PI Vision Configuration

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

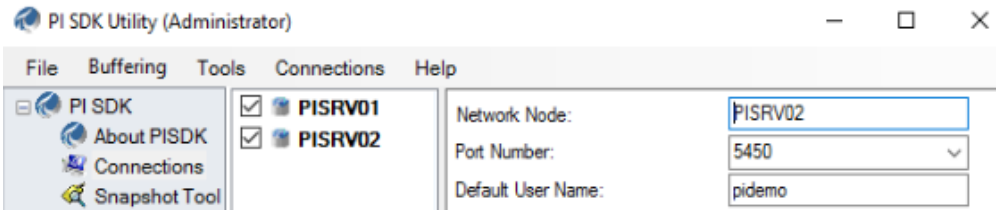
- Configure PI Vision after installation
- Create a simple display to visualize data from default or Pumps PI points

### Problem Description

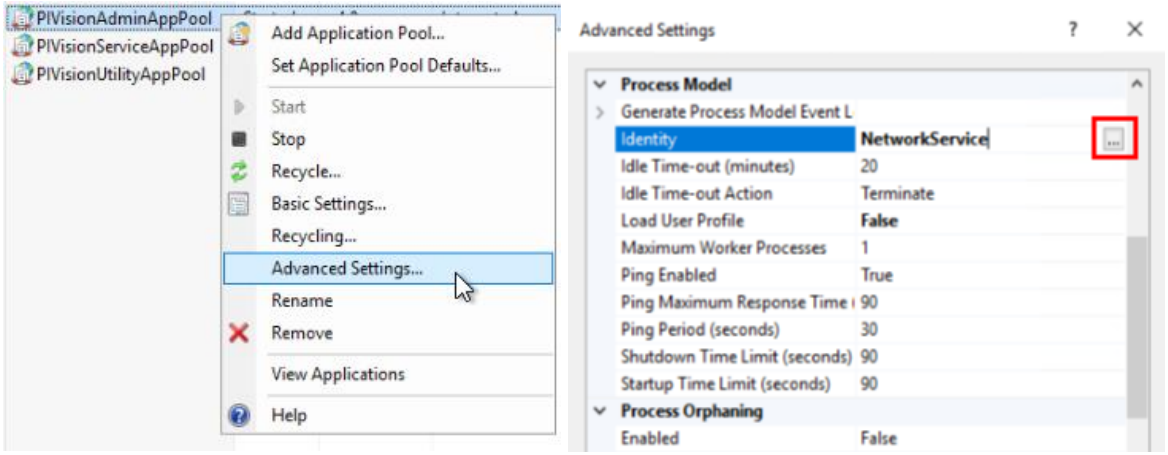
PI Vision is installed, but you are not done yet. Finnish the configuration in IIS Manager and PI Vision Admin website. For verification created a simple display to visualize the data from PI Points.

### Approach

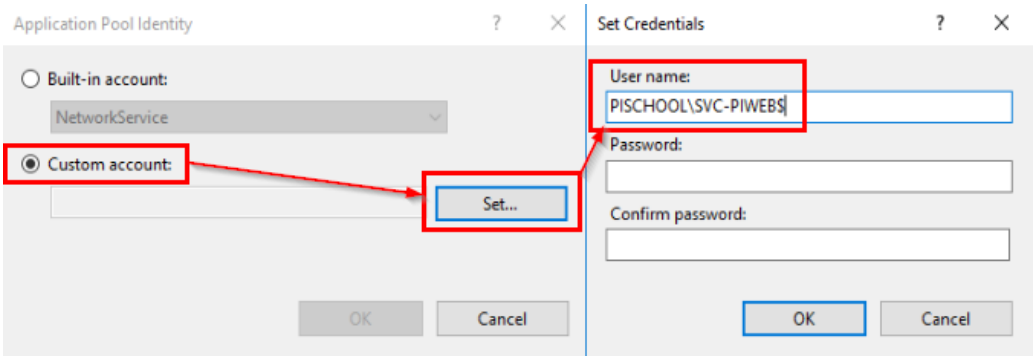
1. On PISRV02 open PI SDK Utility and add PISRV01 PI Data Archive server to the list.




2. Open IIS Manager from Start Menu. Expand the menu and select Application Pools.
3. Right-click on *PIVisionAdminAppPool* and go to **Advanced Settings**.
4. From Process Model section select Identity and click on […]



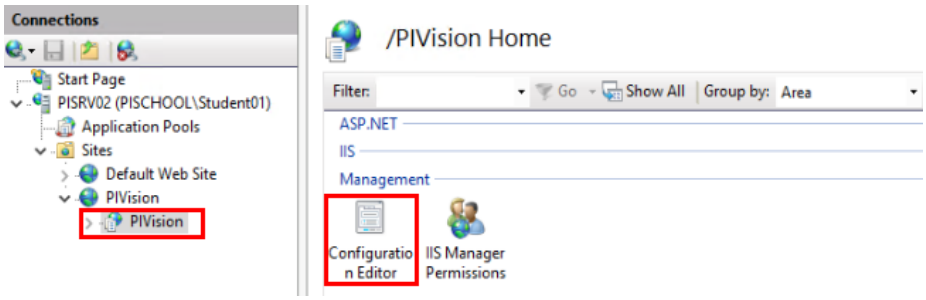

5. Select Custom account and click Set…. Insert **PISCHOOL\SVC-PIWEB$** gMSA and confirm.



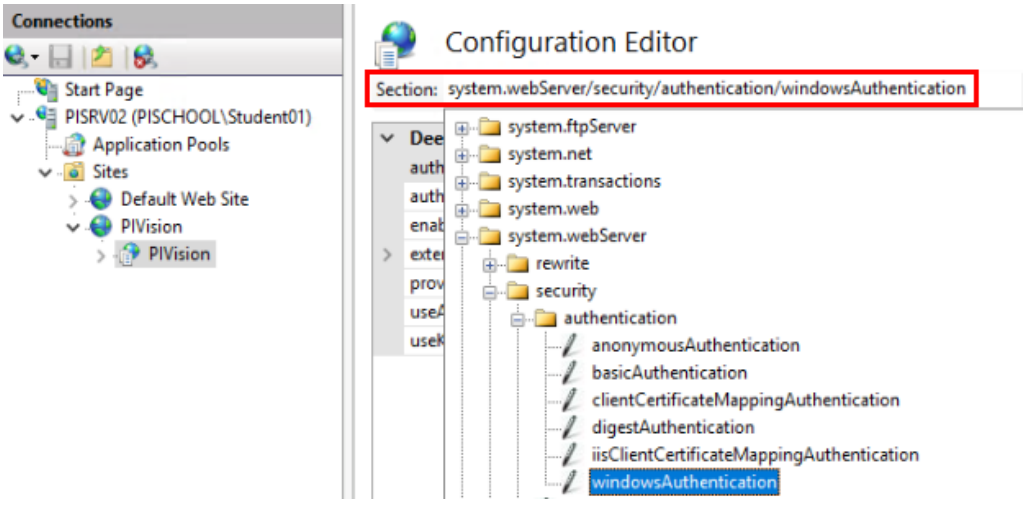

6. Repeat the steps for *PIVisionServiceAppPool* and *PIVisionUtilityAppPool*.

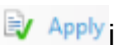
| | | | | |
|---|---|---|---|---|
| PIVisionAdminAppPool | Started | v4.0 | Integrated | PISCHOOL\SVC-PIWEB$ |
| PIVisionServiceAppPool | Started | v4.0 | Integrated | PISCHOOL\SVC-PIWEB$ |
| PIVisionUtilityAppPool | Started | v4.0 | Integrated | PISCHOOL\SVC-PIWEB$ |

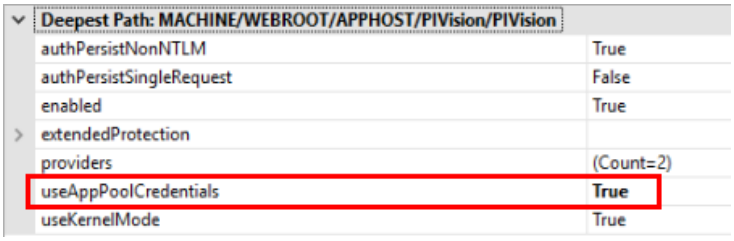
7. To apply the new identity the Application Pools must be recycled by right click → Recycle or Stop and Start.
8. Expand Sites and PIVision website in navigation tree. Click on PIVision application. From Management section at the bottom, select **Configuration Editor**.




9. In Section field navigate to:
***system.webServer/security/authentication/windowsAuthentication***

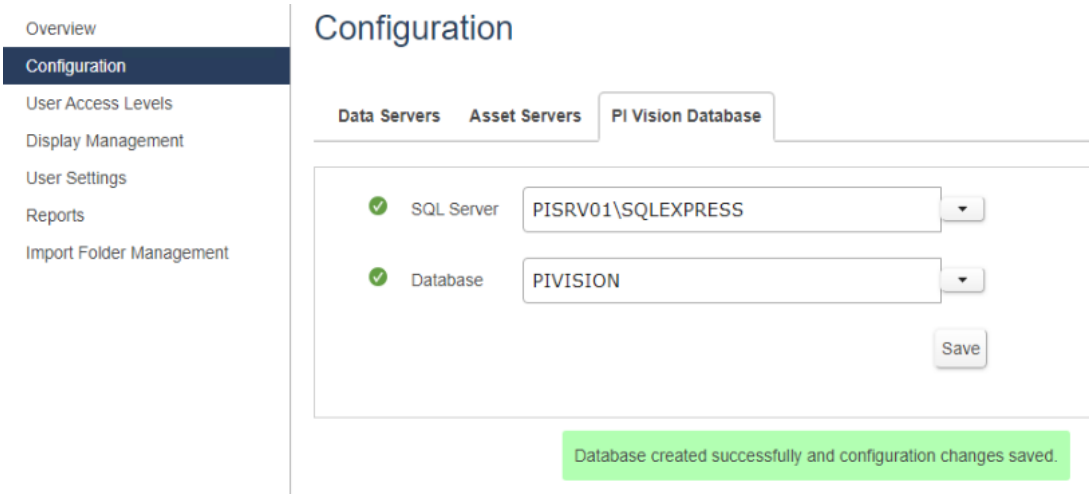


10. Set usAppPoolCredentials from False to **True** and hit Apply  in Actions pane.

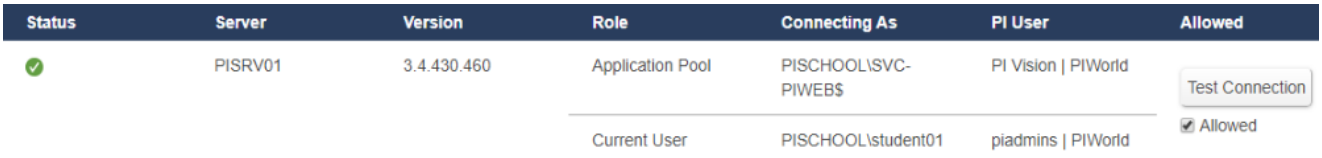



11. Or completely restart Internet Information Services by **IISRESET** command from elevated command line.

12. Open Google Chrome and enter URL https://pisrv02/PIVision/Admin. Go to Configuration and select PI Vision Database tab. For SQL Server enter **PISRV01\SQLEXPRESS** and name the database **PIVISION**. Click **Save**. This creates SQL database for PI Vision on MS SQL Server.



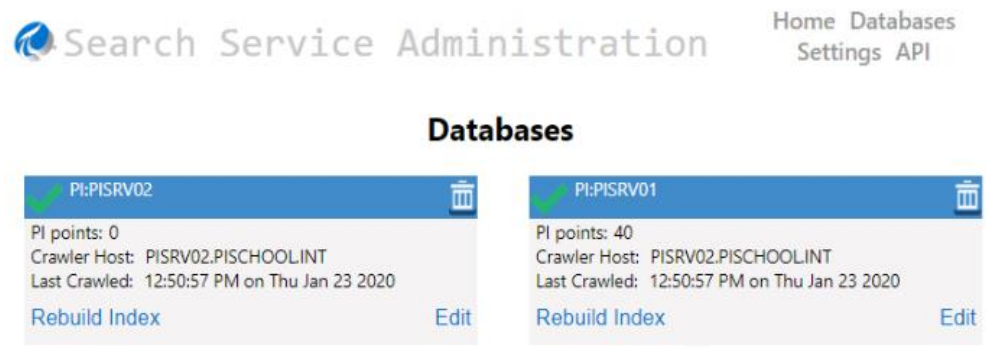

13. Select Data Servers tab. Click on Test Connection for PISRV01. After green check mark check **Allow** and click on **Save**. Do not allow PISRV02 as it is not necessary now.




14. No configuration necessary in Asset Servers tab now as we do not have any AF Database available, yet.

15. On the Overview page there should be only green check marks . Pay attention to the "**Indexed?**" column. PI Web API Crawler must build search indexes. Depending on the size of the PI System (# of PI points, AF elements and attributes) the process can take from seconds up to ten or more minutes.

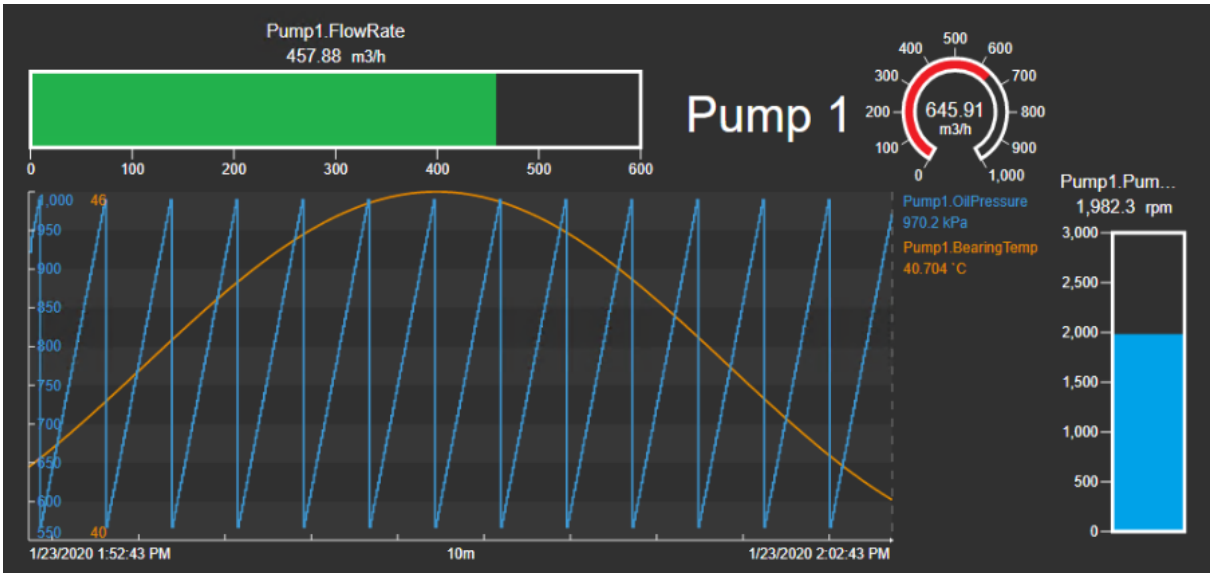
16. To check the status of the process, enter URL https://pisrv02.pischool.int/piwebapi/admin/search/database.html.



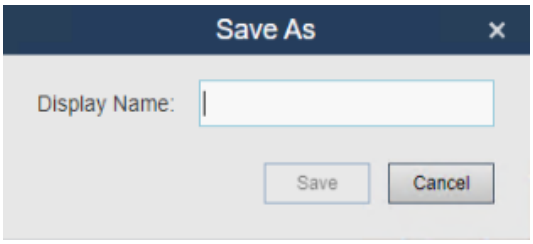

17. Configuration is now complete. To test everything works access PI Vision from client machine to have the Kerberos double hop scenario. Use PIINT02. Enter URL https://pisrv02/pivision to the web browser.

18. From the PI Vision main page create new display 

. Search for any PI points, either default points or Pumps points an place differrent object to the canvas. Verify you can see values.




19. Save the display using by clicking on Save  and name it as you like.



# 9. Data Flow

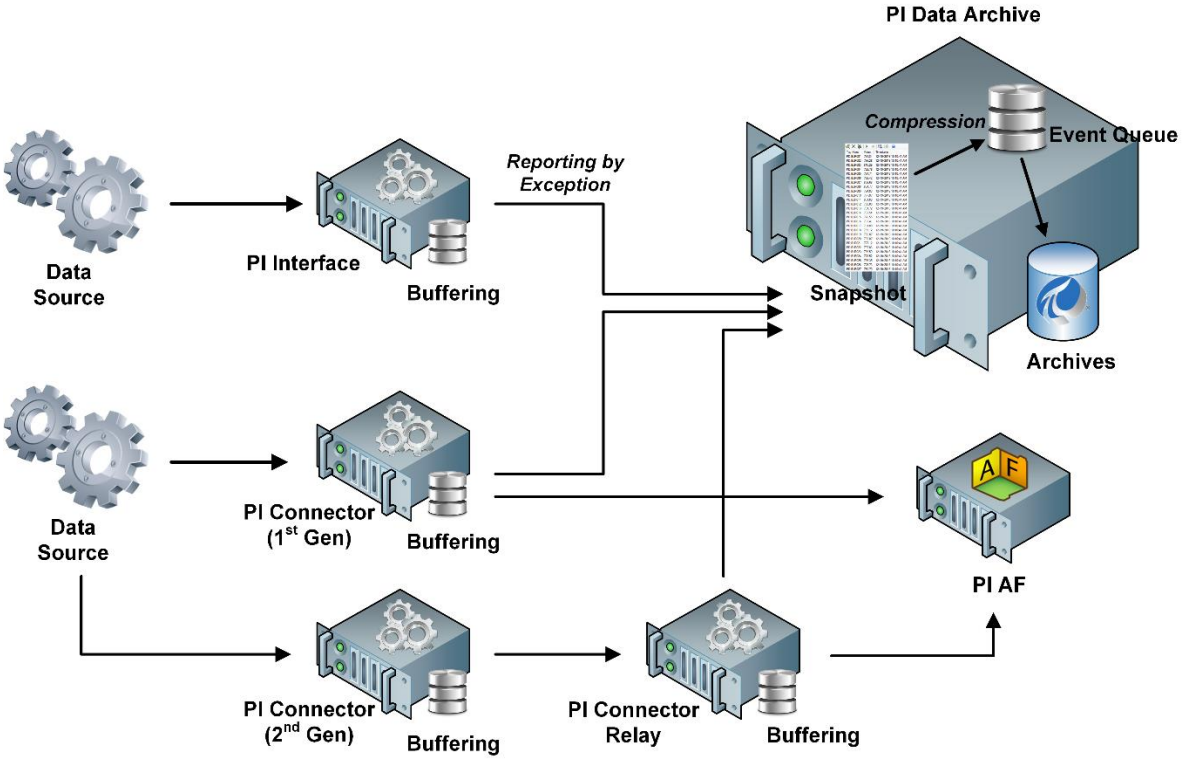
- Describe the data flow through the PI System
- Describe reporting by exception
- Describe the compression test
- Describe the data structures in the PI Data Archive

## 9.1 Data Flow: From the Data Source to the PI Server

In order to determine that everything was installed and set up correctly (and checking for proper function in the future), it is key that you understand the various data structures or "touch points" that the data encounters along the way.

The PI Data Archive stores data in the form of events. ***Each event has a value and a timestamp*** that indicates what time the value was collected.

PI Interfaces collect data from the data sources and typically use exception reporting, meaning that they pass significant events on to the PI Data Archive and discard the rest. If the buffering service is configured on the data collection node, then the events go through the buffering service. If the interface node cannot connect to the PI Data Archive, the PI Buffer Subsystem service holds the data until the PI Data Archive connection is restored.



PI Connectors (1st Gen & 2nd Gen) collect data and timestamps from the data sources, but they do not timestamp the data and do not perform the exception reporting. All PI Connectors have their own internal buffering which is separate from buffer for PI Interfaces. PI Connector's buffer is capable of buffering information for PI point and AF asset creation.

PI Connectors (2nd Gen) do not connect directly to PI Server but send data to PI Connector Relay which has another internal buffering, same type as PI Connector and values are then forwarded to PI Server.

## 9.2 Data Flow on the Data Collection Computers

As described earlier, the **PI Interface** has these basic functions:

1. Collects data
2. Timestamps the data (or validate timestamp provided by the source device)
3. Applies the Exception Deviation
4. Passes the values to PI Buffer Subsystem (if not configured then sends data directly to PI Data Archive)

Applying the exception parameters is referred to as "Reporting by Exception."

**PI Connector (1st Gen)** has these basic functions

1. Crawls the data source for information about data and their structure
2. Creates the appropriate points in PI Data Archive and asset structure in PI AF
3. Collects data
4. Checks for updates on the data source to update it on PI Server side

**PI Connector (2nd Gen)** has these basic functions

1. Crawls the data source for information about data and their structure
2. Passes the points and asset structure configuration to PI Connector Relay
3. Collects data
4. Checks for updates on the data source and passes the updates to PI Connector Relay

**PI Connector Relay** has these functions:

1. Connects to PI Data Archive and PI AF
2. Creates points and asset structure based on the configuration received from PI Connector (2nd Gen)
3. Forwards values updates received from PI Connector (2nd Gen)

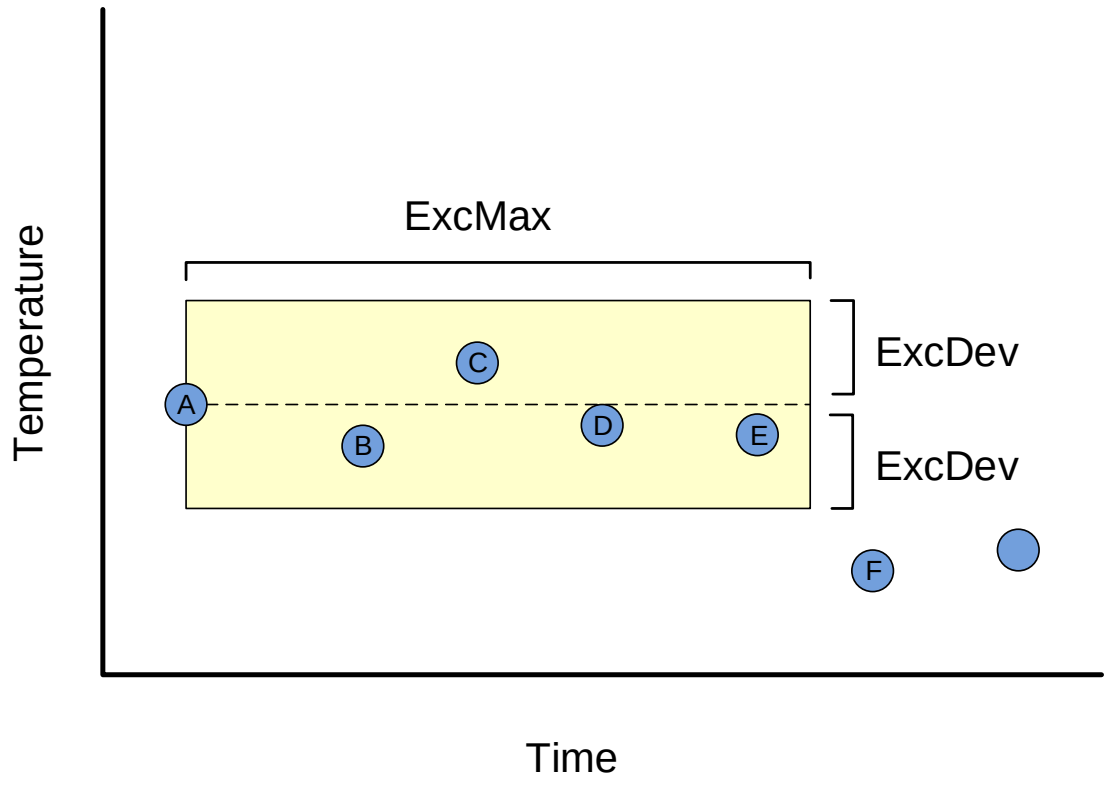
## 9.3 Reporting by Exception

The object of **exception reporting** is simply to reduce noise. In other words, PI Interface should send you the data you are interested in, rather than overloading the network connection by sending data that is not meaningful. It is up to the user to decide the terms for "meaningful".

Exception reporting uses a simple dead band algorithm to determine whether to send events to the PI Data Archive. For each point, you can set exception reporting specifications that create the dead band. The PI Interface ignores values that fall inside the dead band.

The dead band is expressed as a deviation and is applied equally in positive and negative manner. There is also a maximum time applied.

As mentioned earlier, PI Connectors do not perform exception reporting.




In the above illustration, values A, E, and F are reported to the PI Data Archive. Value A is the last reported value, values B, C, D and E fall within the exception dead band, but value F falls outside the dead band, so the interface reports value F and the its previous value, in this case, E.

These exception parameters are set on a per point basis.

**Note 1:** Some PI Interfaces do not support exception reporting. You know enough by now to check the documentation.

**Note 2:** ExcMin is typically set to 0. It is rarely used

## 9.4 Group Questions – Data Flow

The following questions are intended to reinforce key information, or discover a new insight. Your instructor may choose to have you try answer the questions on your own or have the group answer the together aloud.

**Questions**

1. Why would we apply an Exception Maximum (ExcMax)?

2. How would you “turn OFF” reporting by exception?

3. Give at least one valid reason why you would turn OFF exception processing for a point.

## 9.5 Snapshot

The Snapshot Table is simply the “current” or most recent value for each point in the PI Data Archive.

The Snapshot Subsystem populates this table and performs the Compression Algorithm Calculation. (As it will be explained later, in PI Buffer Subsystem for PI Interfaces, compression is performed back on the data collection machine).

When a new value is received, it is compared to the previous value.

- If that value indicates that the previous value passes compression, then the previous value is ***sent on*** and the new value is retained as the new “current” value.
- If that value indicates that the previous value fails compression, then the previous value is ***discarded,*** and the new value is retained as the new “current” value.

The **Data** → **Current Values** plug-in in PI SMT is the view of that snapshot table.

## 9.6 Directed Exercise – PI Snapshot Values

You are invited to watch what the instructor is doing or perform the same steps at the same time to explore the different concepts presented in this chapter or section.

### Activity Objectives

- Determine which events from PI Interface will make it to the Snapshot in PI Data Archive.

### Approach

Consider the following attribute values for a PI point:

| | |
|---|---|
| **ExcDevPercent** | **2%** |
| **ExcMax** | **180 seconds** |
| **Span** | **200 units** |
| **Zero** | **0 units** |

The current snapshot received in the PI Data Archive for this point is:

Value: 70.3 Timestamp:10:00:00

Which of the following values collected by the PI Interface pass the exception test?

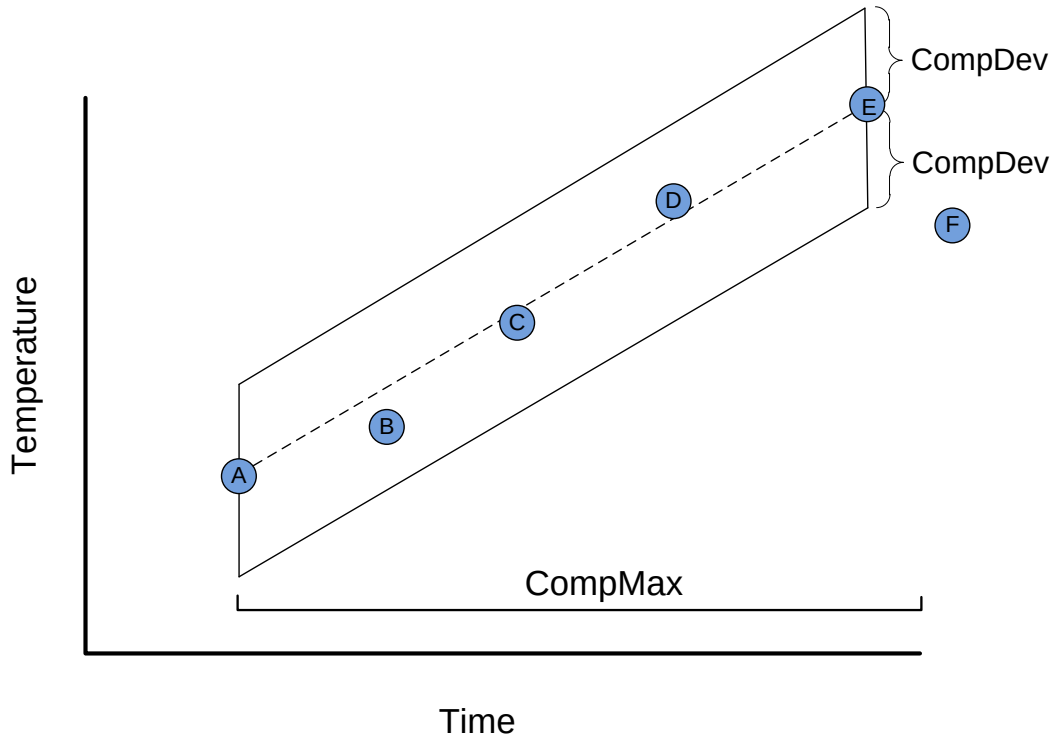
| PI Interface Node | | PI Data Archive Node | |
|---|---|---|---|
| **Time** | **Value** | **Snapshot Time** | **Current Snapshot** |
| 10:00:00 | 70.3 | 10:00:00 | 70.3 |
| 10:01:00 | 67.1 | | |
| 10:02:00 | 71.4 | | |
| 10:03:00 | 70.1 | | |
| 10:04:00 | 68.2 | | |
| 10:05:00 | 66.0 | | |
| 10:06:00 | 65.8 | | |
| 10:07:00 | 64.2 | | |
| 10:08:00 | 60.0 | | |
| 10:09:00 | 63.1 | | |

## 9.7 Compression

The point of compression testing is to store just enough data to reproduce the original data from the data source, within the limits of accuracy required. It is up to the user to decide the limits of accuracy.

The compression process applies a deviation in a similar manner to exception ***except that it considers the slope of the data.***

In the following illustration, all the events fall around the same straight line. In a simple case like this, there is no need to store all the points on the line. If you store just two points, you can recreate the point value for any other time, with the accuracy of ±CompDev.

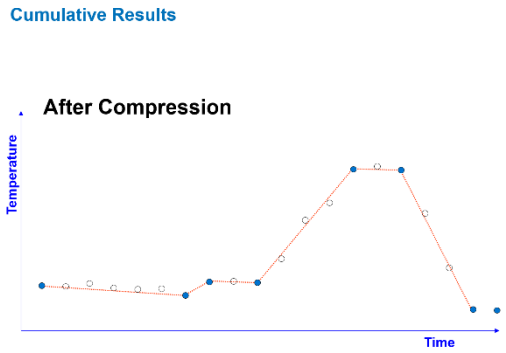



The same principle applies to compressing real-world data. PI Data Archive uses a sophisticated compression algorithm to determine which events it needs to keep in order to provide an accurate data history. The **CompDev** and **CompMax** attributes allow you to control the granularity of the compression algorithm.

Like the exception parameters, these **compression** parameters are also set on a per point basis.

## 9.8 Compression Test, Cumulative Impact and Defaults

There is a cumulative impact of the exception and compression process. This is illustrated in the slide shown below:




The animation in slide is not 100% correct how Compression works and it should only present the algorithm in a simple demonstration. For more detailed info please see KB00699 – Compression Explained

**Tip**

If you create a point in PI Data Archive and do not specify values for exception and compression specifications, the default values will be used. This should be avoided because the exception and compression values for each point should correctly reflect the desired point values. PI Connectors create points with the default values, which can be changed later, but exception attributes are ignored.

The default values for exception and compression are as follow:

| | |
|---|---|
| **ExcDevPercent** | 0.1 (% of span) |
| **ExcMax** | 600 (10 minutes) |
| **CompDevPercent** | 0.2 (% of span) |
| **CompMax** | 28800 (8 hours) |
| **Zero** | 0 |
| **Span** | 100 |

If you turn exception and compression off, you may adversely affect the performance of the system by archiving many more values than necessary. For example, a valve scanned every few seconds. With no exception or compression, the value of *OPEN* would be recorded thousands of times unnecessarily. While disk space is cheap, the more limiting factor is the speed of retrieval. Disk latency and network bandwidth limitations may impede performance when retrieving data

As a starting point recommendation for these settings, we recommend setting the compression deviation (**CompDev**) to the minimum change measurable by the instrument. The exception deviation (**ExcDev**) should be set to half of the compression deviation. It is important to note that these are only starting point recommendations and you should inspect your data for the desired resolution. In some cases, it may be advisable to turn off exception and compression entirely. To achieve this, set the exception deviation (**ExcDev**) and the exception maximum (**ExcMax**) to *0*. Turn off compression directly (**Compressing** set to *Off*), although it is recommended to leave compression *On*, and **Compdev** at 0. If set properly, PI Data Archive will archive values that reflect an accurate change in the device, without wasting space on duplicating values or losing meaningful values.

## 9.9 PI Archives

The data tables that the PI System stores data in files that are called “Archives.” Archives have the following characteristics:

- Starting from PI Data Archive 2015, Archives are either Historical or Future
- Historical Archives are fixed sized files
- Future Archives are dynamical sized files
- Archives must be registered
- The PI Data Archive only writes current values to one file at a time
- Historical Archives are sequential in time
- An archive has all the data for every point in the system for the period bounded by its start and end times.
- Each archive has an associated Annotation file

**Historical Archive files vs. Future Archive Files**

With the addition of Future Data, PI Data Archive has two types of archive files: Historical archives and Future archives. The historical archive files are only used to store historical data while the future archive files can store both future and historical data.

Data in future archives is never mixed or interchanged with data in historical archives. When time passes, and future data moves into the past, it is still stored in the same future archives.

**Fixed Files vs. Dynamical Files**

When you create historical archives, they are created of a fixed size and the memory is allocated upon creation to minimize the potential of fragmentation on disk.

You have the option of creating dynamic archives. Dynamic archives are files that grow as they are filled. ***Dynamic archives should only be used for backfilling purposes.***

Full fixed archive file will automatically become dynamic when you backfill data.

When you create future archives, they are created with an initial size of 1MB. If the data stored in that archives ever exceed 1MB the future archive grows dynamically to store the extra data.

**Archives Must Be Registered**

For PI Data Archive to access the data in an archive, the archive must be registered (often referred to as “mounted” in other systems). The archives can be located on any drive available to PI Data Archive, if there is adequate bandwidth. But OSIsoft recommends to store archives on local drives and use network drives only for backup or as a location for the oldest archives.

**Future Archives with non-sequential data**

Future archives are optimized for non-sequential data unlike real-time data stored in the historical archives. This way future archive is only created when necessary.

Every future archive file has a pre-determined time range of 1 month but can be created manually for longer period.

# 10. PI Buffer Subsystem

**Objectives**

- Describe PI Buffer Subsystem
- Configure PI Buffer Subsystem
- Validate the functionality of PI Buffer Subsystem

Throughout this section, you will refer to various sections in the *Buffering User Guide*. Please open this document.

## 10.1 What is PI Buffer Subsystem?

Data sent from the PI Interface to the PI Data Archive are redirected to the buffering process, which stores and forwards events to the home node.

If data flow to any of the associated PI Data Archives is interrupted, then data are buffered to a queue file. When the buffering process is shut down, the data are still located in the queue file. Using FIFO (first-in, first-out), the data are then flushed to the PI Data Archive. Multiple PI Interfaces on a single data acquisition computer share the same buffering process and store data in the same buffer queue file.

When connection is reestablished to the associated PI Data Archives the buffers are flushed chronologically until all data has been transferred.

Tip

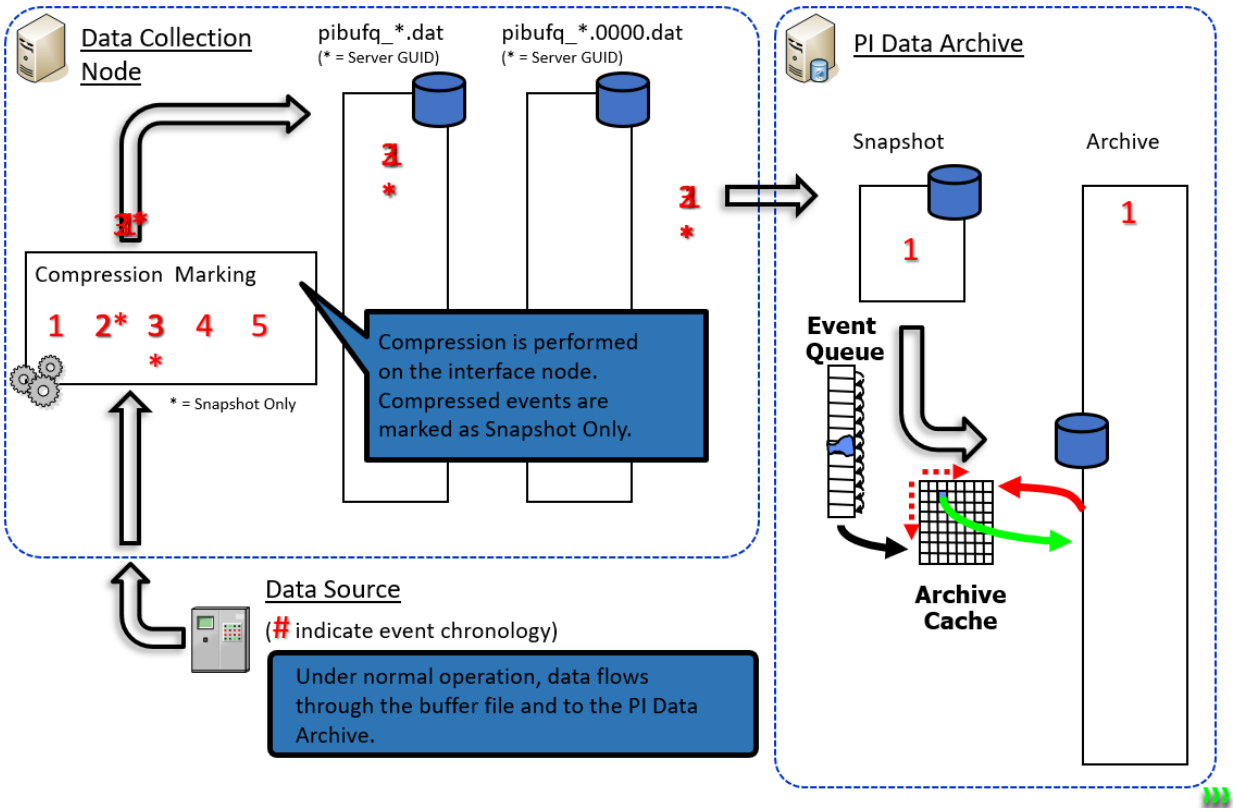
Unless you have a specific reason why you do not want to preserve your data, you should always configure buffering for PI Interface!

PI Buffer Subsystem (pibufss) runs as an **automatic Windows service** under either a Local System or custom service account, which is a recommended option.

## 10.2 The Buffering Mechanism

How does it work?

Put simply, the buffering mechanism intercepts the data as the interface sends it and the buffering application manages the communication between the data collection computer and the PI Data Archive.




There are three buffers used to process events: two memory buffers and a file buffer. Depending on the amount of data being buffered, the buffering process can be in one of three states. Initially, it uses a single memory buffer. When that fills, it switches to using dual memory buffers. When these buffers become full, it switches to using dual memory buffers and a file.

There are three distinct stages in the PI Buffer Subsystem data that flows from the PI Interface to the PI Data Archives. These stages are:

1. **Data Validation and Compression Marking:** PIbufss consumes the data, validating timestamps and values. If point-level **compression is enabled**, events are marked either as *snapshot only* or *to be archived*.
2. **N-Way Splitting and Queuing:** Once validated and marked for compression, all events are stored in, as many queues as there are targeted nodes (pibufq_<GUID>.0000.dat.) The GUID generated is unique for each PI Data Archive. Non-replicated PI Data Archives require only a single queue.
3. **Sending Data to the PI Data Archive:** Whenever the corresponding PI Data Archive is running and accepting data, queued events are sent to the remote server's PI Snapshot Subsystem. There events marked to be archived goes through Event Queue and Archive Cache into the PI Archive Subsystem for permanent storage.

## 10.3 Exercise – Configuring the PI Buffer Subsystem

Let's have our way with Buffering.

### Exercise Objectives

- Configure Buffering on PIINT01
- Validate data are being buffered

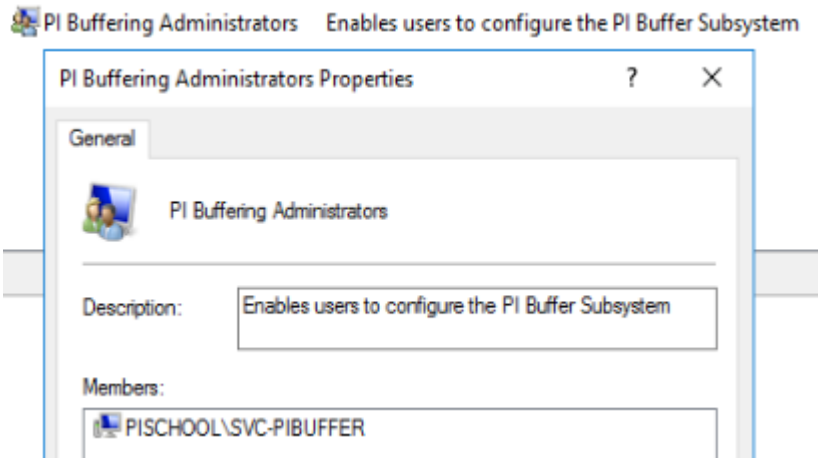
### Problem Description

If the PI Data Archive is shut down or disconnected from the network, you need to have data buffering installed and configured on the data acquisition computer to avoid data loss.

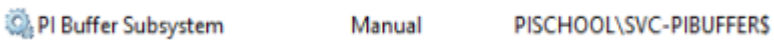
### Before You Start

PI Buffer Subsystem configuration is stored in *%ProgramData%\OSIsoft\Buffering*. Security access to this directory is controlled via local group **PI Buffering Administrators** which was created during installation. If custom account is used for PI Buffer Subsystem service, it also must be member of this group.

1. Open Local Users and Group (type **lusrmgr.msc** in Start Menu) and add **PISCHOOL\SVC-PIBUFFER$** as a member to the PI Buffering Administrators group.



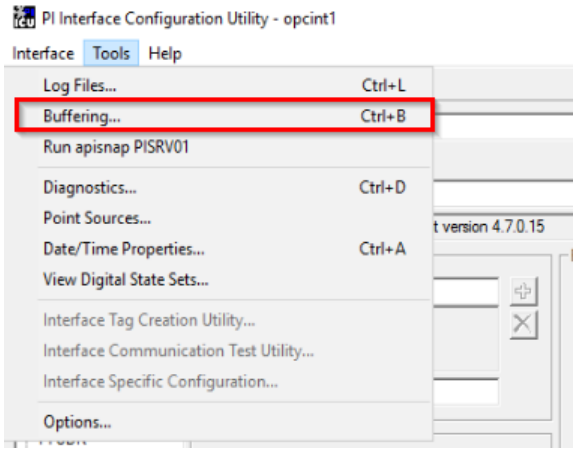

2. Buffering Manager utility is not capable to accept gMSA in configuration wizard. Therefore, prior running the wizard switch *Local System* account for gMSA **PISCHOOL\SVC-PIBUFFER$** in Services Manager for PI Buffer Subsystem service.



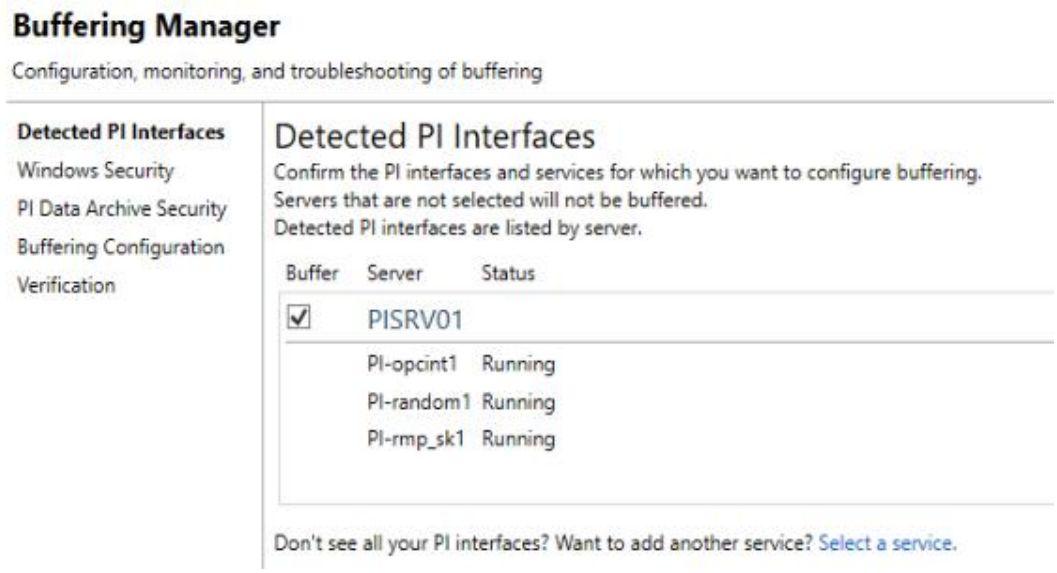
## Approach

To enable buffering on a data collector, follow these steps. (Buffering is also configured through entries in **piclient.ini** files located in both %PIHOME%\dat and %PIHOME64%\dat directories.)

1. Use the PI ICU to open the Buffering Manager in **Tools** → **Buffering**.

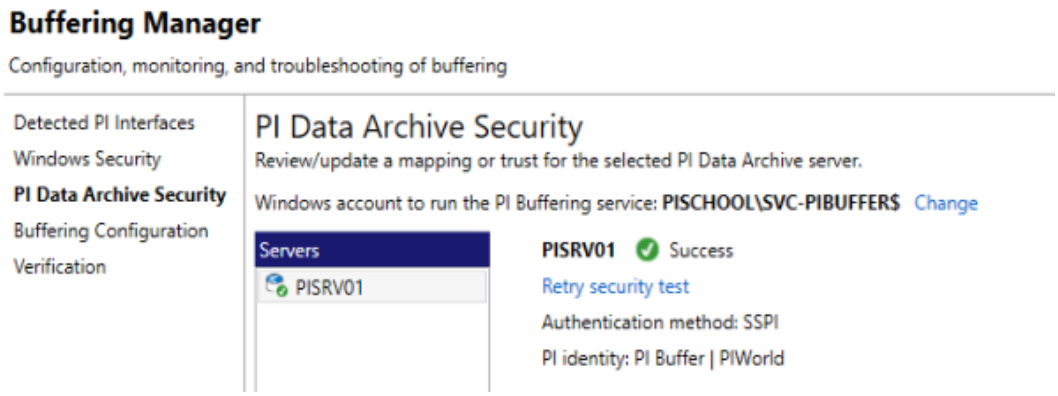



2. Select **Yes** and then **Continue with configuration**.
3. Check the **PI Data Archive** to confirm the PI Interfaces and services for which you want to configure buffering and click **Next**.

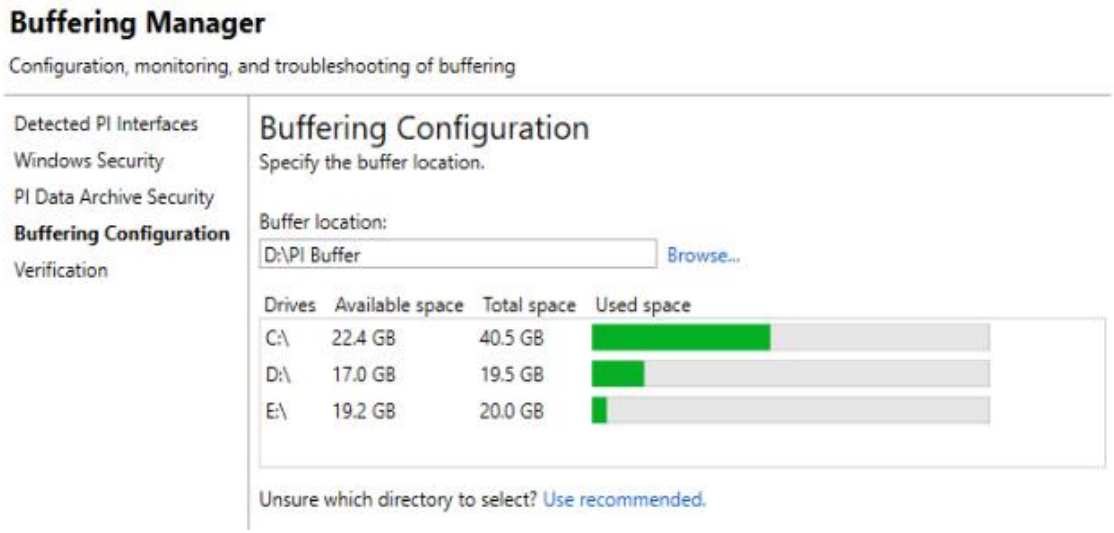



4. The next step asks to define the account for PI Buffer Subsystem Service. As gMSA **PISCHOOL\SVC-PIBUFFER$** was defined prior configuration in Services Manager, it is already filled in so just click **Next**.

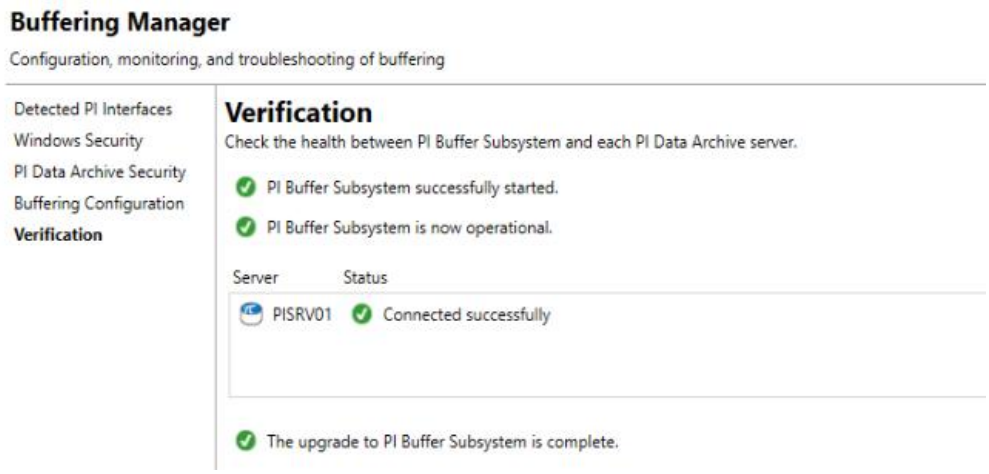

5. Buffering Manager runs the security test to access PI Data Archive, which is now granted as gMSA was mapped to **PI Buffer** identity previously in PI Interfaces exercise. Click **Next**.



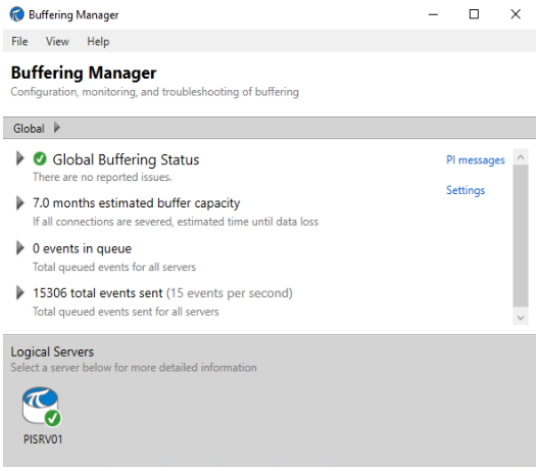

6. Select the Buffer Queue location. For this environment set the location to **D:\PI Buffer**. OSIsoft strongly recommends not having the buffer queue on the same drive as the OS to avoid failure on the interface node by filling up the main drive.



7. The last window will run a verification on the status of the PI Buffer Subsystem. If there are no errors, exit the installation wizard.




8. Once the installation wizard completed and closed the Buffering Manager window will open showing the status and statistics of the PI Buffer Subsystem.

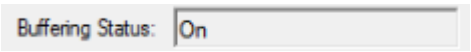



**Tip**

**CAUTION**: The Buffering mechanism will ***NOT*** collect data unless it starts ***BEFORE*** an interface.

After configuring PI Interfaces and buffering through the PI ICU, pibufss is added to the PI Interface service dependency.

9. Restart PI ICU to show the new field **Buffering status** in General tab. It should be **On**.




10. Data should go through the PI Buffer Subsystem as configuration wizard also restarts the PI Interfaces services to start after PI Buffer Subsystem service.

## 10.4 Directed Activity - Validating Data Buffering

How it is possible to verify that our data are buffered.

### Activity Objectives

Validate that PI Buffer Subsystem is collecting data. Watch the PI Vision display.

### Approach

1. Open the PI Vision display created in PI Vision chapter in browser on PIINT01.
2. Validate that you are buffering data from the previous directed activity. Open command line and navigate to directory ***%PIHOME64%\bin***. Use command ***pibufss –cfg***. If you see state **SendingData** then events are flowing through.

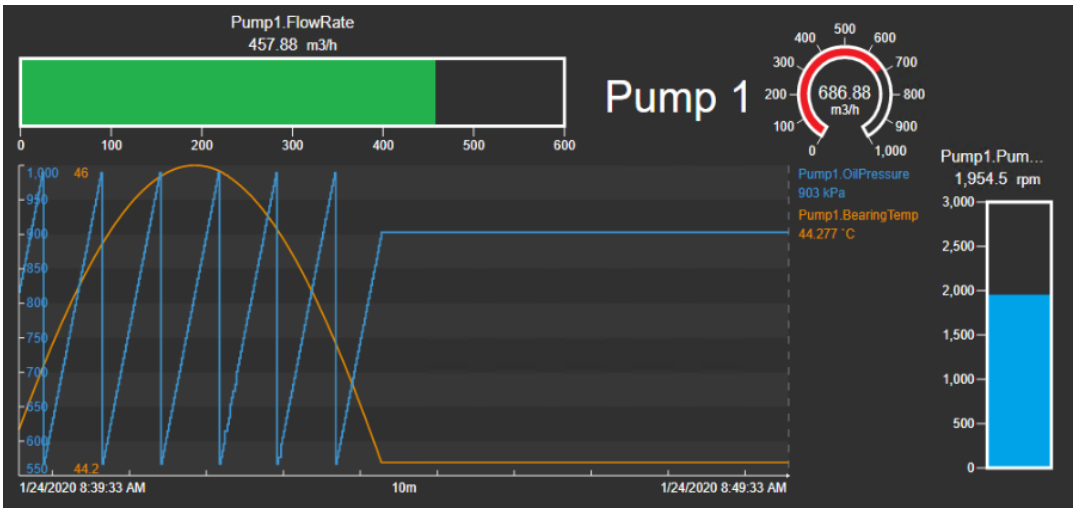
```
D:\Program Files\PIPC\bin>pibufss -cfg

*** Configuration:
Buffering: On (API data buffered)
Loaded physical server global parameters: queuePath=D:\PI Buffer authenticationOptions=SSPI;TRUST

*** Buffer Sessions:
  1 non-HA server, name: PISRV01, session count: 1
    1 [PISRV01] state: SendingData, successful connections: 1
      PI identities: PI Buffer | PIWorld, auth type: SSPI
      firstcon: 21-Jan-20 12:00:07, lastreg: 24-Jan-20 09:28:31, regid: 2
      total events sent: 3862783, snapshot posts: 496759, queued events: 0
```

3. Disconnect the PI Buffer Subsystem by command ***pibufss –bc stop*** from PI Data Archive.
4. Monitor the buffer queue with ***pibufss –qs.***

```
Counters for 24-Jan-20 09:52:11.37487 (pibufq_d3d3ba47-e54e-4ef7-8105-e25299a1446b.0000.dat)
            Primary File Size:      33554432           0
            Primary Page Size:         65536           0
          Primary Data Pages:            511           0
           Write Page Index:              20           0
            Read Page Index:              17           0
    Current Write Queue File:              0           0
     Current Read Queue File:              0           0
          Total Page Shifts:              20           0
            Available Pages:             507           0      (99.2%)
     Average Events per Page:               0           0
 Estimated Remaining Capacity:       1282721         -86  (1796.9 hr)
       Bytes in Primary File:        237734        1586
      Events in Primary File:          9171          61
          Total Event Writes:       3873589          61    (0.2/sec)
           Total Event Reads:       3864418           0
       Number of Queue Files:             1           0
            Events in Queue:          9171          61
```

In the screenshot above you can see that the **Events in Queue** are increasing.

5. Wait a moment to build up buffered data.
6. Monitor the PI Vision display. Notice the traces in trend object became flat.

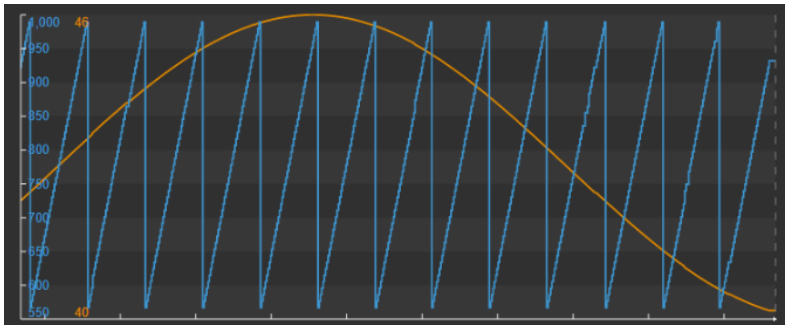



7. Now reconnect start the PI Data Archive or type ***%PIHOME64%\bin>pibufss -bc start***.

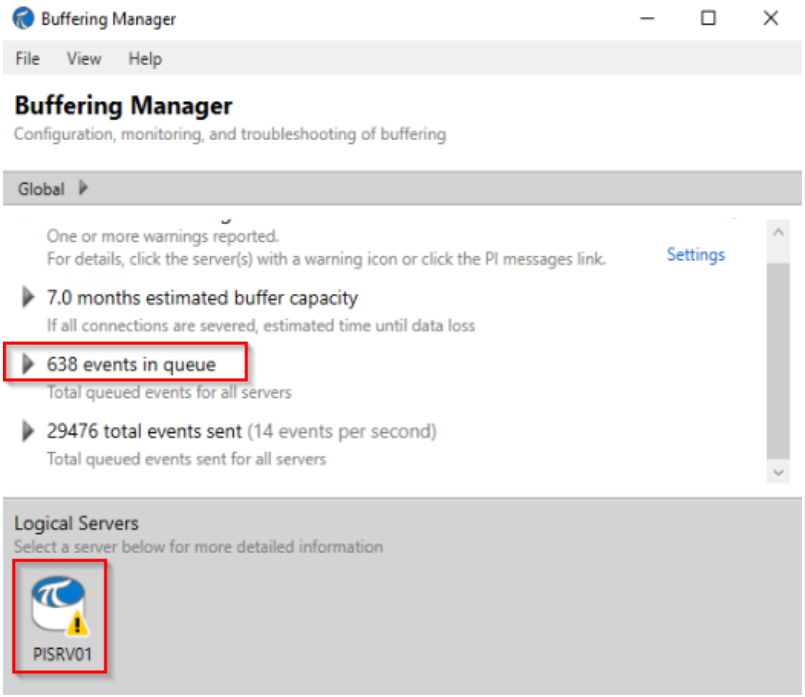
```
Counters for 24-Jan-20 09:53:37.37233 (pibufq_d3d3ba47-e54e-4ef7-8105-e25299a1446b.0000.dat)
              Primary File Size:       33554432           0
              Primary Page Size:          65536           0
            Primary Data Pages:             511           0
              Write Page Index:              21           0
               Read Page Index:              21           4
     Current Write Queue File:               0           0
      Current Read Queue File:               0           0
            Total Page Shifts:              21           0
              Available Pages:             510           4      (99.8%)
       Average Events per Page:               0           0
 Estimated Remaining Capacity:         1281575          85  (1803.2 hr)
        Bytes in Primary File:               0     -270030
      Events in Primary File:                0      -10417
          Total Event Writes:          3874922          87    (0.2/sec)
           Total Event Reads:          3874922       10504    (0.2/sec)
        Number of Queue Files:                1           0
              Events in Queue:                0      -10417
```

In the screenshot above you can see that the number of **Events in Queue** is now zero and the queued data were flushed. When there are no events queued, **Total Event Writes** and **Total Event Reads** counters need to match.

8. PI Vision display trend object has been backfilled.



9. In the Buffering Manager, the buffering statistics update automatically in order to show the Global Buffering Status, the estimated buffer capacity, the events in queue, and the total events sent in real time:




In the screenshot above the global buffering status has **Warning** icon. **Total Events Sent** counter is not increasing but **Events in Queue** counter is.

# 11. PI AF Tools

**Objectives**

- Introduce the PI System Explorer
- Describe elements and attributes
- Build elements and templates
- Build elements with the PI Builder add-in to Excel.

## 11.1 Assets – The Basic Building Blocks in the PI AF server

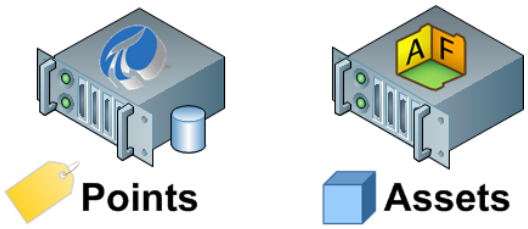
**What is an Asset?**

The PI Asset Framework (PI AF) Server is a part of the PI System. It contains assets or "metadata" usually organized according to the assets containing the attributes monitored. PI AF can be helpful to users of the PI Data Archive who know the assets but are not familiar with attribute nomenclature. With assets, data can be located without understanding the technical details of each piece of equipment.

Properly configured assets are helpful in finding all the SCADA points associated with a specific piece of equipment, as well as non-SCADA information such as serial numbers, maintenance dates and/or calculations associated with an asset.

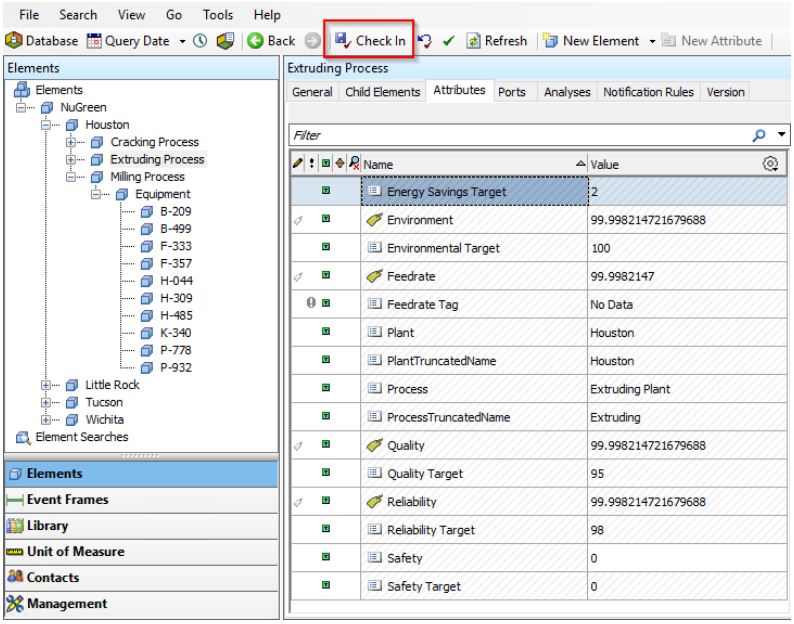
PI AF allows you to base similar objects on a single template. Templates define a set of base attributes for all the objects that use that template. Create the template once and you can create as many elements based on the template as are needed.

Modifying a template results in the change automatically propagating to all elements based on that template.




## 11.2 The PI System Explorer

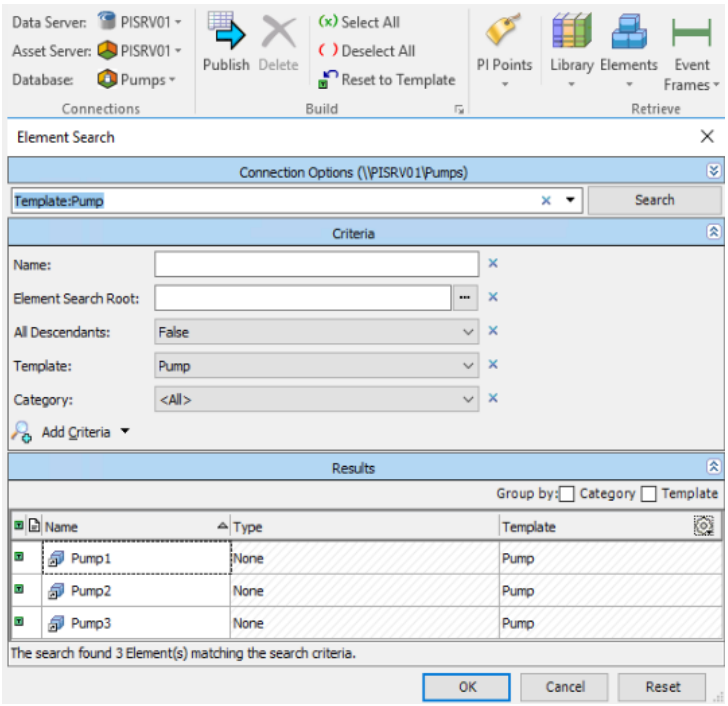
The ***PI System Explorer*** (**PSE**) is your access to the AF databases. Use the Explorer to configure the objects that represent your process, for example, tanks, transformers, pumps, boilers, meters. The PSE allows the definition of units of measure (UOM), templates, enumeration sets, and so forth. Use PSE to specify security permissions and default databases. All work can be checked in and out of the databases with the PSE.




## 11.3 The PI Builder

The ***PI Builder*** add-in to Excel is a tool allowing PI AF configuration in Excel, as well as PI point configuration. In the case of PI AF, it enables bulk importation and exporting of elements and attributes, to and from the AF database.

The element search dialog in PI Builder showing how to search for all pumps under the element search root of 'Site Pumps'.



## 11.4 PI Asset Analytics & Calculations

PI Asset Analytics & Calculations are defined in PI AF server over defined assets and asset structure and PI Analysis Service is the calculation engine.

PI AF Analysis consists of an expression that performs a calculation, and the scheduling for its execution. It takes existing values as inputs to produce new outputs, new calculated values or event frames. You can specify attributes from anywhere in your PI AF hierarchy as inputs to an analysis.

Every analysis is associated with an element or, preferably, with an element template. You can save analysis outputs by mapping them to attributes on that element or element template.

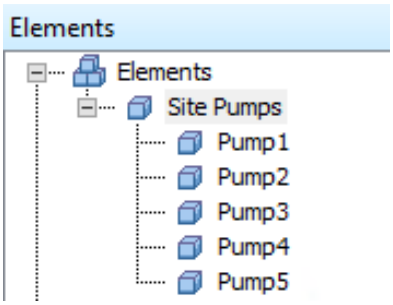
Three types of analysis are available:

- Expression
- Rollup
- Event frame generation
- SQC

As indicated previously PI AF can be used for asset-based calculations. In the AF Help file is a list of Data Reference Functions that can be utilised. The calculations can be one formula or a sequence of calculations and can have many input attributes. Calculations may be derived that use PI point data, and both internal and external table data, as well as static attribute data.

## 11.5 Directed Activity – Explore the Power of PI AF

Your instructor will coach you if you need assistance during the activity.




**Exercise Objectives**

- Explore features of AF
- Implement an asset hierarchy
- Build a PI Vision display upon Asset hierarchy
- The goal of the exercise is to build a meaningful PI AF structure and build a display based on the asset hierarchy shown.

**Exercise Description**

PI AF allows the definition of consistent representations of organization assets and equipment and uses these representations in simple or complex analyses that yield actionable information.

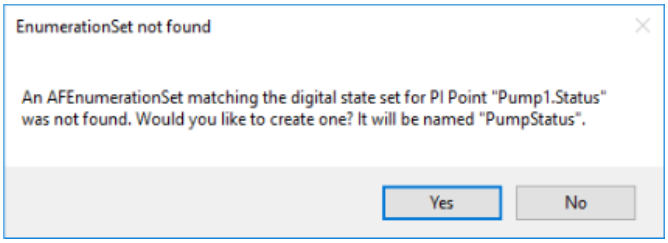
Without PI AF, displaying the data from five pumps in PI Vision may require five different displays, if referencing PI Points directly. With the usage of Element Template in PI AF, only one display need be created.

**Approach**

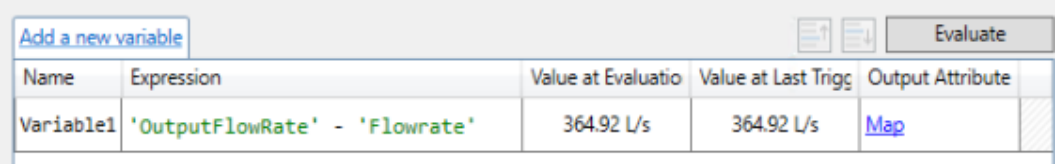
1. Use the PSE on PISRV01 to create a new database within the AF environment. Select Database, New Database and enter a database name, '**Pumps**', Select the database just created.
2. Create an element named "Site Pumps", precisely.
3. Right-click on 'Site Pumps' element to create **Child Element** of "Site Pumps" and name it **Pump1**
4. Start adding attributes according the following description:

| Attribute Name | Value Type | Value | Data Reference | Default UOM | Settings |
|---|---|---|---|---|---|
| Unit | string | | String Builder | none | Right("%Element%",1) |
| Fluid | string | Water | none | none | |
| Flowrate | single | | PI Point | L/s (from Volume Flow Rate UoM) | Pump1.Flowrate |
| PumpSpeed | single | | PI Point | RPM (from Angular Velocity UoM) | Pump1.PumpSpeed |
| BearingTemp | single | | PI Point | °C (from Temperature UoM) | Pump1.BearingTemp |
| OilPressure | single | | PI Point | kPa (from Pressure UoM) | Pump1.OilPressure |
| OutputFlowRate | single | | PI Point | L/s (from Volume Flow Rate UoM) | Pump1.OutputFlowRate |
| Status | string | | PI Point | none | Pump1.Status |

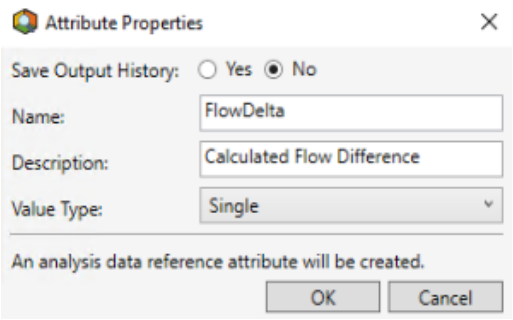
5. Status attribute creation displays a message, that Enumeration Set was not found. Enumeration set in PI AF is like Digital State Set in PI Data Archive. Click **Yes**.

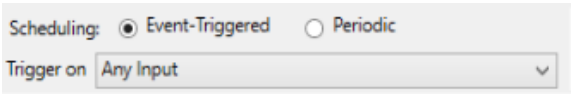


6. Switch to **Analyses** tab and **Create a new analysis**.
7. Keep Analysis Type: Expression and named it **FlowDelta**.
8. Insert expression **'OutputFlowrate' – 'Flowrate'** and click **Evaluate**.

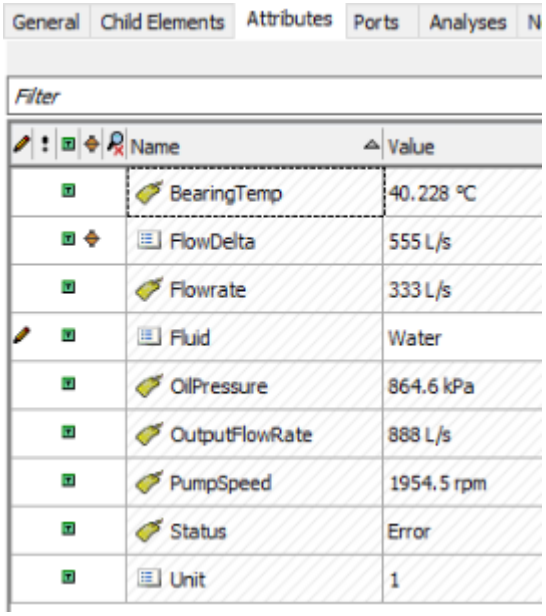



9. Map the analysis result to an output attribute. Select **New Attribute**.
10. In the Attribute Properties select **No** in Save Output History. Name: **FlowDelta**. Description: **Calculated flow difference**. Value Type: **Single**. Click **OK**.
11. Keep the default scheduling **Event-Triggered** and Trigger on **Any Input** at the bottom.



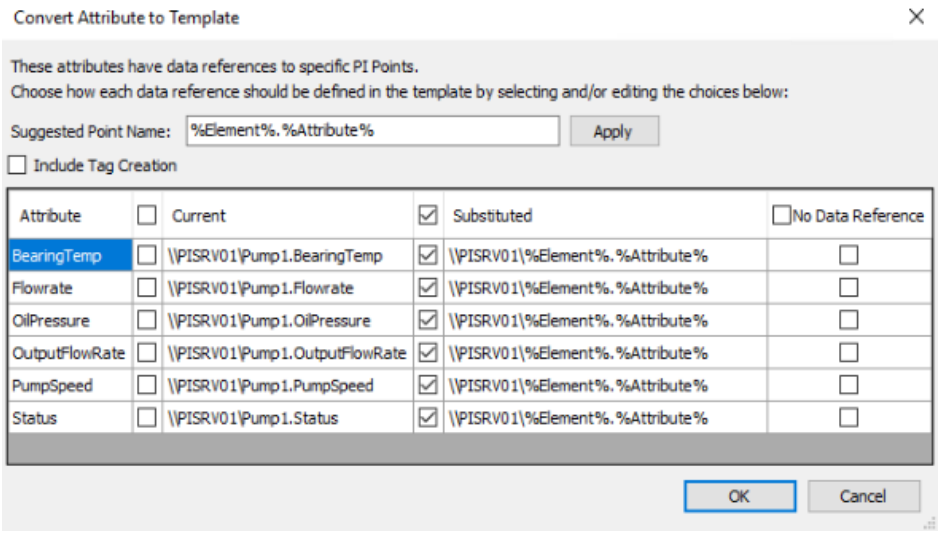

12. Check that all values are correct. And **Check In** the changes.



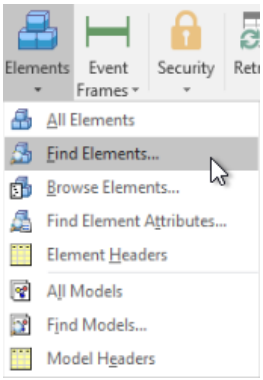

13. To create the other four pumps, make a **template** from Pump 1. Create an element template "**PumpTemplate**" by using *Convert to Template* when right clicking Pump1 in the Elements pane.

14. Leave 'Include Tag Creation' unticked as shown below and ensure the 'Substituted' definitions are used. **Remove the %ID%** from Suggested Point Name to have only **%Element%.%Attribute%** and click **Apply**.

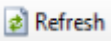
15. Go to Library and rename the template to just **Pump.**




16. Create new Elements from Pump  template for Pump2 and Pump3.
17. When you use the template, it will automatically associate the correct PI points with the attribute. **Why does this work?**
18. Keep PSE opened and open MS Excel. Select PI Builder tab. In Connections → Database select Pumps AF database.
19. Click on **Elements** and select **Find Elements** from drop-down list to open **Element Search** dialog window.
20. Select **Pump** template from Template drop-down list and hit **Search**. Select one of the pumps.
21. In Select Object Types and Column Headers deselect all options and only keep Required Columns and from Element section pick just **Template**.
22. Add new rows **Pump4** and **Pump5** to column Name.
23. Replicate values in columns Parent, ObjectType and Template and unselect first row, till it looks like this:

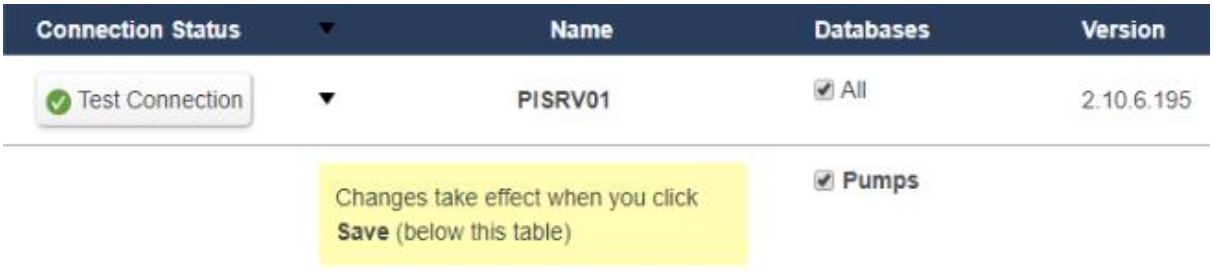



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Selected(x)** | **Parent** | **Name** | **ObjectType** | **Template** |
| 2 | | Site Pumps | Pump1 | Element | Pump |
| 3 | x | Site Pumps | Pump4 | Element | Pump |
| 4 | x | Site Pumps | Pump5 | Element | Pump |

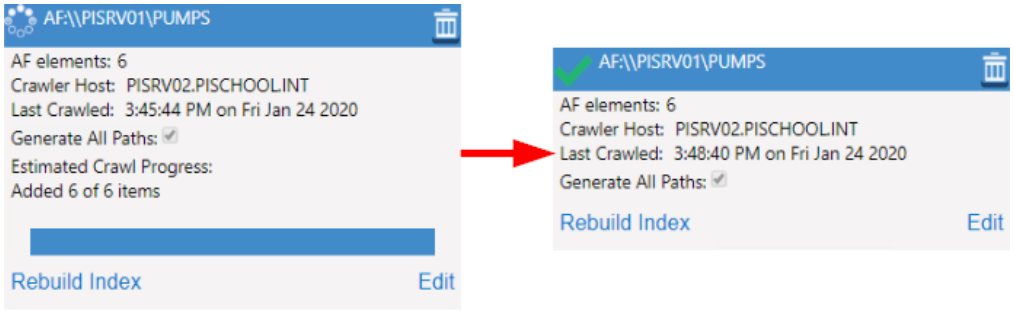
24. Click **Publish** and select **Create Only** edit mode.
25. Go back to PSE and refresh AF structure using **Refresh**  button. There are Pumps 4 and 5 in the asset tree.
26. Verify all attributes are referencing PI Points and there are no errors.
27. Modify each pump's **Fluid** attribute (or chose whatever liquids are available in the table):

| Unit | Pump | Material |
|---|---|---|
| Unit 2 | **Pump2** | Acetone |
| Unit 3 | **Pump3** | Castor Oil |
| Unit 4 | **Pump4** | Dioxan |
| Unit 5 | **Pump5** | Gasoline |

28. We need to add our Pumps AF database to PI Vision configuration. Open the PI Vision Administration (https://pisrv02/pivision/admin) and in Overview page locate Asset Servers/Databases Allowed section.
29. Click on Manage Configuration and tick the box next to **Pumps** database and hit **Save** button

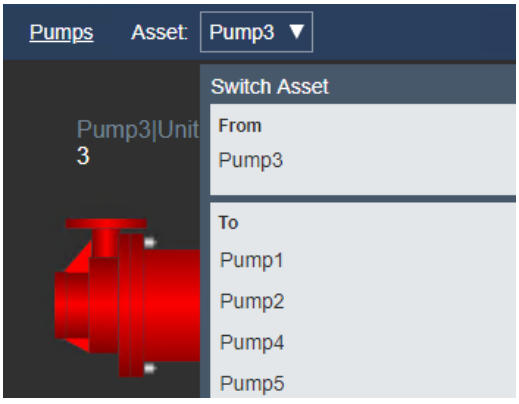



30. To be able to search for Elements and Attributes in PI Vision, PI Web API Crawler must build an index of the added Pumps database. Open the Search Service Administration page (https://pisrv02.pischool.int/piwebapi/admin/search/database.html) and wait until the index is built.




31. Open PI Vision main page and click on “New Display”.
32. Navigate to one of the Pump elements.
33. Create a new PI Vision display to show the characteristic of the Pump asset in a trend (OutputFlowRate, OilPressure,Flowrate) , and the pump unit number and fluid type as values. Display the FlowDelta in another trend (to see that the analysis calculation is backfilled by the PI Vision).
34. Save the display.


35. Because our Pumps elements are based on the same Element template, PI Vision automatically recognizes it and enables option to switch between assets, therefore one display can show data from all our Pumps.



# 12. PI Connectors

**Objectives**

- Introduction to PI Connectors, PI Connector Relay and PI Data Collection Manager
- Difference between PI Connectors (1st Gen) and PI Connectors (2nd Gen)
- Get to know the OPC UA standard
- Configuring the PI Connector for OPC UA (2nd Gen).

PI Connectors are the new generation of applications that simplify the process of adding new data to the PI System by scanning data sources and discovering data items. Data streams are then auto configured on the PI Server. There is no need to manually create PI Points during initial setup or in the future. The PI Connector (1st Gen) or PI Connector Relay does this for you by continually monitoring the source. A reference model is built in the PI AF server acting as a mirror image of the data source.

## 12.1 General Characteristics

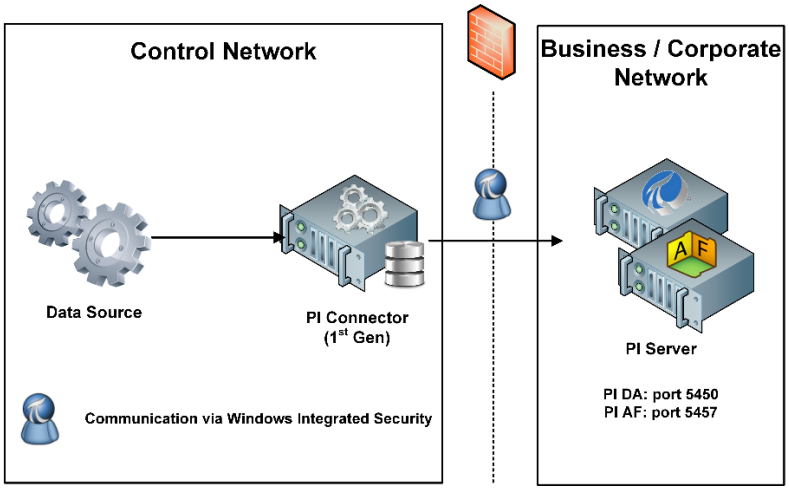
### 12.1.1 PI Interfaces vs. PI Connectors Comparison

The main differences between PI Interfaces and PI Connectors are summarized in the following table:

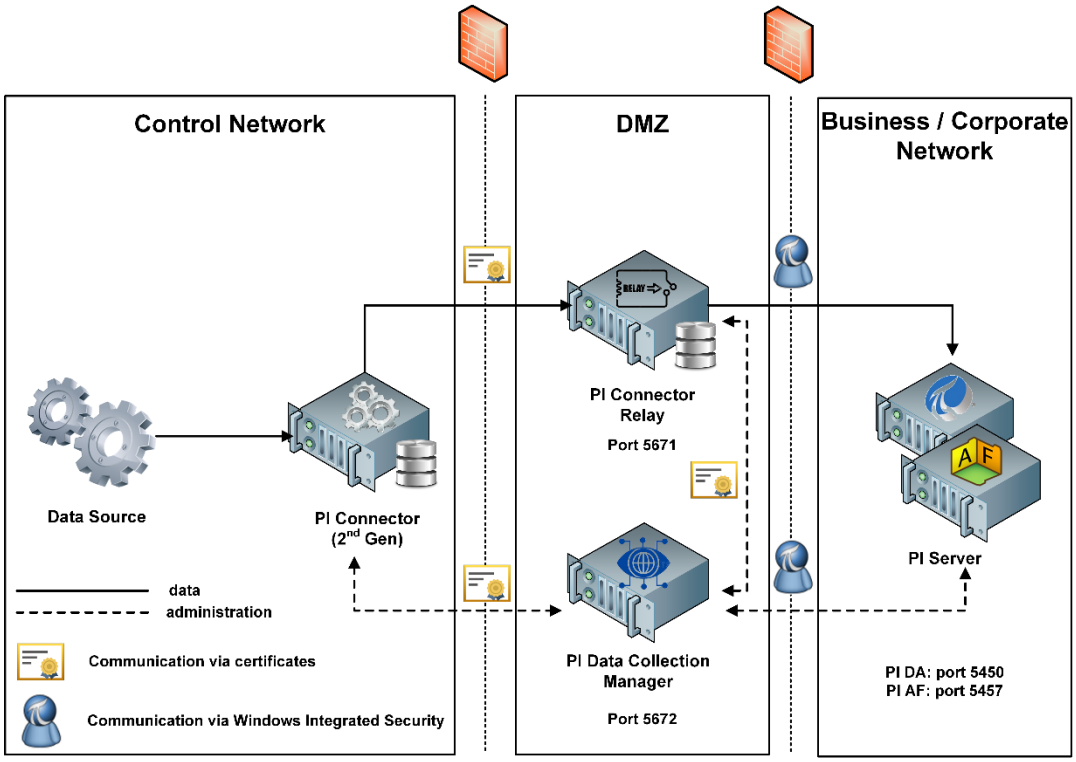
| | PI Interface | PI Connector (1st Gen) | PI Connector (2nd Gen) |
|---|---|---|---|
| PI Points | Must create PI Points | Auto-discovers & creates as needed | Auto-discovers. Creation done by PI Connector Relay |
| Point Type | Classic | Base | Base |
| Buffering | Manually configure buffering | Automatic built-in buffering | Automatic built-in buffering |
| Data types | Time-series data only | Time-series & meta data (Asset structure, Event Frames) | Time-series & meta data (Asset structure, Event Frames) |
| Administration | Only locally via PI ICU | Locally or Remotely in its own web-based UI | Remotely via PI Data Collection Manager |
| Configuration changes | Interface restart required | Does not require restart | Does not require restart |
| Number of instances | Once instance per data source | Only one instance on a server for multiple data sources | Only one instance on a server for multiple data sources |
| Exception filtering | Yes | No | No |
| Development environment | PI API | AF SDK | AF SDK |

### 12.1.2 PI Connectors 1st Generation vs. 2nd Generation

The **PI Connectors (1st Gen)** are standalone applications that are responsible for the whole data collection operation. Browsing data sources for PI points and asset hierarchy creation information, collecting data, sending them to PI Data Archive and PI AF and running buffering operations. Configuration is done directly with connector its own administration web-based user interface.




The **PI Connectors (2nd Gen)** do not connect to PI Server directly, but values are sent through middle-ware PI Connector Relay application. PI Connector configuration and administration is done via PI Data Collection Manager application and not directly on it.

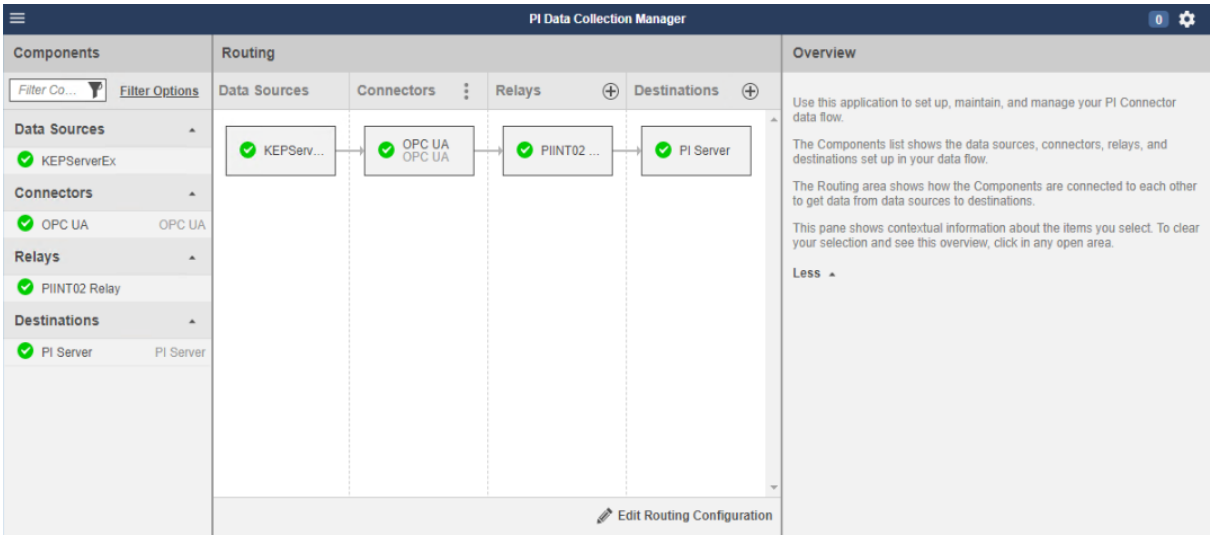


This architecture provides more security to the isolated control network. Communication between PI Connector, PI Connector Relay and PI Data Collection Manager is processed over secured HTTPS and AMQPS protocols connections using X.509 certificates instead of Windows Integrated Security that would require domain trusts or usage of windows credential manager.

**Note:** Always consult PI Connector user guide to determine the architecture. Whether it is a standalone 1st generation connector or 2nd generation that requires PI Connector Relay and PI Data Collection Manager. It cannot be determined from connector version. Version 1.x does not always mean 1st generation architecture. For example, the first PI Connector for oBIX 1.x version was already released for 2nd generation architecture.

### 12.1.3 PI Connector 2nd generation architecture components

**PI Data Collection Manager (PI DCM)**
Is a browser-based web application that enables users to configure and managed multiple PI Connectors and PI Connector Relays to multiple PI Systems. From PI DCM, users register PI Connectors, select data sources and route data through PI Connector Relays to specific destinations within the PI AF database. Once registered and configured, users can then view and monitor data flow in real-time on PI DCM dashboard. The dashboard also provides functionality to modify data sources, start and stop connectors and re-route data flow to alternative destinations.




**PI Connector Relay**
PI Connector Relay is a connector component that relays data coming from a PI Connector to the PI System. The PI Connector sends a generalized representation of the assets and real time measurements that it collects from the data sources to PI Connector Relay. PI Connector

Relay creates elements, event frames in PI AF servers, and PI points in PI Data Archive. PI Connector Relay can send data that comes from PI Connectors to multiple PI Systems.

### 12.1.4 Principles of Operation

#### Administration

A PI Connector or PI Connector Relay must first be registered before it can be administered by the PI DCM and participate in data collection. After registration, PI Connector and Relay initiate the connection to PI DCM. If the connection is allowed, PI DCM uses the connection to remotely administer the PI Connector and PI Connector Relay. PI DCM cannot initiate a connection. Administration communication occurs over port **5672** on PI DCM host.

#### Data Collection

PI Connectors and Relays must be registered with PI DCM and routed in data flow path to participate in data collection. PI Connector establishes a connection and sends data to port **5671** on PI Connector Relay host. The data flow is one-directional, but protocol control messages are sent by the PI Connector Relay when responding to the PI Connector.

### 12.1.5 Security

#### Types of connections

- **HTTPS**
  Service web-based user interfaces and listen for registration requests. HTTPS port is configurable. The X.509 certificate associated with the HTTPS port is self-signed and created during installation of PI Connector, PI Connector Relay and PI DCM along with port configuration.
- **AMQPS**
  AMQPS is a secure version of the AMQP (Advanced Message Queuing Protocol). AMQPS connections process administration message and forward data from PI Connector to PI Connector Relay. The AMQPS connections are secured using self-signed X.509 certificates created when PI Connector, PI Connector Relay or PI DCM are installed. The certificate used by an AMQPS connection is different from the certificate associated with HTTPS connection and has unconfigurable port. **5671** for PI Connector Relay and **5672** for PI DCM.

#### Local Groups

Installation of PI Connector (2nd Gen), PI Connector Relay and PI DCM creates two local groups:

- **PI Connector Administrators**
  Members of this group are allowed to access the administration web-based interfaces of PI Connector, PI Connector Relay and PI DCM. Perform administrative tasks and configuration.
- **PI Trusted Installers**
  Members of this group are allowed to register PI Connector to PI DCM and in PI DCM confirm the PI Connector registration and register PI Connector Relay.

### 12.1.6 Buffering data

PI Connectors and PI Connector Relay are inherently capable of buffering data whenever data flows from one process boundary to another. Buffering is always on and is not user configurable. There are no separate processes associated with buffering. Each connector and relay processes themselves buffers the data. Configuration for buffering is limited to specifying the folder during setup to determine where data is buffered.

Buffering to PI System servers (PI Data Archive and PI AF) includes time-series values, tag creation, and asset creation events.

**Note:** PI Connector Relay's buffering is now PI Collective aware. PI Connectors (1st Gen) buffering is not. But the PI Collective pickup is not automatic when Destination server is already configured. It has to be removed and added again in PI Data Collection Manager.

### 12.1.7 Modifications to PI Connector's AF structure

It is possible to modify the PI AF structure that is generated by the PI Connector Relay, but with several limitations and only a few actions allowed.

**The supported modifications are:**
- Adding custom AF attributes to the elements created by PI Connector Relay
- Adding Extended Properties to the element template
- Assigning a category to attribute templates
- Adding PI AF Analysis rules directly to an element or element template

**Unsupported modifications:**
- Deleting element templates created by PI Connector Relay
- Renaming element templates created by PI Connector Relay
- Deleting template attributes crated by PI Connector Relay
- Renaming template attributes created by PI Connector Relay
- Changing data type of attributes created by PI Connector Relay

### 12.1.8 Modifications of PI Connector's PI Points configuration

It is possible to modify selected point attributes after they are created by PI Connector Relay.
- Modification of the compression settings attributes
- Modification of PI Point security settings
- Attributes like Step, Scan, Archiving, Span, Zero, Typical Value

It is not possible to:
- Change the tag name
- Extended Descriptor

## 12.2 Exercise – Install PI Connector Relay and PI Data Collection Manager

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

Use minimum privileged accounts and set the security on PI Server. Install the PI Connector Relay and PI Data Collection Manager.

### Before You Start

PI Connector Relay and PI Data Collection Manager require permissions according the following table:

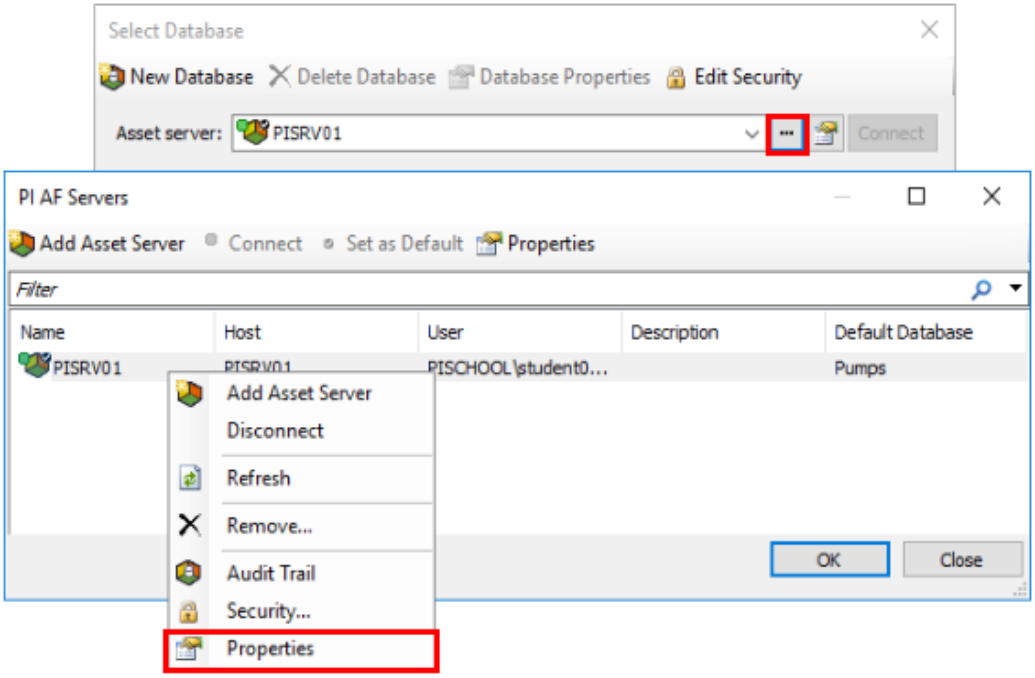
| Application | Destination | Items to access | Required access levels |
|---|---|---|---|
| PI Connector Relay | PI Data Archive  | PIDS | Read, Write |
| | | PIPOINT | Read, Write |
| | | PIUSER | Read |
| | PI AF  | AF Server Level | Read |
| | | AF Database | Read, Read Data, Write, Write Data |
| | | All items in AF DB | Read, Read Data, Write, Write Data |
| PI Data Collection Manager | PI Data Archive  | PIDBSEC | Read |
| | | PIDS | Read |
| | | PIPOINT | Read |
| | | PIREPLICATION | Read |
| | | PIUSER | Read |
| | PI AF  | AF Server Level | Read |
| | | AF Database | Read, Read Data |
| | | All items in AF DB | Read, Read Data |

For PI Data Archive:

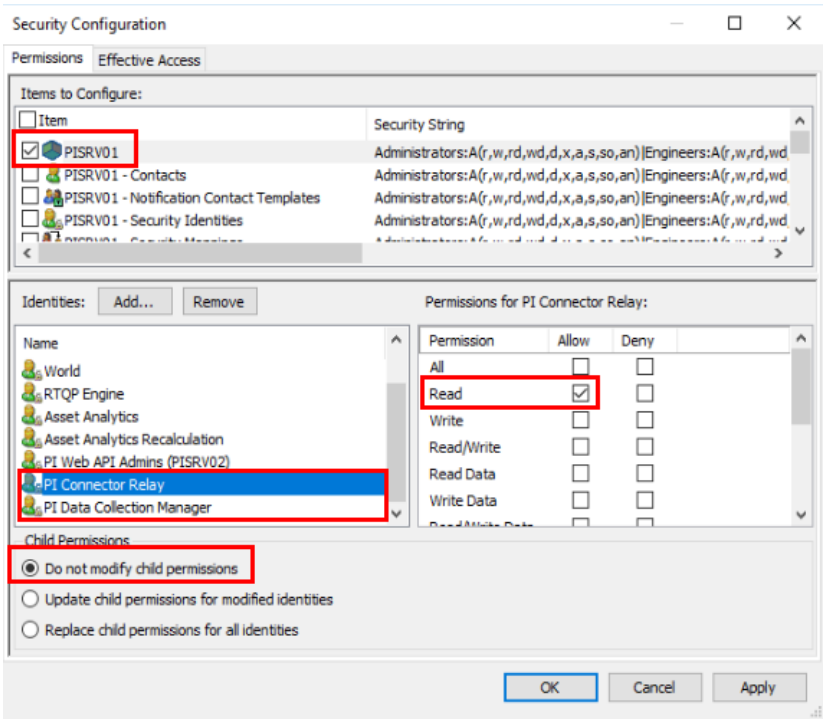
1. Open PI SMT on PISRV01 and navigate to Security → Identities, Users & Groups plug-in. Create new PI Identities **PI Connector Relay** and **PI Data Collection Manager**.
2. Map PI Connector Relay identity to **PISCHOOL\SVC-PIRELAY$** gMSA
3. Map PI Data Collection Manager identity to **PISCHOOL\SVC-PIDCM$** gMSA
4. Assign the permissions according the table in Database Security plug-in.

For PI AF:

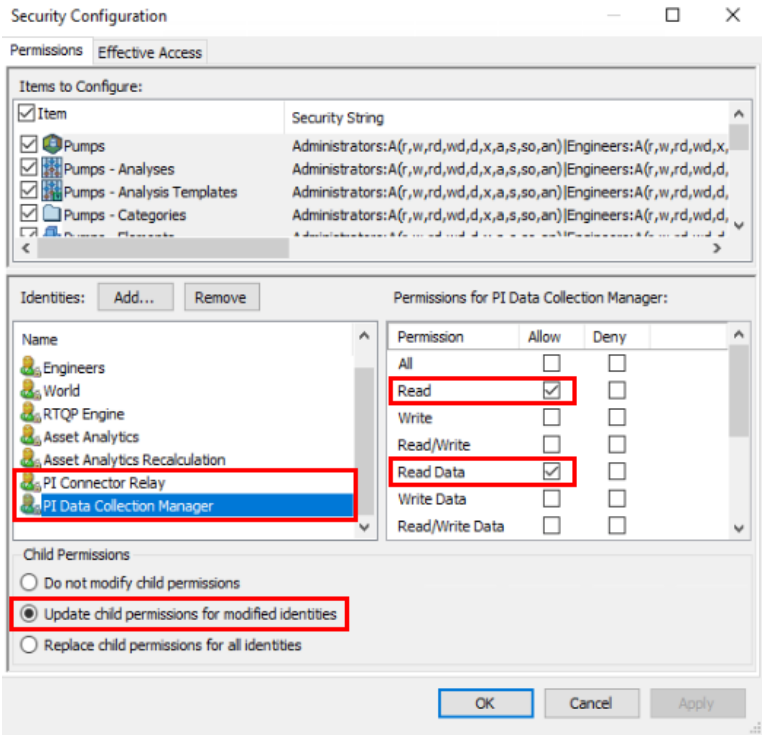
1. Open PSE on PISRV01 and click on Database.
2. In the Select Database window click on **[…]** (3 dots) button, then right-click on the PISRV01 AF server and select Properties.




3. In Identities tab do a right-click on blank space, select New Identity and enter name **PI Connector Relay**. Switch to Mappings tab and click on Add and map it to **PISCHOOL\SVC-PIRELAY$** gMSA.
4. Repeat the procedure for **PI Data Collection Manager** AF identity mapped to **PISCHOOL\SVC-PIDCM$** gMSA.
5. Back in PI AF Servers window, right-click on PISRV01 and select Security. From Items to Configure select only PI AF Server. Click on Add and select both AF Identities: PI Connector Relay and PI Data Collection Manager. After adding them to the list, assign just **Read** permissions to both.
6. From Child Permissions select Do not modify child permissions and confirm settings.



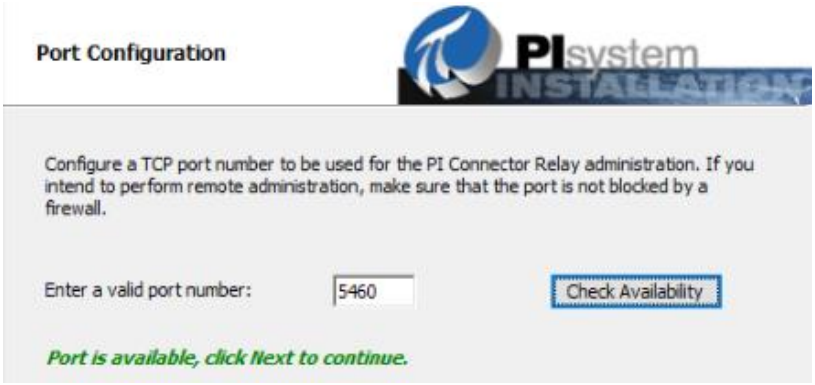

7. Open the security settings for **Pumps** database. Keep all items selected. Add **PI Connector Relay** AF identity and check **Read/Write** and **Read/Write Data** permissions (it automatically selects Read, Write, Read Data, Write Data).
8. Add **PI Data Collection Manager** AF identity and check **Read** and **Read Data**. Keep Update child permissions for modified identities.



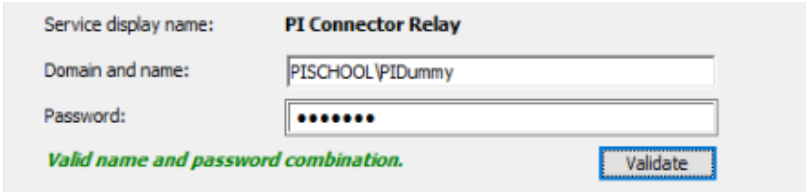
**Approach**

1. On PIINT02 in the PI Install Kits folder, run the PI Connector Relay installation kit.
2. As mentioned, PI Connector Relay and PI Data Collection Manager are using AF SDK, therefore PI AF Client is always bundled within their installation kit.
3. Fill the information about the Asset Server (PI AF) – **PISRV01**
4. Only PI AF SDK .NET 4 is necessary be installed. No other components are required, but PSE provides GUI access to AF SDK Connection Manager, therefore install it too.


5. In PI Connector Relay port configuration keep default port **5460** and Check Availability. Click **Next**.

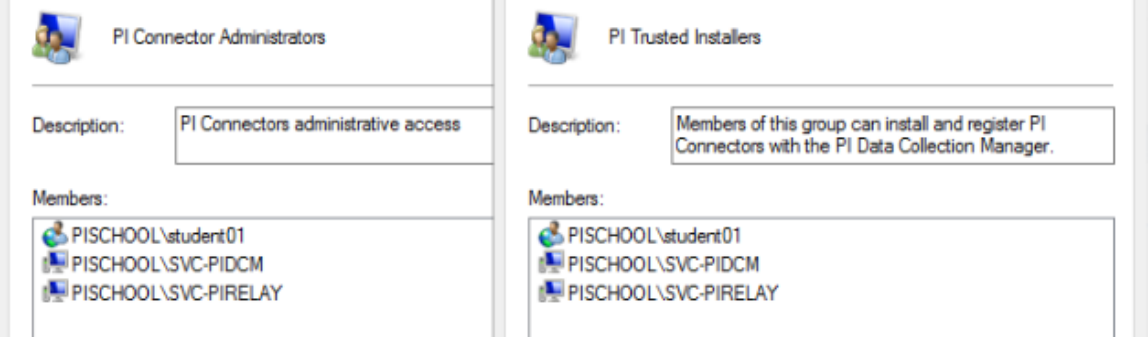



6. In Windows Service Configuration gMSA cannot be accepted, resulting into message *Invalid name and password combination.* . Use a dummy user account **PISCHOOL\PIDummy** (pass: pidummy) instead and **Validate**. Click **Next**.

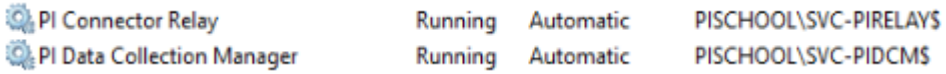



7. Instead of default buffer file location in %ProgramData% browse to **D:\PI Buffer** directory. Click Next and Install.

8. Add **PISCHOOL\Student01** user to *PI Connector Administrators* local group by **Add User**.
9. Click **Add More Users** and add **PISCHOOL\SVC-PIRELAY$** gMSA to the *PI Connectors Administrators* and *PI Trusted Installers* local groups. Also add **PISCHOOL\Student01** to *PI Trusted Installers* too.
10. Run PI Data Collection Manager install kit.
11. In port configuration select port **5461** as 5460 is already taken by PI Connector Relay and **Check Availability** and **Next**.
12. Again, in Windows service configuration gMSA is not accepted. Use **PISCHOOL\PIDummy** (pass: pidummy). **Validate**, click **Next** and **Install**.
13. **PISCHOOL\Student01** is already part of the group from PI Connector Relay installation. Click **Add More Users** and add **PISCHOOL\SVC-PIDCM$** to *PI Connector Administrators* and *PI Trusted Installers* local groups and remove **PISCHOOL\PIDummy**.




14. In Services Manager switch **PISCHOOL\PIDummy** user account of PI Data Collection Manager service for **PISCHOOL\SVC-PIDCM$** gMSA and restart it.

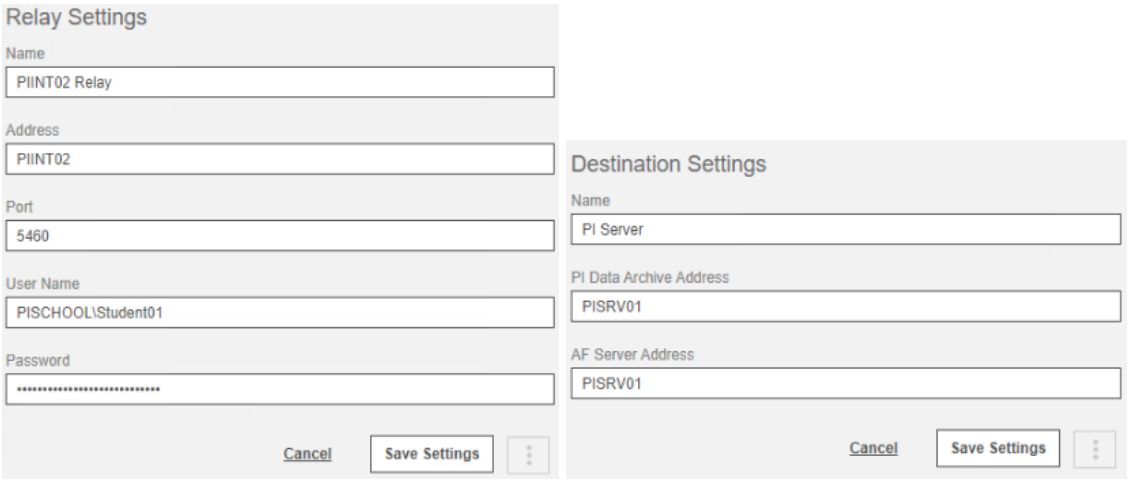



15. Verify the installation by accessing the PI Data Collection Manager and PI Connector Relay pages. Open Google Chrome and insert URL https://piint02:5460/admin/ui to access PI Connector Relay page and https://piint02:5461/ui to access PI Data Collection Manager.

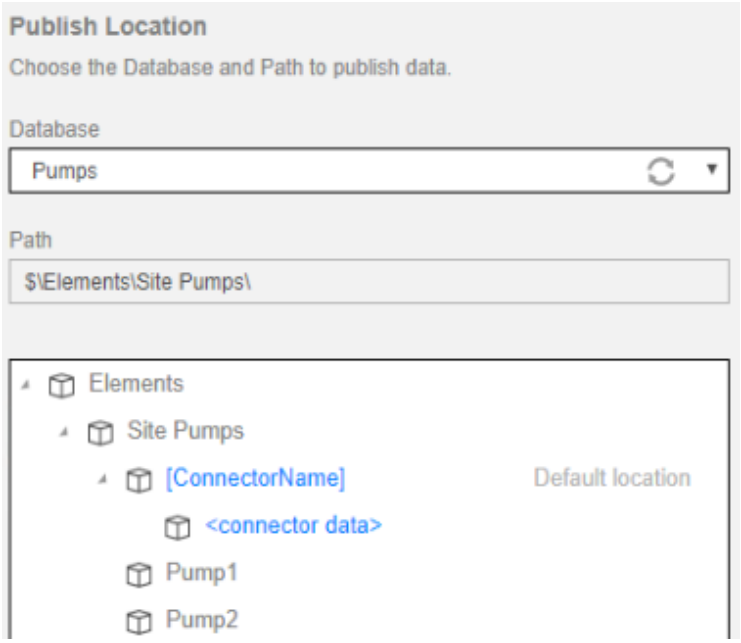
**Note:** Even though PISCHOOL\Student01 is a member of the *PI Connectors Administrators* and *PI Trusted Installers* local groups, there is still a login prompt for security reasons.

16. In PI Data Collection Manager add PI Connector Relay by clicking on the plus symbol ⊕ in **Relays** column and fill the Relay Settings:
    - Name: **PIINT02 Relay**
    - Address: **PIINT02**

- Port: **5460**
- User Name: **PISCHOOL\Student01** and password.



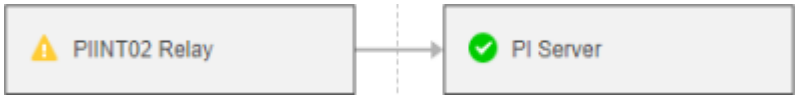

17. In **Destinations** column click Add Destination and fill Destination Settings:
    - Name: **PI Server**
    - PI Data Archive Address: **PISRV01**
    - AF Server Address: **PISRV01**
18. Save Settings. Click on PI Server and select **Data** tab.
19. Publish Location Pumps AF database is selected, but publish location is in root. Click on **Edit Destination Data Settings** at the bottom.
20. Then select a new path when prompted by message <select a path using the tree below> and click on Site Pumps to publish into this element.



21. Keep the selections Create root element and Prefix points checked and **Save Destination Data Settings**.
22. Click on PIINT02 Relay in Relays column. A check box appears next to the destination PI Server.


23. Check the box.


24. And click on **Safe Configuration** button at the bottom. In Summary page click **Save and Start All Components**.




25. PI Connector Relay is now connected to the destination PI Server, but with Warning message as there is currently no PI Connector configured, which will be done PI Connector for OPC UA configuration exercise.

## 12.3 PI Connector for OPC UA

### 12.3.1 About OPC UA

**OPC UA** = Open Platform Communication Unified Architecture

**5 Things to Know About the OPC UA Servers**

1. *OPC UA is the successor to OPC classic*: OPC DA (Data Access), OPC HAD (Historical Data Access) and OPC A&E (Alarms and Events) are all bundled into OPC UA. Instead of DCOM, which has several drawbacks such as frequent configuration issues, lower security and limitation to Microsoft Windows OS only, certificates are used for security.
2. *OPC UA is platform-independent and extremely scalable*: Classic OPC protocols were built for Microsoft operating systems (as DCOM was leveraged). OPC UA can operate on any platform. OPC UA is being deployed to everything from small chips with less than 64K memory (Nano Profile) to large workstations.
3. *A sophisticated address space model*: One should be able to model data, systems, machines and even entire plants into the OPC UA address space.
4. *OPC UA supports client-server architecture*: The OPC UA server is much more sophisticated than other factory floor systems such as Modbus, EtherNet/IP and BACnet. OPC UA servers can be configured to accept connections with any number of clients. The servers will never initiate connections. An interesting aspect of this relationship is that a server device can allow a client to dynamically discover what level of interoperability is supported, what services are offered, and type definitions for data types and objects.
5. *OPC UA is NOT just a protocol*: It's a common misconception that OPC UA is just another protocol (Set of rules that govern the transfer of information from one computer to another). Even though OPC UA specifies the rules for communication between computers, its vision is more than just moving data. OPC UA is about complete interoperability.

OPC UA servers on a network are defined by a server endpoint. A Server Endpoint is a physical address that allows clients to access one or more services provided by a server. Server endpoint is specified by its URL string.

Beside standard URLs e.g. HTTP or HTTPS, OPC Unified Architecture uses its own scheme **opc.tcp**. The endpoint has then the following syntax:

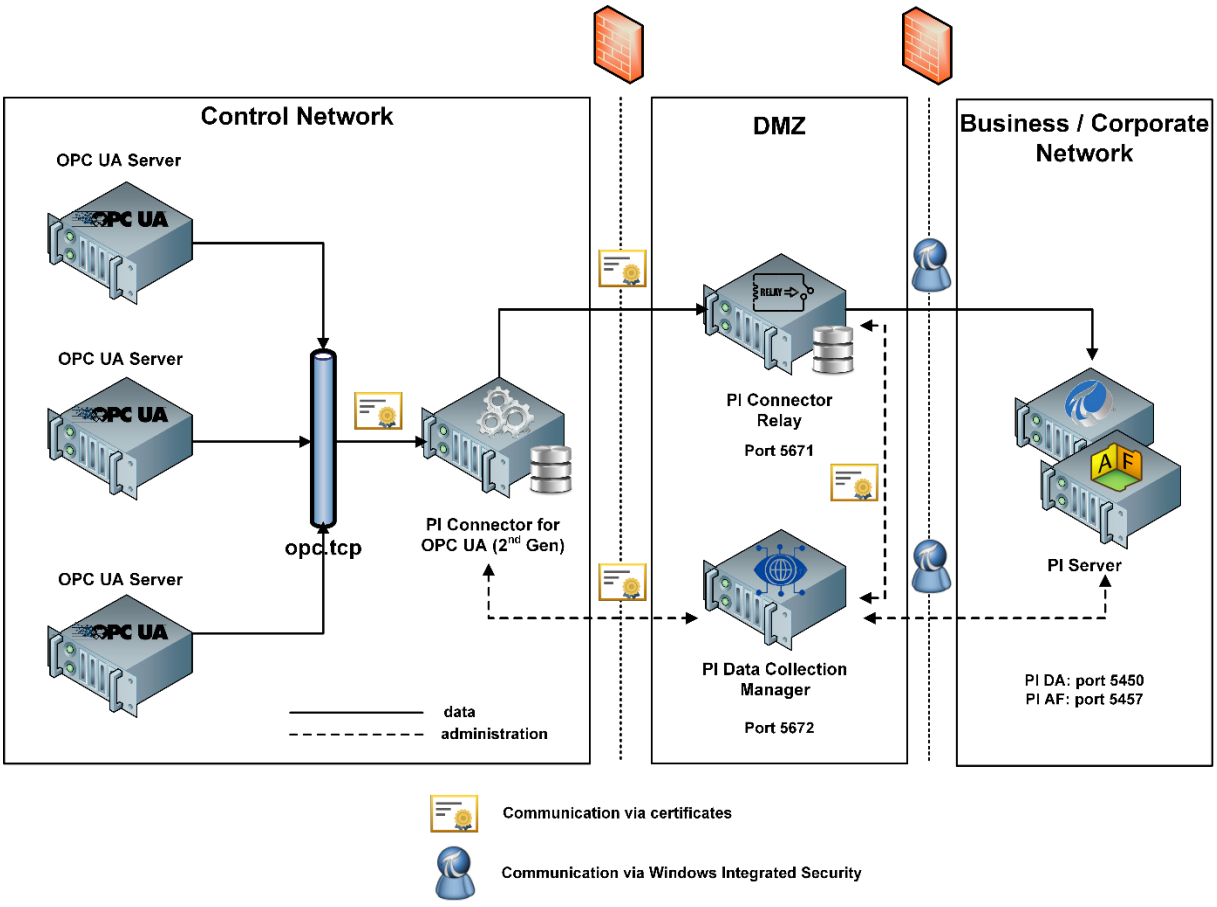
**opc.tcp://<OPCUAServerHost>:<port>**

**Note:** PI Connector for OPC UA currently only supports **opc.tcp** protocol.

### 12.3.2 PI Connector for OPC UA (2nd Gen) characteristics

PI Connector for OPC UA copies contextual and time-series data from OPC UA servers to PI Data Archive and PI AF Servers. Static OPC UA variables are mapped to AF elements and attributes, dynamic variables are converted to PI Points. Before starting, users have an option to browse the whole OPC UA address space and select manually, or by using a query, what they want to have replicated to PI System.

PI AF templates, elements, their attributes, and PI points are created automatically based on the information obtained from browsing the OPC UA address space. Data from OPC UA variables is read through subscriptions. The connector supports the Data Access (DA) and the Historical Data Access (HDA) parts of the OPC UA specification.




PI Connector for OPC UA **2.x** does **NOT** support a direct upgrade from version **1.x**! If you are upgrading your PI Connector for OPC UA 1.x version follow the procedure in section ***Upgrade to version 2.x*** in *PI Connector for OPC UA User Guide*.

Once configured, the connector performs the following tasks:

- Passes information to PI Connector Relay to create PI AF templates, elements, attributes and PI points.
- Collects data from OPC UA variables through subscriptions
- Reads history from those OPC UA dynamic variables that have historical data.

### 12.3.3 Modes of Operation

PI Connector can run in 4 different modes of operation:

**Generic mode**

Default mode and the closest one to how PI Connector for OPC UA 1.x worked. The most significant difference is that the filtering is no longer based on OPC UA object types. Users have the option to filter manually or using a query to select the actual OPC UA objects. Hence the filter file, known from version 1.x of this connector, is no longer needed.

**PCS7 mode**

The PCS7 mode was included to enable users to run this connector also with SIMATIC OpenPCS 7 UA servers, for which they previously needed a separate connector (PI Connector for Siemens SIMATIC PCS7). The main differences compared to *Generic* mode are how the PI AF structure is built, and how the static and dynamic variables are recognized.

**TagsOnly mode**

In this mode, users have the option to reference a .csv file with a list of NodeIds and several other optional columns. Only those OPC UA variables, which have their NodeIds listed in the file, will be subscribed for changes and will consequently forward events to the PI System. In this mode, **no PI AF hierarchy is created**.

**ISA95 mode**

In this mode, the connector makes a few assumptions described in more details in connector's user guide about where to search for information in OPC UA address space for AF templates to create PI AF hierarchy and how to name them, unlike in *Generic* mode.

For more details consult the *PI Connector for OPC UA User Guide*

## 12.4 Exercise – Install PI Connector for OPC UA

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

Install the PI Connector for OPC UA.

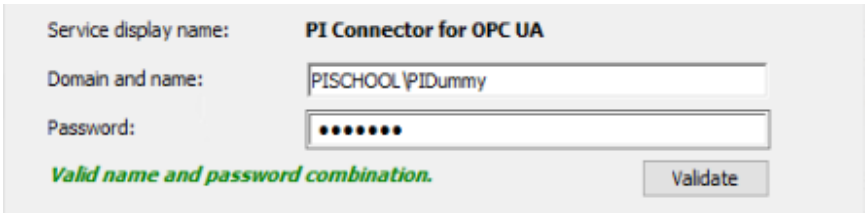
### Description

This exercise will guide you through the PI Connectors installation.

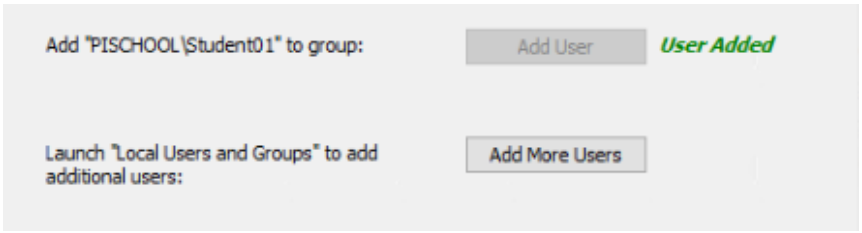
### Approach

1. On PIINT01 in PI Install Kits directory locate PI Connection for OPC UA install kit and run it.
2. In port configuration keep the default port **5460** and Check Availability. Click Next.

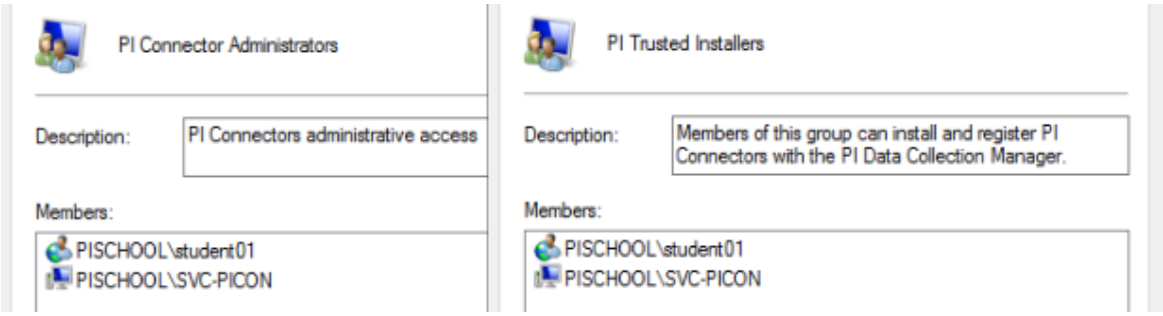

3. As for PI Connector Relay and PI Data Collection Manager, PI Connector install kit does not accept gMSA. Use **PISCHOOL\PIDummy** (pass: pidummy) instead and **Validate**. Click **Next**.



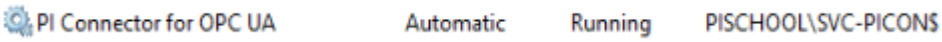

4. Select **D:\PI Buffer** directory for buffer files location instead of default %ProgramData%. Click **Next** and **Install**.
5. Add **PISCHOOL\Student01** to the *PI Connectors Administrators* local group.



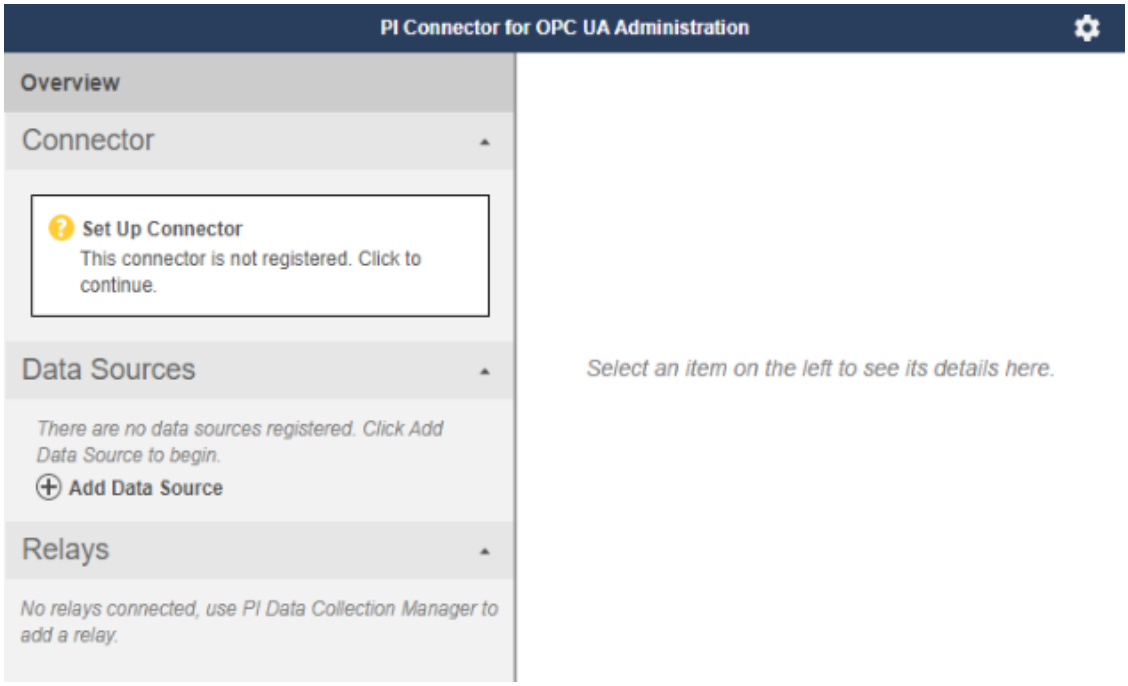
6. Click **Add More Users** and add **PISCHOOL\SVC-PICON$** gMSA to the *PI Connectors Administrators* and *PI Trusted Installers* local groups.
7. Remove **PISCHOOL\PIDummy** from the groups and add **PISCHOOL\Student01** also to *PI Trusted Installers*.




8. Finish and Close the installation wizard.
9. In Services Manager switch **PISCHOOL\PIDummy** user account of PI Connector for OPC UA service for **PISCHOOL\SVC-PICON$** gMSA and restart it.




10. Verify the installation by accessing PI Connector for OPC UA page. Open Google Chrome and insert URL https://piint01:5460/ui.



## 12.5 Directed Activity – Configuring PI Connector for OPC UA

Follow the instructor to guide you through the configuration settings until the asset structure and points are created

### Activity Objectives

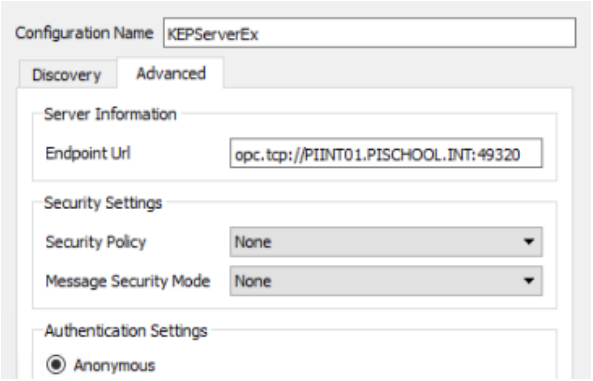
- Use the OPC UA Client Tool to verify the connection to the OPC UA Server and browse the structure and obtain Root Node ID
- Register PI Connector for OPC UA to PI Data Collection Manager
- Configure PI Connector for OPC UA via PI Data Collection Manager
- Verify the asset structure and points were created after configuration

### Approach

1. Verify the KEPServer EX 5.17 OPC UA Server services are running on PIINT01.

| | | | | |
|---|---|---|---|---|
| KEPServerEX 5.17 Event Logger | Event logger component of Kepware Communications Server 5.17 | Automatic | Running | Local System |
| KEPServerEX 5.17 Key Service | Key service component of Kepware Communications Server 5.17 | Manual | Running | Local System |
| KEPServerEX 5.17 Runtime | Runtime component of Kepware Communications Server 5.17 | Automatic | Running | Local System |

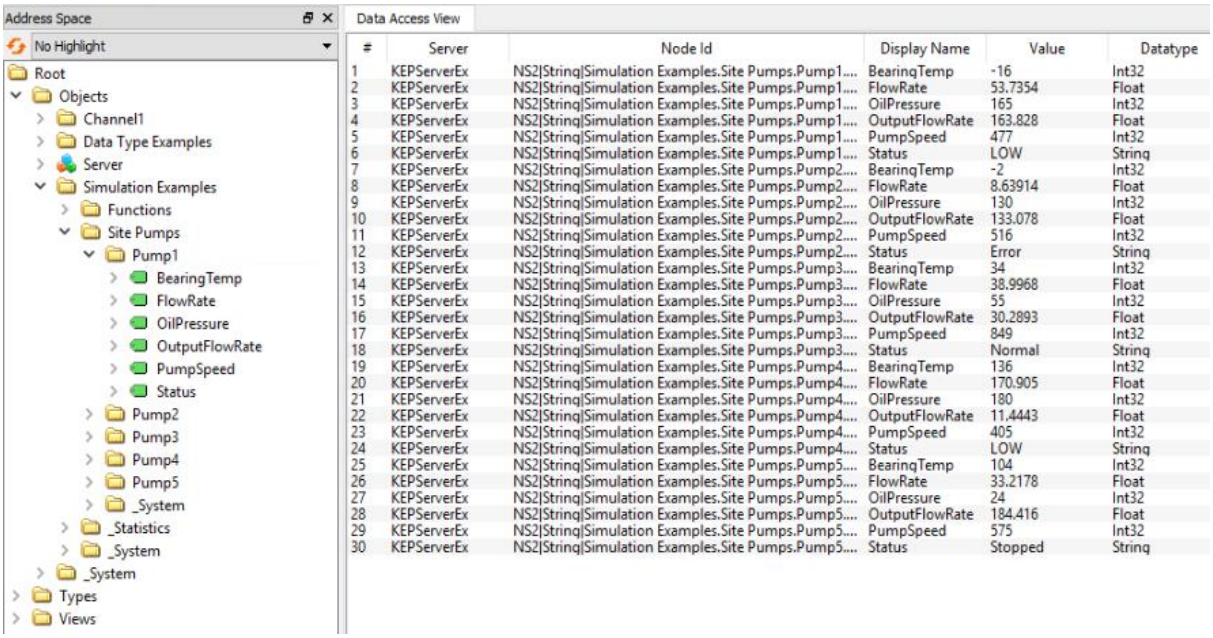
2. KEPServerEx icon  is available in system tray. Right-click on it and select OPC UA Configuration to obtain opc.tcp endpoint of the OPC UA Server. In this case ***opc.tcp://PIINT01.PISCHOOL.INT:49320***.
3. Open **UA Expert** OPC UA Client via desktop shortcut.
4. In the main menu go to **Servers** → **Add**. In Add Server dialog switch to **Advanced** tab.
    a. Configuration name is up to you. For example, KEPServerEx.
    b. Insert endpoint URL obtained from OPC UA Server configuration.
    c. Select **None** in both lists in Security Settings and **Anonymous** in Authentication Settings.





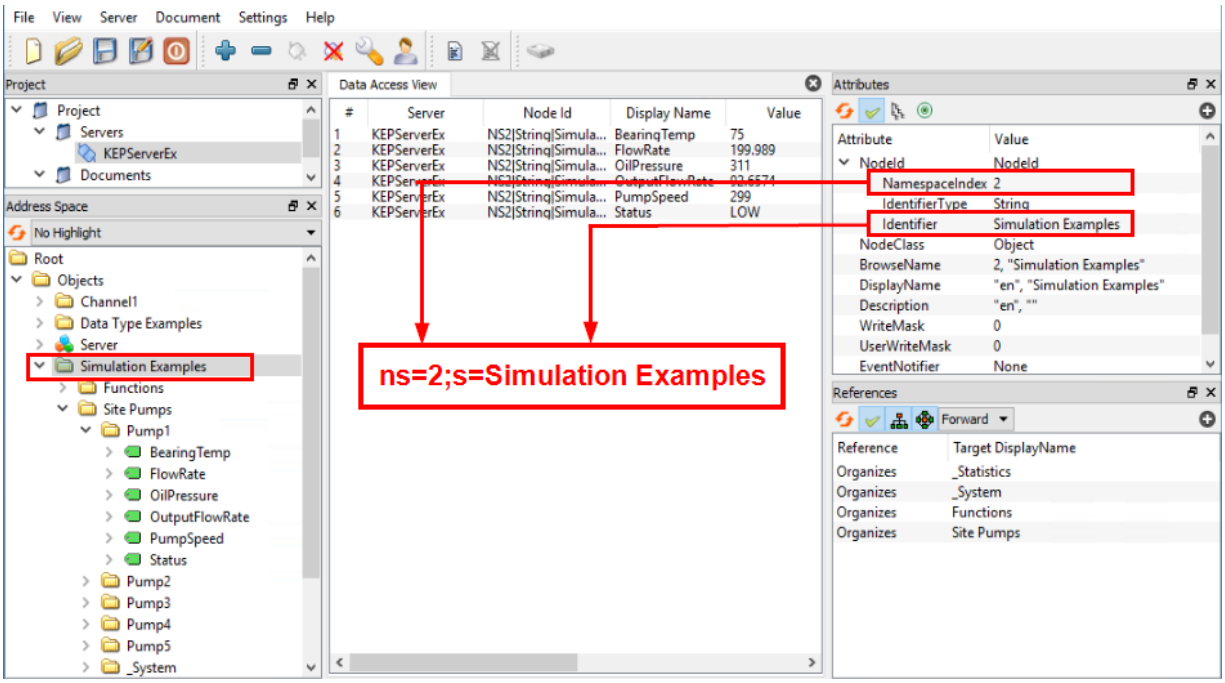
5. Server configuration is added to the **Project** section. Right-click and **Connect**.
6. In Certificate Validation dialog select **Trust Server Certificate**.

7. In **Address Space** section browse to Objects → Simulation Examples → Site Pumps → Pump1. Select all variables, drag and drop them to the **Data Access View** section. To see the values are updating. You may repeat it for other Pumps.




| # | Server | Node Id | Display Name | Value | Datatype |
|---|---|---|---|---|---|
| 1 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump1.... | BearingTemp | -16 | Int32 |
| 2 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump1.... | FlowRate | 53.7354 | Float |
| 3 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump1.... | OilPressure | 165 | Int32 |
| 4 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump1.... | OutputFlowRate | 163.828 | Float |
| 5 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump1.... | PumpSpeed | 477 | Int32 |
| 6 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump1.... | Status | LOW | String |
| 7 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump2.... | BearingTemp | -2 | Int32 |
| 8 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump2.... | FlowRate | 8.63914 | Float |
| 9 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump2.... | OilPressure | 130 | Int32 |
| 10 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump2.... | OutputFlowRate | 133.078 | Float |
| 11 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump2.... | PumpSpeed | 516 | Int32 |
| 12 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump2.... | Status | Error | String |
| 13 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump3.... | BearingTemp | 34 | Int32 |
| 14 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump3.... | FlowRate | 38.9968 | Float |
| 15 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump3.... | OilPressure | 55 | Int32 |
| 16 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump3.... | OutputFlowRate | 30.2893 | Float |
| 17 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump3.... | PumpSpeed | 849 | Int32 |
| 18 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump3.... | Status | Normal | String |
| 19 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump4.... | BearingTemp | 136 | Int32 |
| 20 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump4.... | FlowRate | 170.905 | Float |
| 21 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump4.... | OilPressure | 180 | Int32 |
| 22 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump4.... | OutputFlowRate | 11.4443 | Float |
| 23 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump4.... | PumpSpeed | 405 | Int32 |
| 24 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump4.... | Status | LOW | String |
| 25 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump5.... | BearingTemp | 104 | Int32 |
| 26 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump5.... | FlowRate | 33.2178 | Float |
| 27 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump5.... | OilPressure | 24 | Int32 |
| 28 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump5.... | OutputFlowRate | 184.416 | Float |
| 29 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump5.... | PumpSpeed | 575 | Int32 |
| 30 | KEPServerEx | NS2\|String\|Simulation Examples.Site Pumps.Pump5.... | Status | Stopped | String |

PI Connector for OPC UA must browse the entire OPC UA server address space for the first time. This may take a long time for OPC UA Servers with complex address space containing thousands of items and variables. But for data selection it is possible to narrow it down to only specific part of that address space, by defining the **Root Node ID** in PI Connector's data source settings. It is a unique combination of **Namespaceindex (ns)** and **Identifier (s)** in syntax **ns=<index#>;s=<string>**




| # | Server | Node Id | Display Name | Value |
|---|---|---|---|---|
| 1 | KEPServerEx | NS2\|String\|Simula... | BearingTemp | 75 |
| 2 | KEPServerEx | NS2\|String\|Simula... | FlowRate | 199.989 |
| 3 | KEPServerEx | NS2\|String\|Simula... | OilPressure | 311 |
| 4 | KEPServerEx | NS2\|String\|Simula... | OutputFlowRate | 93.6574 |
| 5 | KEPServerEx | NS2\|String\|Simula... | PumpSpeed | 299 |
| 6 | KEPServerEx | NS2\|String\|Simula... | Status | LOW |

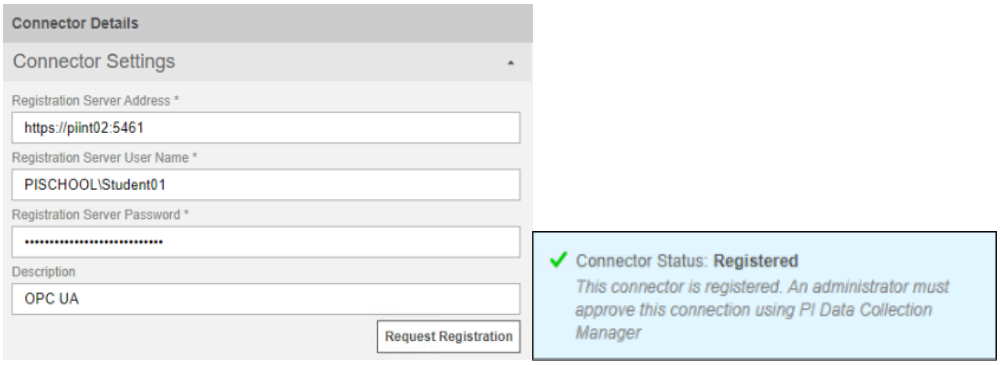
8. To present only Simulation Examples address space section for data selection, Node ID: **ns=2;s=Simulation Examples** will be used in PI Connector configuration.

**Note**: There are several OPC UA Clients from different vendors. In some of them, you need to put together the Node ID using the Namespaceindex and Identifier. Some can display the Node ID directly in ns=<index>;s=<string> form.

This is the validation of the updating variables on OPC UA Server. Now to configuration of PI Connector for OPC UA.

To be able to configure PI Connector it must be registered in PI Data Collection Manager:

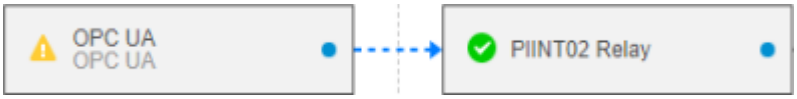
9. To register PI Connector open PI Connector for OPC UA Administration page locally on PIINT01 (https://piint01:5460/ui). Click on **Set Up Connector** and fill **Connector Settings**.
    - Registration Server Address: **https://piint02:5461**
    - Registration Server User Name: **PISCHOOL\Student01**
    - Registration Server Password: **<Student01 password>**
    - Description: **OPC UA**
10. After clicking on Request Registration, the PI Connector is now registered in PI Data Collection Manager and awaiting approval.



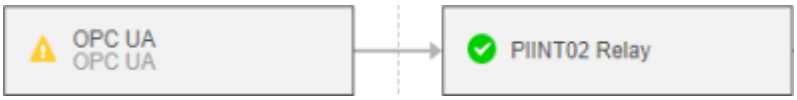

**Note:** OSIsoft recommends **registering** PI Connectors into PI Data Collection Manager **locally**. Attempting PI Connectors registration accessing their administration page remotely may result into errors.

11. Switch to PI Data Collection Manager. PI Connector appeared in **Connectors** column. Click on it and change the name to **OPC UA**. Then click on **Approve This Registration and Configure**.

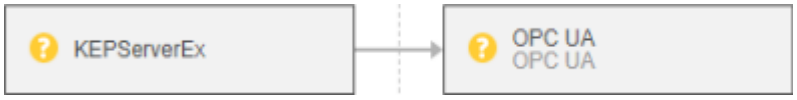

12. Tick the box next to PI Connector Relay to connect PI Connector.




13. Click on **Save Configuration** at the bottom and in Summary window select **Save and Start All Components**.




14. PI Connector is now connected to PI Connector Relay, but it is missing the Data Source. Click on OPC UA PI Connector and in Connector Details select **Data** tab and **Add Data Source**.
15. Fill the Data Source Settings:
    - Name: **KEPServerEx**
    - Description: **Site Pumps**
    - Mapping Type: **Generic**
    - Discovery or Server Endpoint URL: **opc.tcp://PIINT01.PISCHOOL.INT:49320**
    - User Name and Password: **<blank>**
    - Root NodeIds: **ns=2;s=Simulation Examples**
16. Click **Discover Available Endpoints** and from drop-down list that appeared select **[SignAndEncrypt:Basic128Rsa15:Binary]**and **Save**.

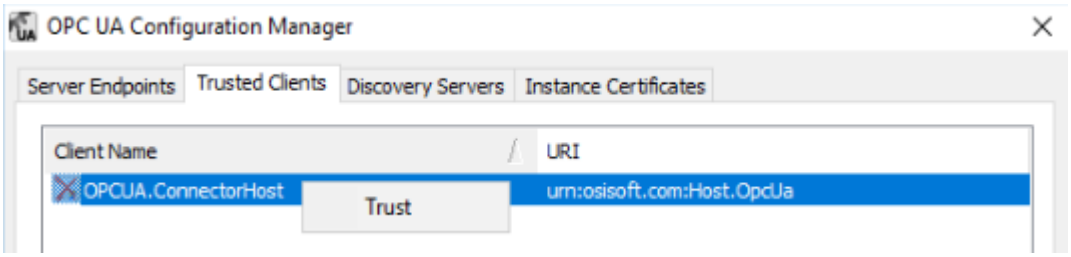



17. Select OPC UA connector and in **Data** tab click on **Discover Data Source Contents**. Message window pops up to inform that entire address space will be browsed, and it can take a long time. Agree and continue.

PI Connector and OPC UA Server exchange certificates, but they do not trust each other certificates right-away. Thus, the error message in Message Log in Diagnostics tab for PI Connector: "***Error encountered while running discovery task for data source***"


To make PI Connector and OPC UA server trust each other certificates, perform following tasks:

18. On PIINT01. Go to **%PIHOME64%\Connectors\OPCUA\pkclient\rejected\certs** directory. There is KEPServerEx certificate. Move the certificate to the **…\pkclient\trusted\certs** directory.
19. In PI Data Collection Manager click on **Discover Data Source Contents** again. There still will be an error, but PI Connector certificate now appears in KEPServerEX OPC UA Configuration.
20. Back on PIINT01 open the OPC UA Configuration Manager from KEPServerEx system tray . Select **Trusted Clients** tab. Right-click on the OPCUA.ConnectorHost certificate and **Trust**.

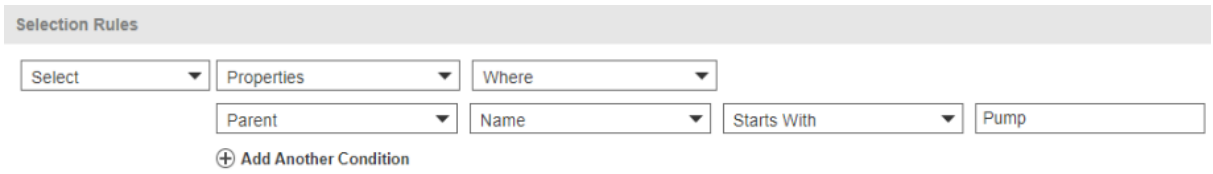



**Note:** OPC UA servers from various vendors have different ways to trust PI Connector's certificate. You must consult the OPC UA server manual.

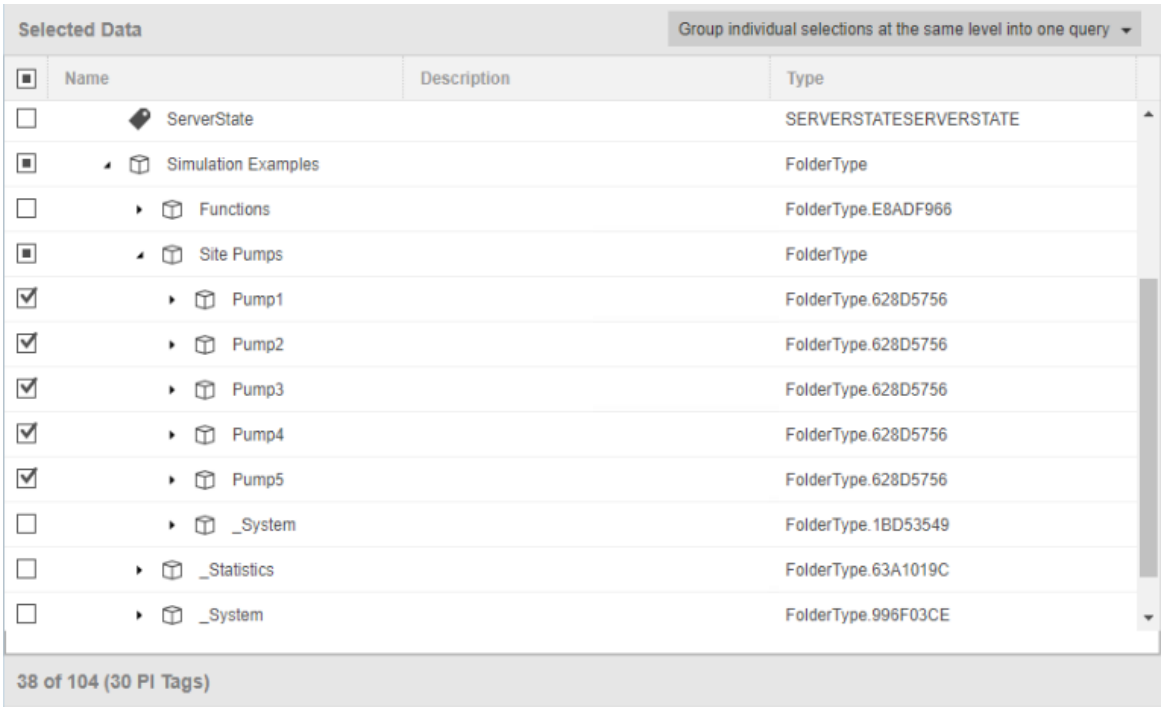
21. Back in PI Data Collection Manager, click again **Discover Data Source Content**. The error is now gone and **Select Data** button is enabled. Click on it.

This opens data selection window. There it is possible to select data manually from address space tree (the selection is limited by Root Node ID specification in PI Connector Data Source configuration) or build queries for selection.

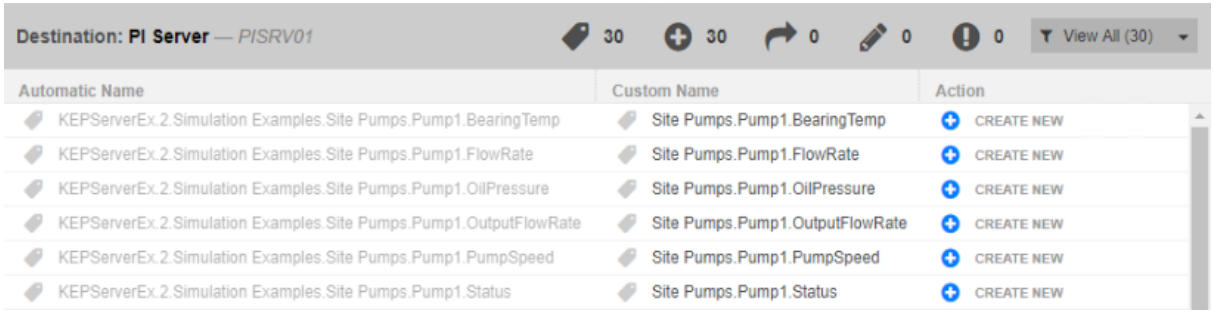
22. To select all our Site Pumps there are two options:
    a. Build a query using **New Selection Query** button. For example, like this:



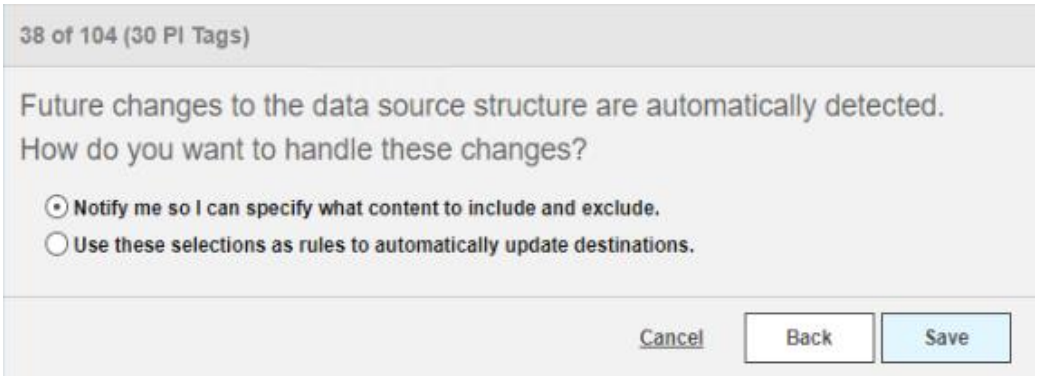

    b. Manually pick objects that should be replicated to PI Server. Expand KEPServerEx node to Simulation Examples → Site Pumps and check Pump1 to Pump5.




23. Both options return result 38 of 104 object selected and 30 PI Tags will be created. Click **Next** to continue to Tag Name Configuration.
24. Instead of Automatic Tag Name Configuration select **Custom** and then **Export** tag naming worksheet.
25. Open it on server with MS Excel (PISRV01 or PISRV02) and copy Automatic Name column to Custom Name column. Using Replace function (CTRL+H) remove the prefix "**KEPServerEx.2.Simulation Examples.**" from tall tags. Save the file.
26. Back in PI Data Collection Manager **Import** the edited worksheet.
27. Click **Check Tag Names** to check whether tags with same names do not already exist in PI Data Archive.




28. Click **Next** to go to Summary page. There are two options how future changes can be handled:

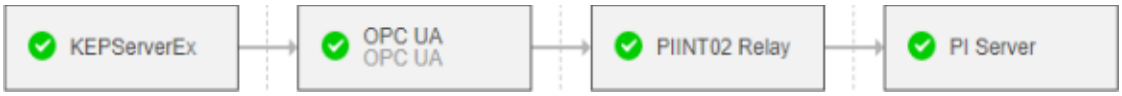



**Notify me so I can specify what content to include and exclude**: If this option is selected, when new assets are discovered, the notification count  in the upper-right area of the screen updates. From any page in PI Data Collection Manager, click the notification count to view notifications and then click the notification to go to the data selection screen, where you can update your data selection queries as needed.

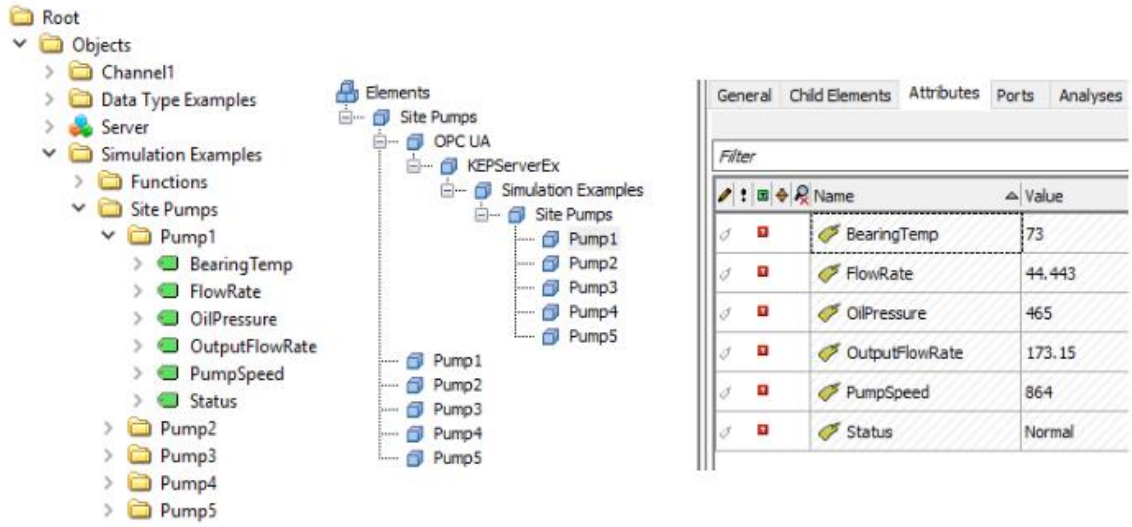
**Use these selections as rules to automatically update destinations**: As with previous option, when the data source structure changes, the existing data selection query runs, and any new elements and assets that are found withing the query are automatically added. However, no notifications are posted, so you must manually check the structure for changes.

29. Keep the selection to be notified and click **Save**.

    PI Connector creates PI points on PI Data Archive and asset structure in PI AF. All is now green in the PI Data Collection Manager main page.




30. Open PSE and Pumps AF database to verify the structure exists and values are updating.



## 12.6 Group Recap Question – PI Connectors

The following questions are intended to reinforce key information presented in this chapter or section.

Answer the following questions:

1. What other type of data can be buffered by PI Connectors and PI Connector Relay internal buffer beside time-series data?

2. Can you administer multiple PI Connectors on multiple servers from a single computer?

3. Why does stopping the PI Connector in PI Data Collection Manager or in PI Connector Administration UI does not also stop the PI Connector Windows Service?

4. How does PI Connector (2nd Gen) authenticate to PI Server?

# 13. PI Security

**Objectives**

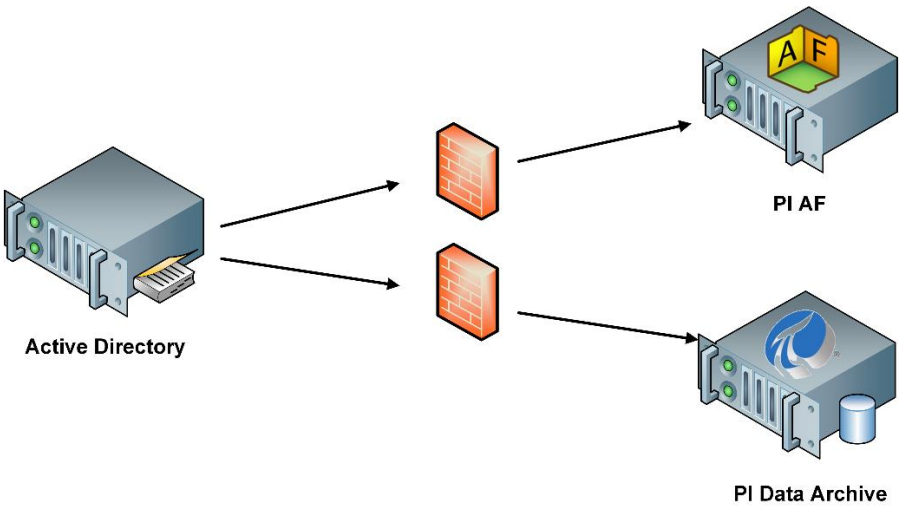
- Describe the function of the PI License in the connecting logic
- Describe how users connect
- Describe the security implications of working on the server box
- Describe the ACL syntax and how access is granted
- Create and manage PI Identities
- Configure PI mappings
- Configure PI database security
- Configure PI point security
- Configure PI security settings slider

For the PI System to be useful people must be able to get critical data to make decisions.

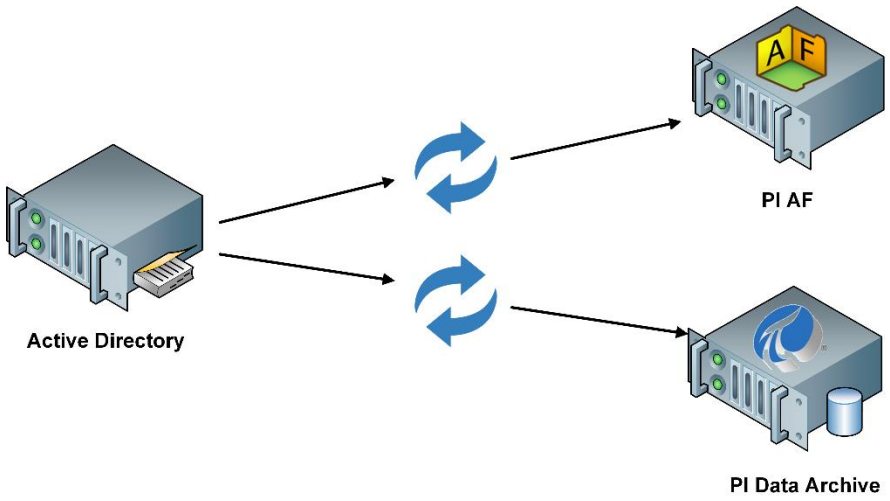
In this chapter addresses the methods and best practices for allowing users proper access to their data on PI Data Archive. PI AF security is not covered in this chapter.

## 13.1 PI Data Archive and PI AF security

A PI System has two components: **PI Data Archive** and **PI AF** server. PI Data Archive as well as PI AF Server (starting from PI AF server 2.7 version) **cannot be directly** tied to Active Directory.




In the PI Data Archive, security can be set on a PI Database or PI Point level. In PI AF on Server level, Databases, Elements, Analysis, Templates etc. using ***PI Identities and AF Identities***. ***Mappings*** (or if necessary, ***PI Trusts*** on PI Data Archive; PI AF uses only WIS) to grant connections access and to decide with which PI Identity or AF Identity that connection is associated.




**Caution**: Do we do not recommend using ***PI Users*** and ***PI Groups*** as they are less secure than ***PI Identities***.

Having logged into the computer via corporate windows account, the CEO is automatically authenticated on PI Data Archive or PI AF. To accomplish this, PI System uses mapping of users' active directory group to a PI Identity or AF Identity. The identity specifies the read/write permissions for accessing both configuration settings and data on PI Data Archive and on PI AF.

## 13.2 The PI License Subsystem

The PI License subsystem has the capability to determine what applications are allowed. However, as of the current version the only elements that are metered is the point count, collective members and percentage match to the machine configuration from the MSF file which should have not drop down below 50% match.

Applications may generate messages such as:

```
0 pinetmgr 20-Oct-05 11:34:07
>> License Warning (non-fatal): [-12221] Not licensed to use this client
application. Level: 3 Process name: PointBuilder.exe(5676) ID: 51
```

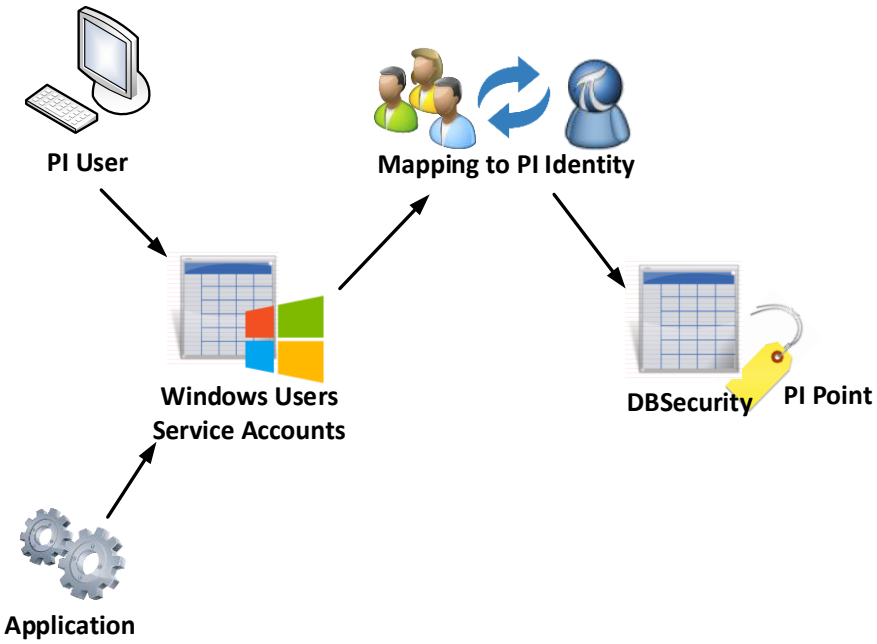
You do not need to respond in any way – these are informational.

You can use the **Operation** → **Licensing** plug-in in PI SMT to view your license statistics.

Even though it is not used, a connection will always check with the PI License Subsystem while connecting.

## 13.3 Connection Logic

The diagram on the following page illustrates the connection logic for users and applications:




**Tip**

You should avoid using the native PI Users and PI Groups in your security scheme. They require explicit login with username and password, and they are inherently insecure and should be used only in legacy systems. Also avoid using PI Trusts for applications.

## 13.4 Working on PI Data Archive

A default loopback PI Trust (**!Proxy_127!**) exists on the PI Data Archive that grants ***piadmin*** access to all applications running locally on the PI Data Archive machine.

By default, no login is required to use the command prompt utilities on the PI Data Archive (piconfig, piartool, pidiag, etc.). You can force a login by modifying the **CheckUtilitylogin** tuning parameter on the PI Data Archive (**Operation** → **Tuning Parameters** → **Security** tab in PI SMT).

Each connecting user or application is associated with a PI Identity and that PI Identity has permissions explicitly granted by an ACL.

### 13.4.1 Granting Access – the ACL

An Access Control List (ACL) string defines the access permissions for that entry.

Choices are “read” and/or “write.”

The syntax is:

```
Identity1:A(r,w) | Identity2:A(r)
```

Where *Identity1* can read and write and *Identity2* can only read.

### 13.4.2 Granting Access - Application

You will apply the ACL in general in two places: Database Security or PI Point attributes.

In the Database Security table, you grant access to data structures on a global level. You can also apply access permissions on a point-by-point basis.

## 13.5 The Database Security Table

In other applications, such as Microsoft SQL Server, each object has their own security control settings. There is the same concept in PI Data Archive, and all the database tables are collected in one place. These are shown in the **Security** → **Database Security** plugin in PI SMT.

**Note:** You must have WRITE access to the PIPOINT database to **create** new points. You do not need this privilege to edit the configuration or data of the points once created. That access is controlled on PI point security level (point security and data security attributes).

The table below lists the common database tables used for user access:

| Database | Controls… |
|---|---|
| PIDS | Access to Digital States and Digital Sets. |
| PIModules | Access to the Modules. This is an important option as some programs (such as PI ICU) and users may require PI Module Database access. |
| PIPOINT | Top-level access to Points, Points Classes and Attribute Sets. Editing existing PI points can be provided through the PI point security configuration on a point-by-point basis. |

For more information, see *PI Data Archive Security Configuration Guide*.

## 13.6 Group Discussion – Default Security

How secure do you want to be?

**Activity Objectives**

- Determine how security is applied

**Approach**

Open PI SMT. Examine the **Security** → **Database Security** table. Also, examine the security settings for the point **SINUSOID** in Point Builder.

Use these resources to answer the following questions:

1. What are the default security permissions for the objects in the DB Security table (hint – there are two)?

2. Why would no read access be provided for PI World identity for some of the tables?

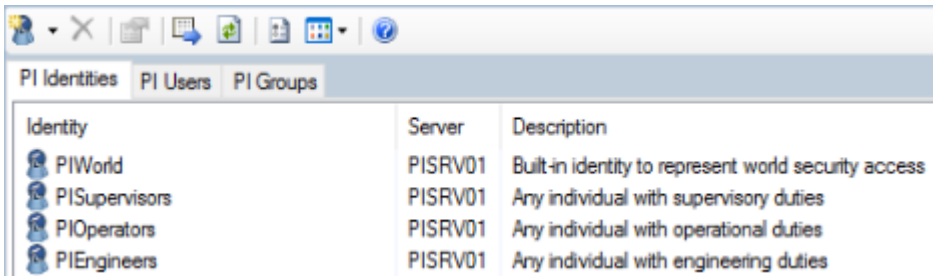
3. What is the default security setting for the PIModules database table?

4. What scenarios can you think of that would require different settings for PI Point security?

5. One table allows PIWorld **write** access. Which one is it, and why would it be configured that way?

## 13.7 Managing PI Identities and Mappings

The **Security** → **Identities, Users, & Groups** plug-in in PI SMT allow you to create and manage PI Identities, PI Users and PI Groups. Several entries are created by default during installation. When configured, you can use the security entries to specify security settings throughout the PI Data Archive, including PI databases security and PI point security.




**Tip**

Ideally, you will have one PI Identity for each Active Directory Group of users, you will use to access the PI System. And PI Identity for service account of applications requiring same level of access.

### 13.7.1 About PIWorld Identity

The **PIWorld** identity represents the Everyone concept of Windows; it specifies the rights of non-explicit users or groups. All authenticated PI Server users automatically get the access permissions defined for PIWorld (in addition to any other access permissions they have been granted).

By default, PIWorld is granted read access to most PI Server databases and objects. You can change the access permissions granted PIWorld, but you **cannot delete** this identity. The PIWorld identity cannot be used in a mapping or a PI Trust.

You can **disable** PIWorld. If you do that, then users no longer get PIWorld access along with their explicitly granted access permissions. This is a recommended action for fresh installations of PI System. This can be risky for PI System upgrades. You might be relying on PIWorld access in several places without knowing it.

### 13.7.2 piadmin vs. piadmins

PI User **piadmin** is a "**GOD**" local user that have unrestricted access to whole PI Data Archive, no matter is it is defined in database or PI point ACL. Avoid mapping piadmin to Windows user or service account.

PI Group **piadmins** is treated as any other PI Identity. If it is removed from PI points or database ACL, it loses access. PI Data Archive administrators should be mapped to it.

## 13.8 Exercise – Tightening the PI Data Archive Security

This activity is designed to maximize learning in a specific topic area Your instructor will have instructions and will coach you if you neec assistance during the activity.

**Exercise Objectives**

- Set Database Security that all applications are accessing PI Data Archive using Windows Integrated Security
- Assign minimum permissions
- Edit PI Points ACL's in Point Security and Data Security attributes using PI Builder
- Disable PI Trusts using PI Security Slider
- Disable PIWorld access

**Problem Description**

Security was configured during different PI System applications installations. But some PI Identities do not have the least possible privileges, they are connecting via PI Trust or PIWorld identity is utilized. Your assignment is to tight the PI Data Archive security so no PI Trusts are used for connection and PIWorld for authentication and therefore they can be disabled. And service accounts have only the minimum privileges assigned necessary for their functionality.

**Approach**

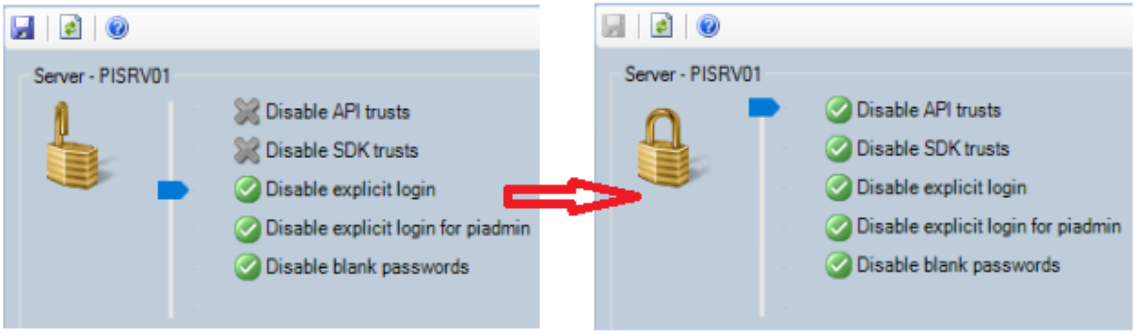
1. Examine Network Manager Statistics plug-in in PI SMT if there is any connection using PI Trust. There should be **PIAnalysisProcessor.exe** application connection under **piadmin** using local **!Proxy_127!** PI trust.
2. Create a new PI Identity **PI Analysis** and map it to **PISCHOOL\SVC-PIANALYT$** gMSA used for PI Analysis Service.
3. Assign this PI Identity to **PIDS** and **PIPOINT** databases ACL with *Read* permissions.

No need to assign it to any existing PI Points. PI Analysis Service needs be assigned only in PI Points it writes calculations results into.

4. Restart PI Analysis Service in Services Manager and check Network Manager Statistics again. It is now connected using WIS under PI Analysis identity and there is no PI Trust utilized.

Now it is possible to disabled usage of PI Trusts using the **"Security Slider"** in **Security → Security Settings**. In a good security environment, you will set the slider to a minimum of explicit logins disabled, which is also a default setting after PI Data Archive installation. This should not affect you at all if you avoid using PI Users and PI Groups.

5. Move the security slider to the top position and hit Save. As informed, you need to restart PI Base Subsystem to apply the change.




Now when PI Trusts are disabled, let's focus on the permissions for different applications. PI Vision, PI Connector Relay and PI Data Collection Manager and PI Analysis identities have all assigned proper permissions on Database Security level. **PI Buffer** and **PI Interfaces** identities are only defined in **PIPOINT** database ACL with *Read and Write* permissions. This is not necessary as neither PI Buffer Subsystem nor PI Interface create PI points.

6. Remove the *Write* access for PI Buffer and PI Interfaces identity from **PIPOINT** database.

For the PI point level security, the settings depend if PI Interface is also using output points and if PI Buffer Subsystem is in place (as it should be at every data collection node) as it is shown in the table below:

| PI Identity | | Point Security | Data Security |
|---|---|---|---|
| PI Buffer | | read | read, write |
| PI Interfaces | no output points | read | -none- |
| | with output points | read | read |
| | no buffering | read | read, write |

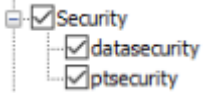
There is no PI interface on PI Data Archive and it does not utilize output points, so for PI Interfaces identity the minimum privileges are *Read* access to point security and *no access* at all needed for data security as PI Buffering is writing values to PI points.

Also, PI points for OPC DA, Random Simulator Data and RampSoak Simulator Data interfaces were created before PI Vision was installed and PI Vision identity was created, yet in PI Vision it is possible to search for those points and display their data.

Why? _____

Use PI Builder MS Excel add-in to modify the PI points security attributes.

7. In PI Builder load all PI points, but in column headers selection, select only datasecurity and ptsecurity attributes.



There are 3 groups of PI points with different ACL's. 3 internal points which should not be modified, PI points created by PI Connector Relay and then PI points for Random, RampSoak and OPC DA interfaces.

8. Use Filter from MS Excel Data tab  to select the ACL group of default PI points (Random and RampSoak interfaces) and Pumps points for PI Interface for OPC DA. The combination is the same for point and data security attributes:

piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffer: A(r,w) | PIWorld: A(r)

9. Modify the security of the tag in the first row:

   Data Security:

   piadmin: A(r,w) | piadmins: A(r,w) | PI Buffer: A(r,w) | PI Vision: A(r) | PI Analysis: A(r)

   Point Security:

   piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r) | PI Buffer: A(r) | PI Vision: A(r) | PI Analysis: A(r)

10. Apply the new security settings to the rest of the filtered rows and Publish the changes in Edit Only mode.

Not really necessary to edit the security of PI points created by PI Connector Relay as PI Vision identity is already included. You may remove the PIWorld, PI Buffer and PI Interfaces identities from data and point security as they do not access these points.

Now it is possible to disable the PIWorld identity.

11. Open PIWorld identity properties and check **Identity is disabled**.

In Network Manager Statistics the PIWorld identity is still present assigned identities for connected applications and we need to remove it.

| | | | |
|---|---|---|---|
| PI Analysis \| PIWorld | PISCHOOL\SVC-PIANALYT$ | PI Analysis | PISCHOOL\SVC-PIANALYT$ |
| PI Buffer \| PIWorld | PISCHOOL\SVC-PIBUFFER$ | PI Buffer | PISCHOOL\SVC-PIBUFFER$ |
| PI Connector Relay \| PIWorld | PISCHOOL\SVC-PIRELAY$ | PI Connector Relay | PISCHOOL\SVC-PIRELAY$ |
| PI Interfaces \| PIWorld | PISCHOOL\SVC-PIINT$ | PI Interfaces | PISCHOOL\SVC-PIINT$ |
| PI Interfaces \| PIWorld | PISCHOOL\SVC-PIINT$ | PI Interfaces | PISCHOOL\SVC-PIINT$ |
| PI Interfaces \| PIWorld | PISCHOOL\SVC-PIINT$ | PI Interfaces | PISCHOOL\SVC-PIINT$ |
| PI Vision \| PIWorld | PISCHOOL\SVC-PIWEB$ | PI Vision | PISCHOOL\SVC-PIWEB$ |

To force applications to reconnect, PI Data Archive can be put into standalone mode which closes all connections beside PI Data Archive subsystems.

12. Open CMD, navigate to **%PISERVER%\adm** and use command **piartool -sys -standalone on** and then **piartool -sys -standalone off**.

```
D:\Program Files\PI\adm>piartool -sys -standalone on
Stand alone mode completed: [0] Success
Current mode: StandAlone

D:\Program Files\PI\adm>piartool -sys -standalone off
Stand alone mode completed: [0] Success
Current mode: Normal
```

After a short moment the applications reconnect without PIWorld permissions.

13. Use **Data** → **Current Values** plug-in to verify values are updating after reconnection and you can search and visualize default and Pumps points in PI Vision.

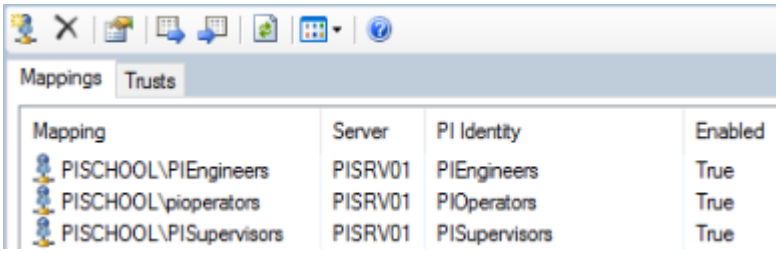
## 13.9 Exercise – Set Different User Access to PI Points

This activity is designed to maximize learning in a specific topic area Your instructor will have instructions and will coach you if you neec assistance during the activity.

**Problem Description**

You have many users requiring access to your PI System, but they all have different tasks to do that require different levels of access to the points. Therefore, you want to grant access to the PI Data Archive and its resources-based user roles.

| PI Identity | Domain Group | User Display Name | User Logon Name | Password |
|---|---|---|---|---|
| PIEngineers | PISCHOOL\PIEngineers | Eric Engineer | eengineer | eengineer |
| PIOperators | PISCHOOL\PIOperators | Olga Operator | ooperator | ooperator |
| PISupervisors | PISCHOOL\PISupervisors | Sue Supervisor | ssupervisor | ssupervisor |




You need to create a security structure that enforces the following business rules:

- The point **BA:LEVEL.1** is a sensitive financial calculation and its data should only be visible to the plant Supervisors (PI Supervisors Identity).
- The **BA:CONC.1** point needs to be able to be written to by your Operators (PI Operators Identity). It can be read by anyone.
- Your engineers (PI Engineers) need to be able to edit the attributes of all of the **BA:*** points (**BA:ACTIVE.1**, **BA:CONC.1**, **BA:LEVEL.1**, **BA.PHASE.1**, **BA:TEMP.1**).
- The point **BA:ACTIVE.1** should only be visible for plant Engineers group. No one else should not be able to find the point.

**Approach**

Map the Domain Groups to the PI Identities according the table in Problem Description section.

Add PIEngineers, PIOperators and PISupervisors identities to the **PIPOINT** database ACL with *Read* access.

Why it is necessary? _____

Use the following table to write down the security settings for each "BA.:" point:

| PI point | PIEngineers | | PIOperators | | PISupervisors | |
|---|---|---|---|---|---|---|
| | Point sec. | Data sec. | Point sec. | Data sec. | Point sec. | Data sec. |
| BA:ACTIVE.1 | | | | | | |
| BA:CONC.1 | | | | | | |
| BA:LEVEL.1 | | | | | | |
| BA:PHASE.1 | | | | | | |
| BA:TEMP.1 | | | | | | |

Modify the points using Point Builder in PI SMT.

Test your security rules. Here are some suggestions (you may need to log off and on between tests, as different users or use Right Click, **'Run as Different User'** on PI SMT):

1. Log on as someone **not** in the Supervisors group and try to search for the point **BA:LEVEL.1**. What result do you get?

2. Log on as someone **not** in the Engineers group and try to edit a point. What result do you get?

3. Test the operator access to **BA.CONC.1** with PI SMT Data → Archive Editor. Can she add or change data in any other point?

4. Log on as someone **not** in the Engineers group and search for point **BA:ACTIVE.1**. Can you find it?

# 14. Adding Power to the PI System Environment

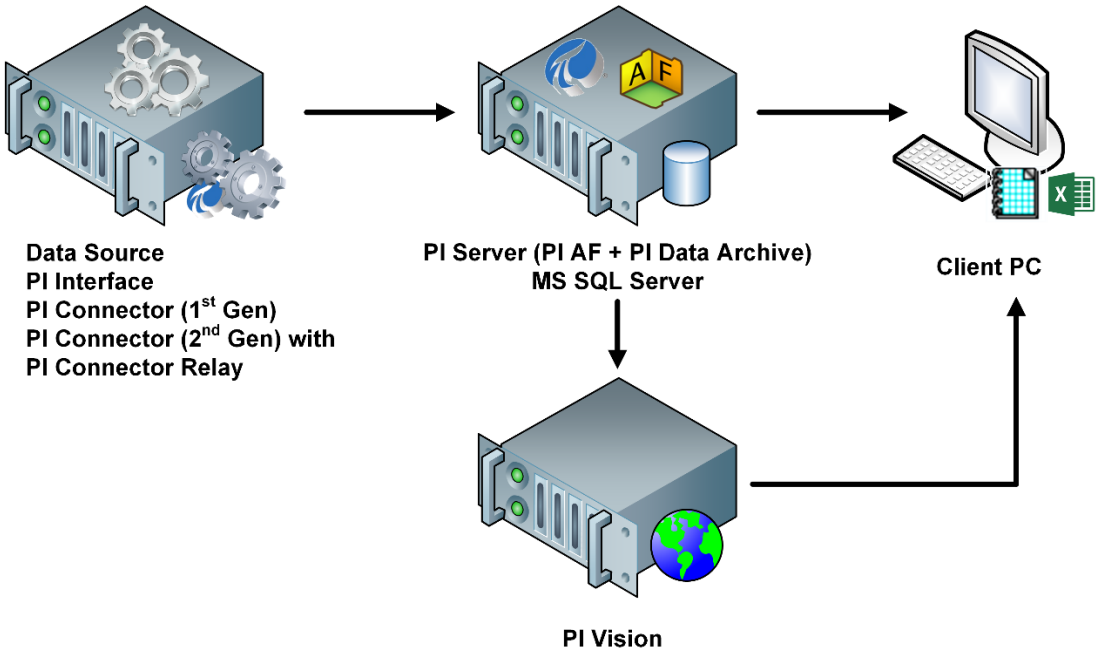
**Objectives**

- Reiterate the components of a small PI system
- Provide further information on the more complex PI systems

## 14.1 A Simple PI System

As we have seen, the simplest possible PI System consists of just tree computers:

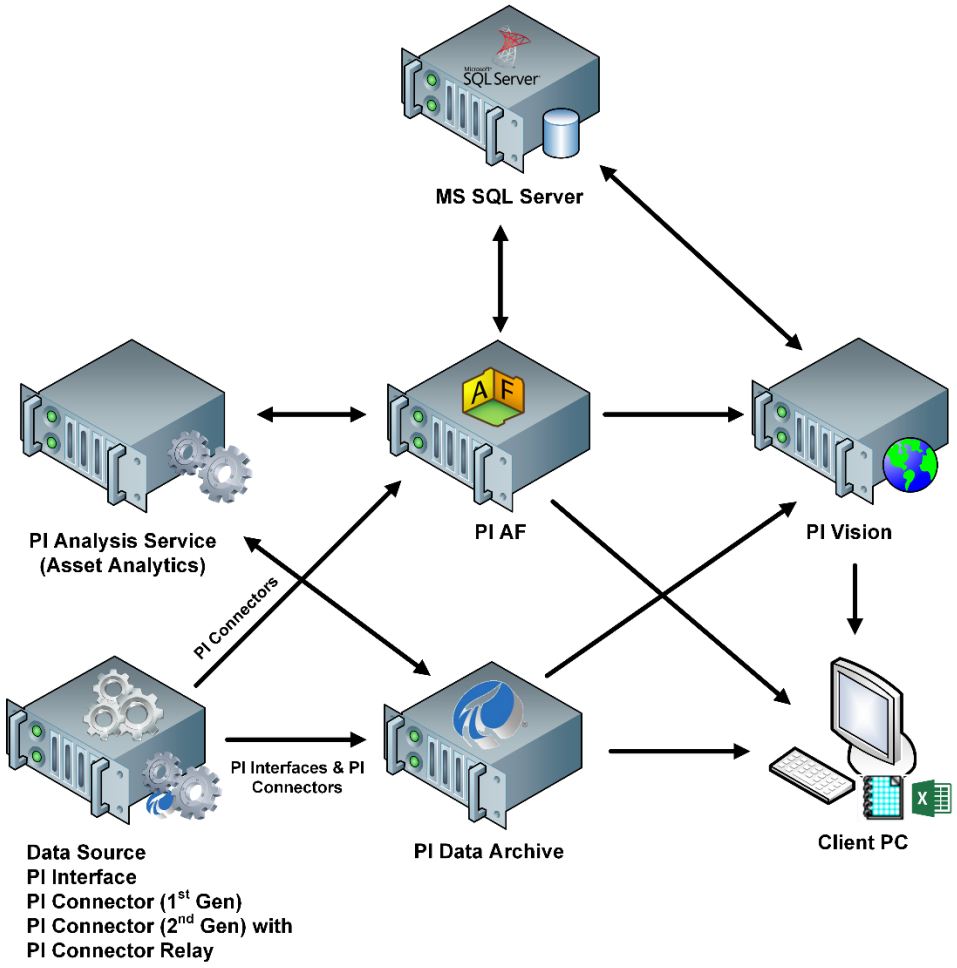
- A single computer for the PI AF and PI Data Archive with MS SQL Server
- A single computer running the PI Interface / PI Connector (1st Gen) or PI Connector (2nd Gen) with PI Connector Relay to collect the real time data.
- A single computer with IIS Web Server role for PI Vision. (In case of small systems PI Vision can be installed together with PI Server and MS SQL Server)




Above, the PI Interface / PI Connector (1st Gen) / PI Connector (2nd Gen) with PI Connector Relay are running as a service on the same computer as the source of the data. As the data is collected, it is transferred in a timely manner to the PI Server. The users then request data from the PI AF and/or PI Data Archive Server as required. Tools such as PI ProcessBook or PI Vision will sign up for updates, so that any new data arriving at the PI Data Archive server is displayed on the client as it is received. There is no need for manual refreshing of the displays.

## 14.2 Expand PI System Components as Needed

As the needs of the PI environment grow, multiple components are added to expand the capabilities of the systems. If the PI AF server becomes a bottleneck for the system because of its extensive size, the PI AF server may *be moved to a separate* server and MS SQL Server also split from PI AF. When sophisticated calculations are applied, the PI Analysis Service can also *be moved to a separate server* not to compete with resources on PI AF Server. This may result in a PI system configuration as shown below. The arrows indicate the direction of data flow.




## 14.3 When You Need to Bet Your Business on PI System

There may come a time when the PI System environment is vital to the running of the company. At such a time, it may be necessary to evaluate a Highly Available (**HA**) PI System. The PI System includes features that facilitate making data highly available. These include the ability to conduct online backups, the distributed nature of data collection within the PI System, and the availability of fault-tolerant third-party solutions that provide redundant hardware solutions. The PI System environment has been designed to provide fault tolerance via server replication and PI Interface / PI Connector (1st Gen) failover.

### 14.3.1 PI Data Archive Server Replication

The PI High Availability (HA) design provides for multiple PI Data Archives, each acting as an independent storage for the time series data. These PI Data Archives function as a unit called a collective. A PI Collective has two types of servers:

- Primary - the main server in a collective where configuration changes are made
- Secondary - the remaining servers in a collective

  These servers automatically adopt configuration changes made on the primary but receive data from the data source individually via a technique called N-way buffering, explained later.

### 14.3.2 PI AF High Availability Options

There are several high availability (HA) options for PI AF, both for the Microsoft SQL Server that hosts the PI AF database (PIFD) and for the PI AF application server itself.

MS SQL Server options:

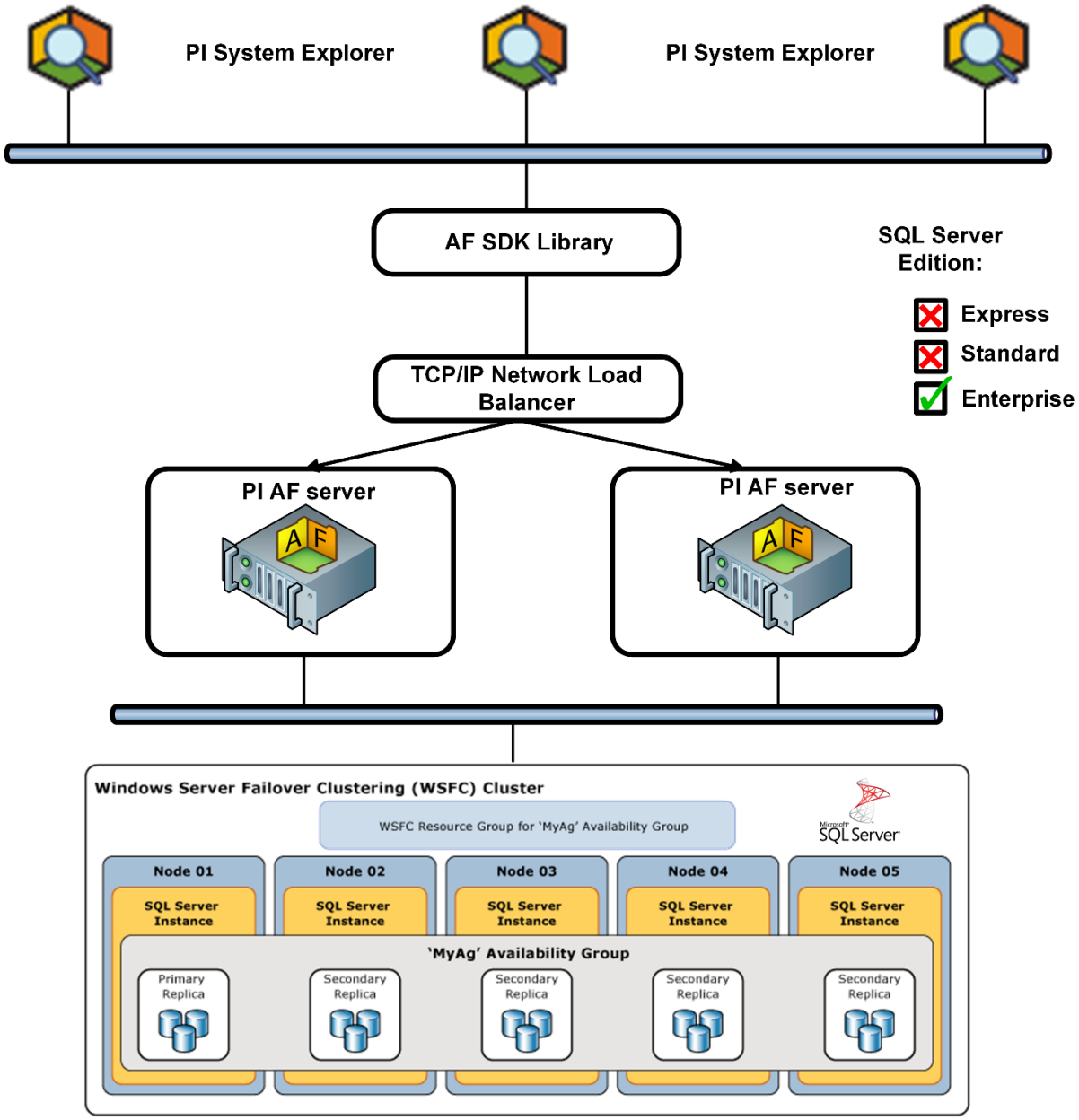
- **AlwaysOn Availability Group** (preferred option)
- Clustered
- Mirrored
- Transactional Replication (with AF Collective)

PI AF application server options:

- Clustered
- **Load Balancing** (preferred option)
- AF Collective

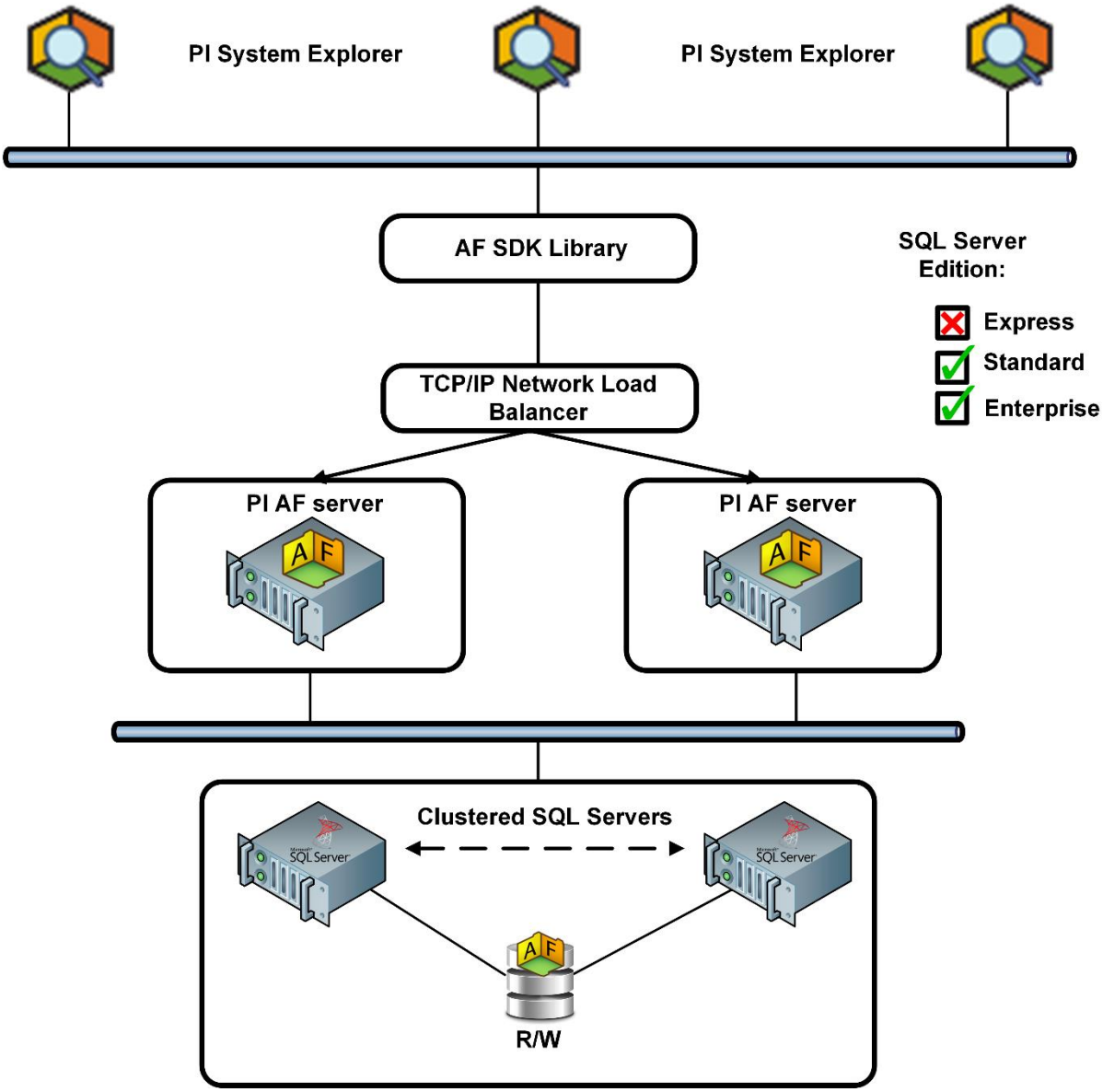
The preferred and recommended option currently is to use the Network Load Balancer over set of PI AF servers. Those PI AF servers have redundant SQL back-end that is using of the options (preferred AlwaysOn Availability Group, Clustered SQL server or Mirrored SQL servers).

**Load Balanced PI AF servers with MS SQL Server AlwaysOn Availability Group**

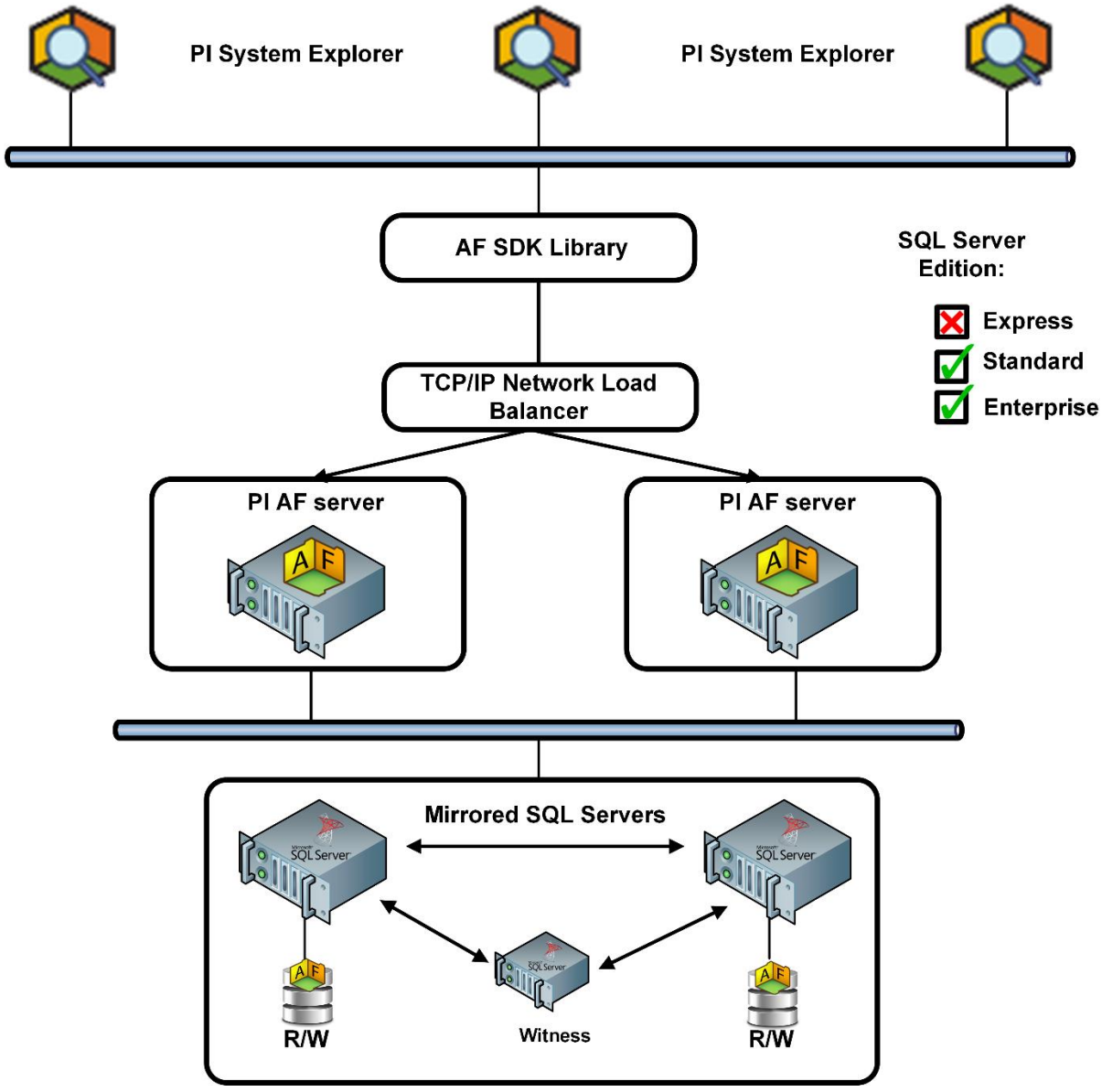



For the overview of Always On Availability Groups for MS SQL Server, see the article https://msdn.microsoft.com/en-us/library/ff877884

## Load Balanced PI AF servers with MS SQL Server cluster



**Load Balanced PI AF server with mirrored MS SQL Servers.**




**Note: PI AF Collective Warning!**

PI AF Collective is not a recommended solution now as applications that require writes to the AF Configuration database (PI Asset Analytics and PI Notifications), or applications that write Event Frames will NOT work when the PI AF Collective primary server is unavailable, as secondary AF Collective members are Read-Only.

For additional details see High Availability options for PI AF knowledge article.

### 14.3.3 PI Interface & PI Connector Buffering and Failover

The PI Interface buffering service writes time-series data directly to all members of the collective, buffering data temporarily for those unable to receive data for a period. This mechanism assures that time-series data stored in each current archive is an exact duplicate of the other current archives in the collective.

The PI Connector and PI Connector Relay buffering is internal and installed with it. No configuration is necessary. Many PI Interfaces and several PI Connectors (1st Gen) incorporate failover mechanisms that allow for redundant data collectors. If one copy of the PI Interface or PI Connector (1st Gen) shuts down, then another will take over the data collection.

PI Connectors (2nd Gen) and PI Connector Relay do not support failover.

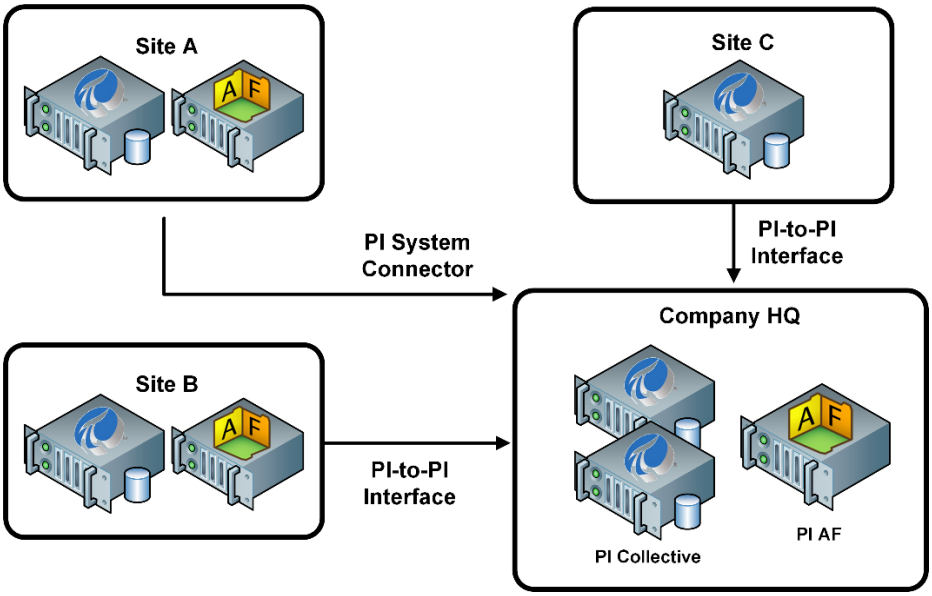
### 14.3.4 PI Visualization Suite

PI Visualization Suite (PI Vision, PI ProcessBook, PI DataLink) can automatically switch from the PI Data Archive server to any of the replicated servers in the event connection to current member is unavailable; guaranteeing that all clients always have read-access to PI System data.

### 14.3.5 Connecting multiple PI Servers

It is possible to combine data from widely distributed PI System sites to a corporate PI System. This allows corporate access to designated data on the distributed site PI Collectives without interfering with the site servers.

Many companies do not have only one PI Server, especially global companies. One potential architecture creates a PI Server for each site and a central PI Server that collects aggregated data from each site. Either each site can have its own PI AF server or use centralized PI AF that serves multiple PI Data Archives.



The **PI-to-PI Interface** allows the aggregation of this data by copying data from one PI Data Archive and making it available on another server. When a corporate PI Data Archive is connected via the PI-to-PI Interface to remote site PI Data Archives, then required site data (such as KPIs) can be presented (with history) at the corporate level. Dissemination of selected data from the remote sites via the corporate PI System to corporate users becomes possible.

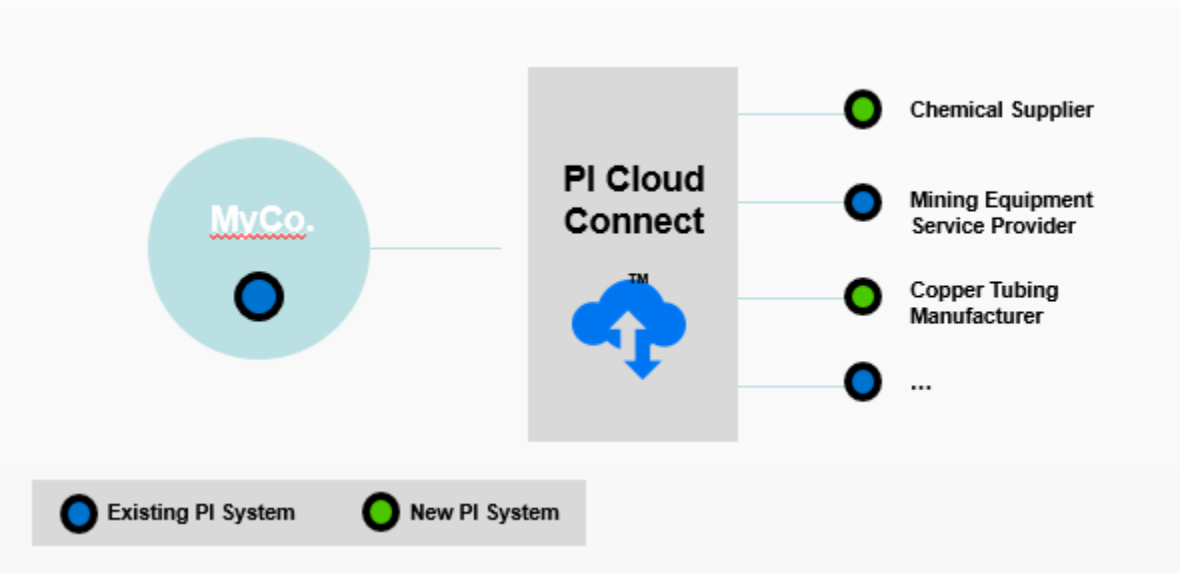
The **PI System Connector** has similar functionality, but more advanced.

- It reads the PI AF structure from a source PI AF server and sends the objects to a destination PI AF server.
- All PI points referenced within the PI AF object structure and their data are also collected from the source PI Data Archive and sent to the destination PI Data Archive.
- The PI System Connector continually monitors changes on the source PI AF database objects and reflects those changes to the destination PI AF database.
- The PI Point data is initially backfilled to the destination and then monitored for current data changes.

### 14.3.6 PI Cloud Connect

PI Cloud Connect allows the quick, easy and secure exchange PI System data between companies. Instead of providing a spreadsheet or a VPN connection for every vendor or business partner that you work with, a connection via PI Cloud Connect is available to publish PI System data. To select the data you want to share, and just who you want to share it with. Your suppliers can then make a connection to PI Cloud Connect and subscribe to the data feed that you control.
Some vendors may already have a PI System today. If you want to exchange data with a vendor who does not currently have a PI System, OSIsoft has a way to deliver on-prem (on-premises) software based on subscription licensing and pricing.

**PI Cloud Connect for Asset Owner**



### 14.3.7 PI System Access

The PI System Access family of products is designed to support the implementation of custom applications on top of the PI System. PI System Access supports the integration of PI System data with other applications and business systems such as Microsoft Office or SQL Server, Enterprise Resource Planning systems (ERPs), reporting and analytics platforms, web portals, geospatial and maintenance systems, just to name a few.

The PI System Access suite covers a wide range of use cases in various environments, programming languages, operating systems and infrastructures.

| | |
|---|---|
| **PI SQL Framework** | The products in this category expose the PI System as a relational database and are particularly useful where the PI System has a role to play in Reporting or Business Intelligence (BI) scenarios, where it must interact with other products or systems that communicate using the Structured Query Language (SQL) as well as in custom application development. |
| PI OLEDB Provider | The PI OLEDB data provider, together with the PI Server's underlying PI SQL subsystem, implements SQL language to allow relational queries to the PI Data Archive. |
| PI OLEDB Enterprise | PI OLEDB Enterprise is an OLE DB data provider you can use to access asset metadata stored in the PI Asset Framework (PI AF). It includes PI OLEDB Provider. |
| PI JDBC Driver | PI JDBC Driver is a Java Database Connectivity driver that provides robust data access to the PI System. PI JDBC Driver offers much of the same functionality as the PI OLEDB Provider. |
| PI ODBC Client | The PI ODBC Client: used to bring process data into any ODBC-compliant client application. |
| RTQP Engine | New SQL access to PI AF. There is a whole new more powerful data model of PI AF than for PI OLEDB Enterprise. Part of PI Server install kit. Must be installed locally with PI AF server. Currently does not include direct access to PI Data Archive. |
| PI SQL Client | Contains OLEDB, ODBC and JDBC Drivers to access PI AF using RTQP Engine |
| **PI Web API** | The PI Web API provides a resource oriented and HTTP(S) based interaction model to data contained within the PI System. The PI Web API provides a resource oriented and HTTP(S) based interaction model to data contained within the PI System. Although the technologies within the PI Web API are suitable for a broad range of needs, they are primarily targeted at supporting web and mobile application development scenarios. |
| **AF SDK** | Provides programmatic access to PI Server data (PI Data Archive and PI AF). PI AF Code examples and programming references for the AF SDK are available (once installed). AF SDK is available at the developers' site at https://pisquare.osisoft.com/community/developers-club |

# 15. High Availability

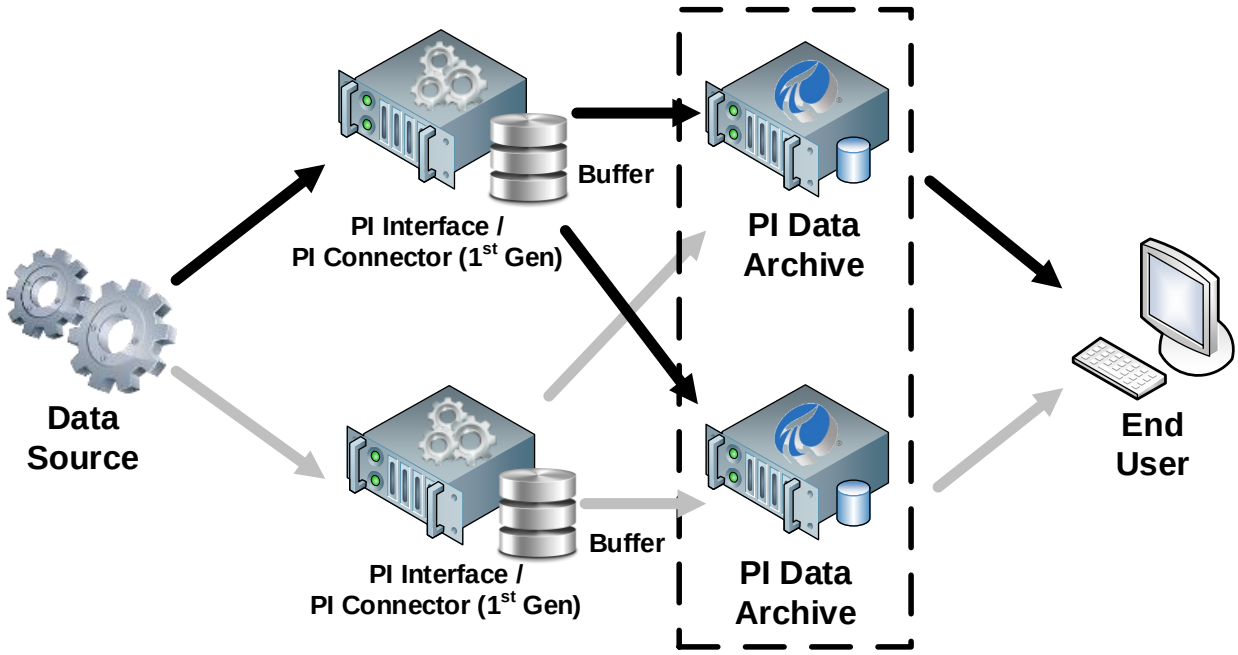
**Objectives:**

- Describe the components of High Availability
- Create a High Availability PI Collective
- Implement PI Interface Failover
- List the limitations of PI High Availability

## 15.1 Components of PI High Availability

High Availability requires three (3) components all working together to function properly. These components are:

- PI Interface / PI Connector (1st Gen) failover and/or N-Way buffering;
- PI Data Archive servers collective;
- PI Client application failover.



## 15.2 PI Data Archive Replication

PI Data Archives replicate by initially cloning the primary source server. All server metadata is replicated.

Tuning parameters are replicated when the PI Collective is created, but then they are independent, because configurations may vary by server machine and thus it should be possible to modify them on each secondary member(s) too. However, they are not replicated when the collective is reinitialized. Message logs are not replicated because the messages are inherent to the specific machine.

PI HA has separate paths for the same time series data for true replication. This is accomplished by replicating the data as it is collected and distributing the data to each server in the PI Collective. Data replication is achieved in the buffering mechanism.

A replication service keeps static, configuration data (point definitions, digital state sets, and so on.) synchronized between the primary and the clones (referred to as secondary nodes).

## 15.3 Constraints

The PI HA mechanism was designed to function in a situation where the servers and interfaces are all contained within the same domain, and that the domain has a coherent structure. By coherent structure, we are implying a functioning domain controller with reliable DNS resolution.

In addition, to have proper initialization of the secondary nodes, there needs to be Windows file copy access between the servers. Open the related TCP ports. Consult Appendix A to get more details.

At the buffering level, data is replicated. Any data sent to the PI Data Archive from a source other than a buffer mechanism is not replicated, only written to a single server. The same situation arises with applications designed to read and write to a single server. Currently these include:

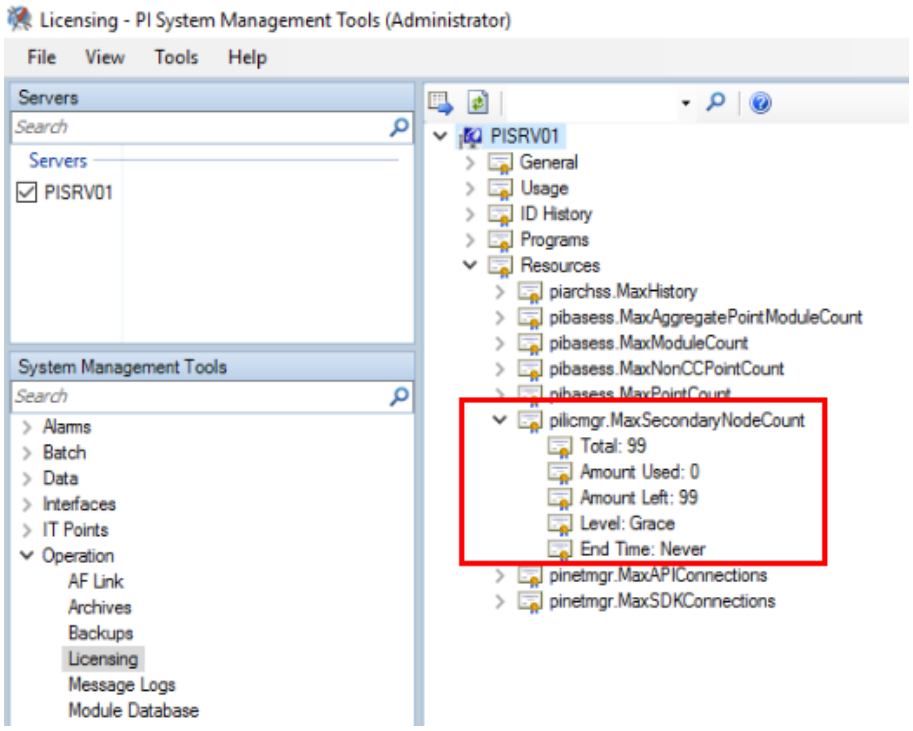
- Performance Equations,
- Totalizer points,
- PI Batch Generator,
- Any custom manual data entry applications that do not use the PI SDK or AF SDK.
- *PI SDK and AF SDK applications can take advantage of PI SDK or AF SDK buffering.*

## 15.4 HA Pre-Installation Checks

### 15.4.1 License

A license file will be required. ***The license file will have to acknowledge at least one secondary server***. The license must be obtained against the Primary PI Data Archive.

Make sure the license supports PI Collectives. Validate this be checking the license information in PI SMT under **Operation → Licensing**. Select Resources from the dropdown and expand *pilicmgr.MaxSecondaryNodeCount* The **Amount Left** should be greater than “0”.




When the PI Collective is formed the license of primary PI Data Archive is copied to all secondary members. Functionality of secondary member of PI Collective is restricted therefore there is no check against machine signature stored in the license.

If PI Data Archive is licensed for PI Collective, the production license already goes bundled with temporary license which is used for soon to be secondary PI Data Archive installation.

If the upgrade to High Availability is done after some time when there is a single PI Data Archive in production, the license needs to be updated to allow PI Collective creation.

### 15.4.2 PI Data Archive version

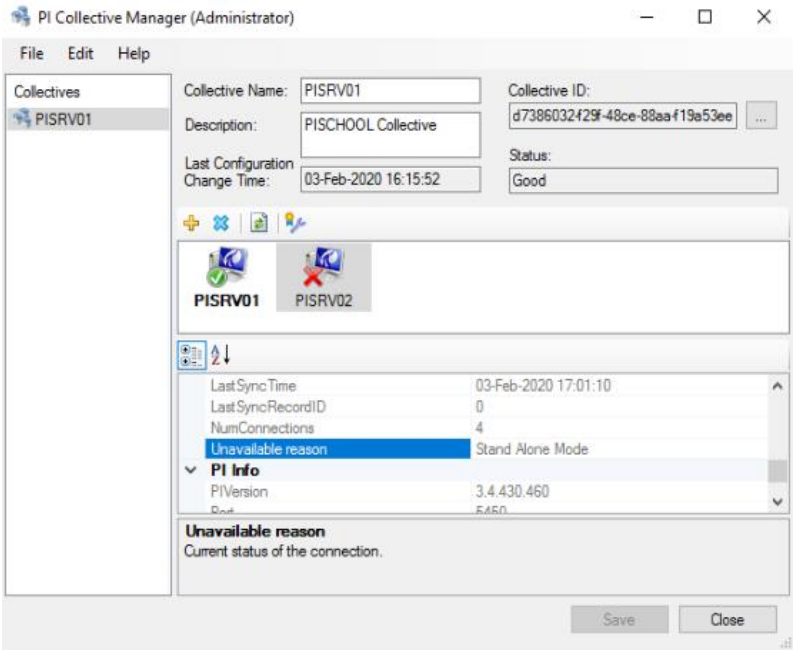
OSIsoft strongly recommends that all members of PI Collective are all the same version of PI Data Archive. Different version may result into unexpected errors and behavior.

Short time version difference for example, during upgrade procedure should not be an issue.

## 15.5 PI Collective Manager

Use the *PI Collective Manager* to create new PI collectives, configure existing collectives and their servers, and view the status of PI Collectives.

An icon in the diagram represents each server in the collective. A green check mark on the icon indicates that the server is communicating properly. A red X indicates that the server is unavailable. A yellow warning icon indicates that the server is available but has errors. **Status** and **Connection Status** show the associated errors.




## 15.6 Group Exercise - HA Step 1: Form the PI Collective

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

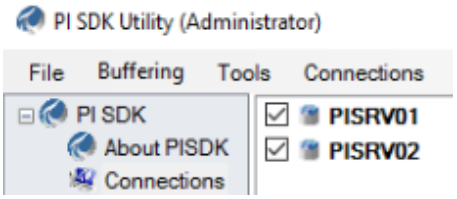
Learn how to create a collective using the PI Collective Manager

### Exercise Description

You need to form a collective by combining two existing PI Data Archives.

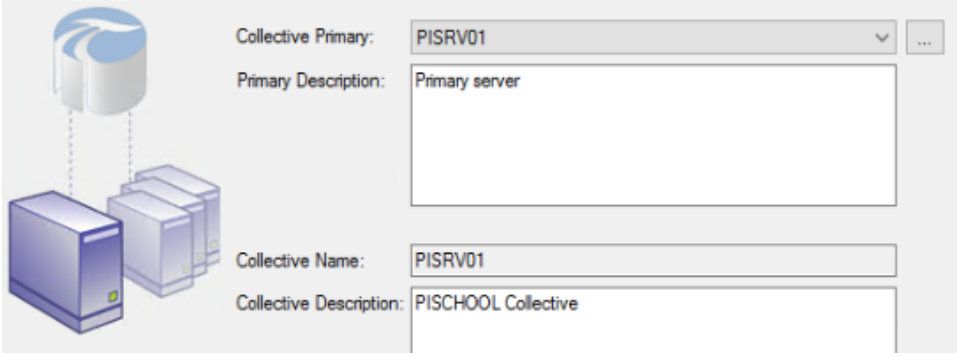
### Approach

1. PI Data Archive of the same version as on PISRV01 is pre-installed on PISRV02.
2. **Add** the future secondary member in the primary server's **PI Connection Manager** in PI SDK Utility on PISRV01 and verify connection.

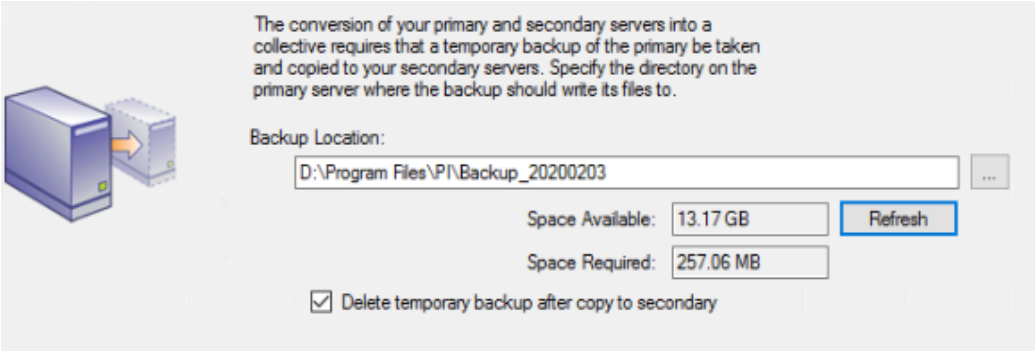



3. Open *PI Collective Manager* from PI System folder in Start Menu.

4. Select **File** → **Create New Collective** which opens a collective creation wizard.
5. Agree with verification of valid backup and PI Interface machines configuration. Click **Next**.
6. Select that primary member is **An existing server that contains data,** because there are already applications connecting to PI Data Archive like PI Interfaces, PI Connectors, PI Buffer and PI Clients. *A newly installed server* should be selected only with fresh installation as this option creates a new PI Data Archive GUID will affect already configured PI Interfaces, PI Clients etc. Click **Next**.
7. On the next page select **PISRV01** to be the Collective Primary. You may add description. Click **Next.**




8. Select PISRV02 from drop-down list as secondary server and click **Add**, then **Next**.
9. Keep all archives selected in the list that will be copied to secondary server and click **Next**.
10. PI Collective Manager will create temporary backup that will be copied and restored on PISRV02. You may select the location and if you want the backup to be deleted after copying. Keep the defaults and click **Next**.




11. On the next page you can click on Review all settings where it is possible to change descriptions or name the PI Collective. Click **Next**.

This starts PI Collective creation procedure. PI Collective Manager will:

I. Code the collective creation.
II. Backup the primary server.
III. Shutdown the secondary server.

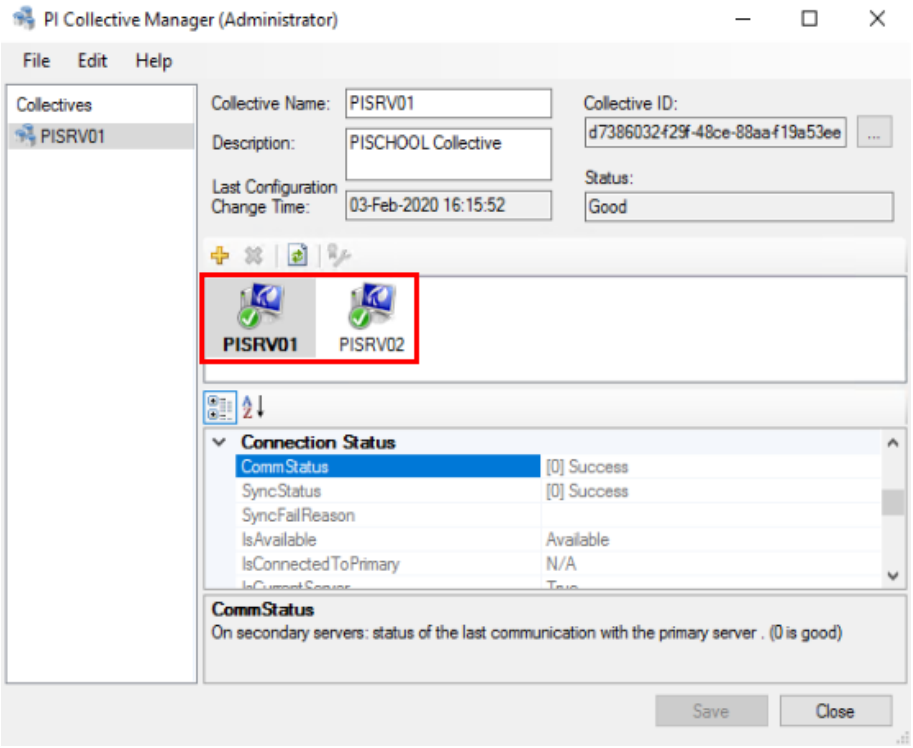
IV. Replicate to the secondary server form primary server backup.

V. Restart the secondary server.

PI Collective creation may take some time depending on how many archives were selected to copy, the size of the archives, and the network bandwidth.

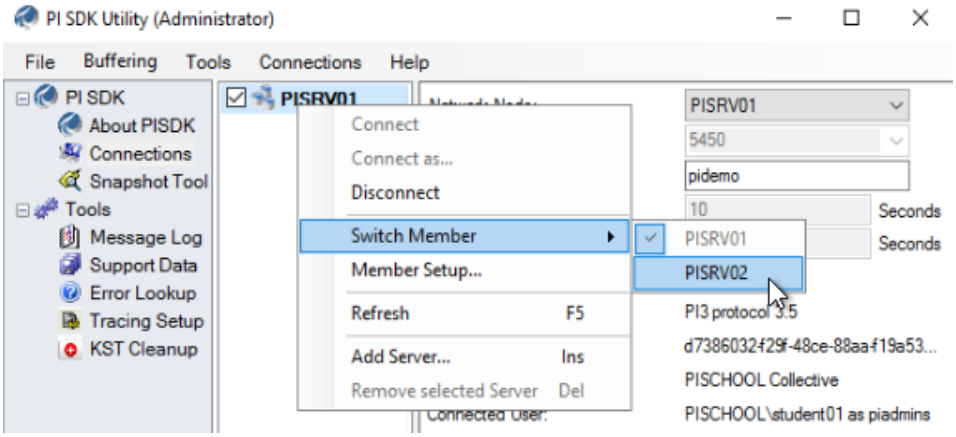
12. When collective is successfully created click **Finish**.

When this process is complete, you should see two PI Data Archives with green check icons:




If you do not see these icons, try to **Reinitialize the Collective**.

You should use the **PI Connection Manager** to verify that you have a Collective and not two individual PI Data Archives. Notice the change of the icon to . Test switching the connection.



## 15.7 Group Exercise - HA Step 2: Testing N-Way Buffering

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

Learn how to test N-Way buffering.

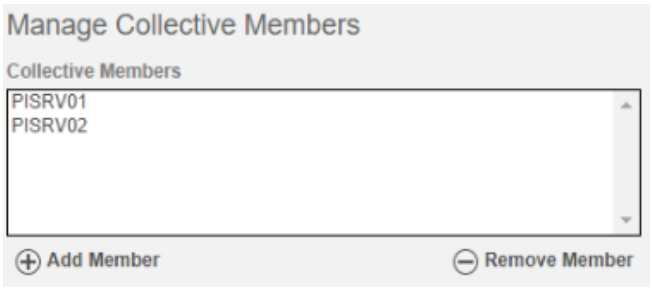
### Exercise Description

Look at the pump point data on the secondary server coming from PI Interface for OPC DA Is it being updated? Why?

Look at the pump point on the secondary server coming from PI Connector for OPC UA through PI Connector Relay. Is it updating?

### Approach

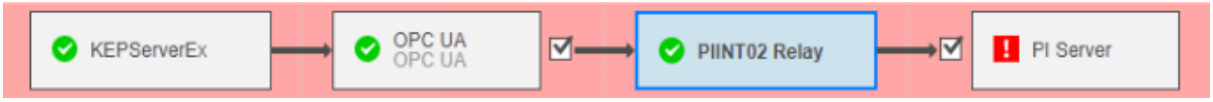
1. If the connector's points are not updating on secondary server, open PI Data Collection Manager and click on destination PI Server. In Configuration tab should be expanding field More, which shows Manage Collective Members section:



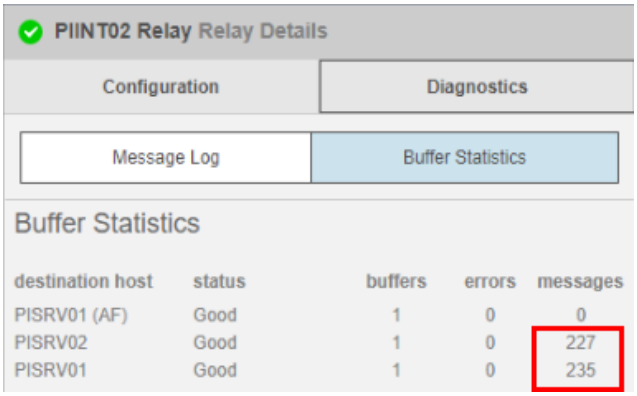

2. If the More button is not there. Try following:
3. Restart PI Connector Relay. Click on the Relay. In Configuration tab click on ⋮ next to **Edit Settings** button. **Stop** the Relay and then **Start** it. Refresh the browser.
4. If this does not help, then remove the destination PI Server in Configuration tab by clicking on ⋮ and **Delete**. Then add the destination PI Server again. It picks up the collective configuration and start updating values on secondary server.
5. On PIINT01 use **pibufss –cfg** to verify the connections to the PI Data Archives or use the Buffering Manager from the PI ICU.
6. Use **pibufss –ss** (Compression and buffer statistics) to check the amount of data going to the PI Data Archive servers.
7. Break the connection between the data collection node and PI Data Archives using the standalone mode (**%PISERVER%\adm>piartool -sys -standalone on**) and use **pibufss -qs** to show the accumulation of data in the buffer or see the Buffering

Manager.

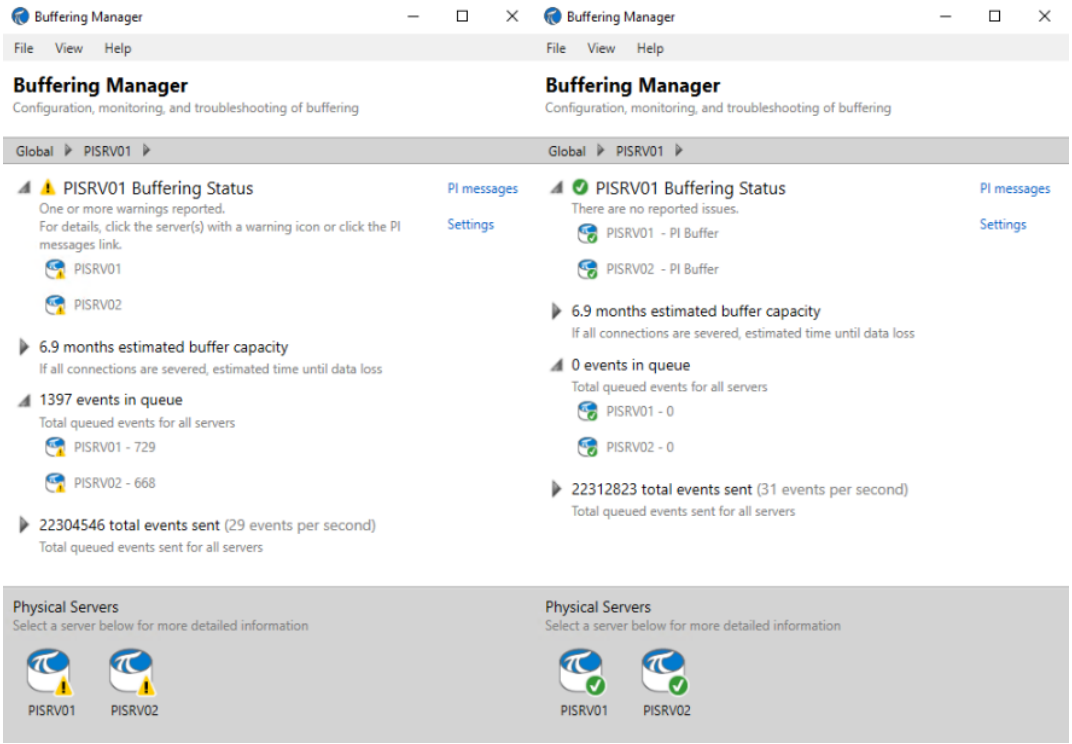
8. In PI Data Collection Manager error is displayed only if the connection to primary PI Data Archive is interrupted:




9. To check the buffer statistics for PI Connector Relay, click on Relay. Go to **Diagnostics → Buffer Statistics**. Messages (same as events in PI Buffer Subsystem statistics) are increasing.




10. Restore the connection between data collection node and the PI Data Archives (**%PISERVER%\adm>piartool -sys -standalone off**) and verify that buffered data is sent to the servers. Errors such are those should resolve to show a screen shot like:



## 15.8 Group Exercise - HA Step 3: Prepare the second data collection node

This activity is designed to maximize learning in a specific topic area Your instructor will have instructions and will coach you if you neec assistance during the activity.

### Exercise Objectives

Preparation for PI Interface failover exercise.

### Problem Description

You need a second, identical PI Interface for the failover scenario. This interface connects to the OPC server remotely which requires a DCOM configuration.

### Approach

The PI Interface for OPC DA, like many applications, is a COM application. The PI OPC Interface communicates with the OPC Server (data source) through the COM layer. When the applications are running on different machines, they must use DCOM – in this instance COM over TCP/IP.

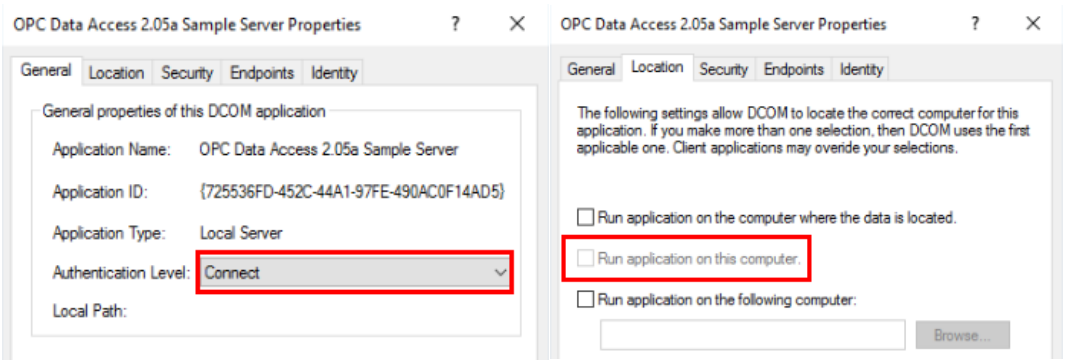
DCOM does not support gMSA in all configuration, but there is workaround again. One option is to change the gMSA account of the interface service for a standard user account and map this account to PI Interfaces identity on PI Data Archive. But we would like to keep the gMSA to authenticate to PI Data Archive.

There is a standard **PISCHOOL\PIInterface** (pw: piinterface) account prepared to use it as identity for OPC DA server instance which is necessary to configure in DCOM for remote connection.

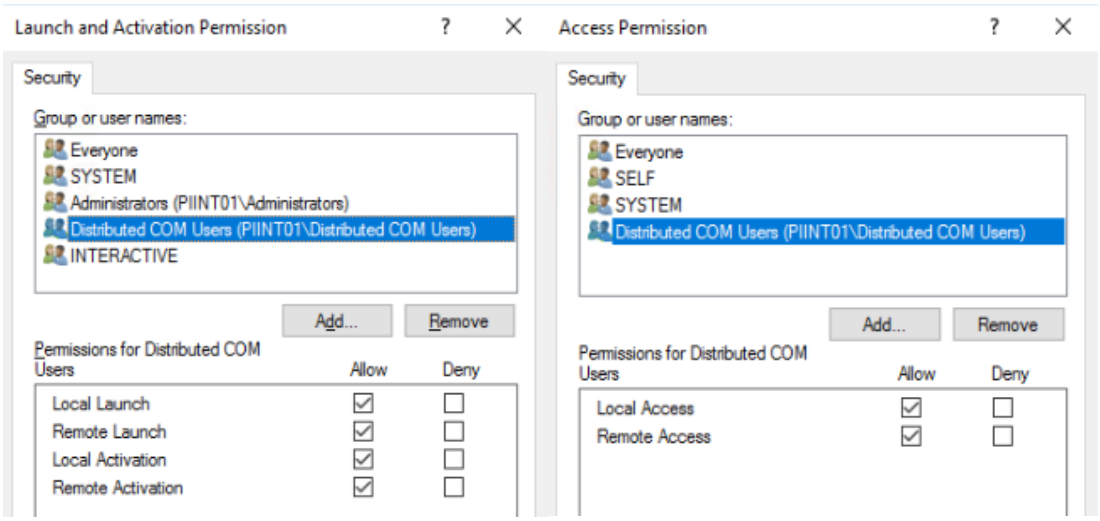
### 15.8.1 DCOM configuration on OPC server node and interface node

To configure the DCOM settings you must perform the following steps:

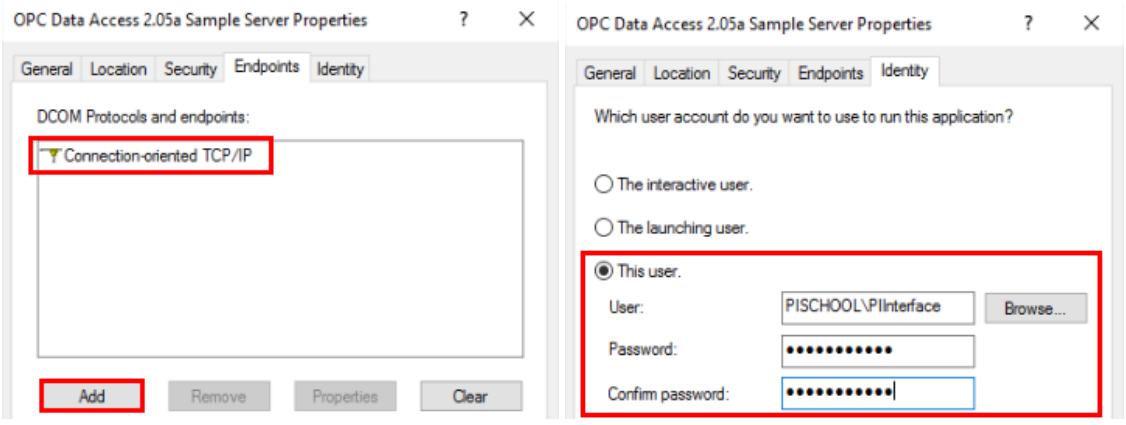
1. **Switch off** Windows Firewall completely on PIINT01 and PIINT02 servers.
2. On the node with OPC DA server (PIINT01), open Local Users and Groups (type **lusrmgr.msc** into Start Menu) and add **PISCHOOL\SVC-PIINT$** into *Distributed COM Users* group
3. Open Component Services console by typing **dcomcnfg.exe** into Start Menu.
4. Go to Computers → My Computer → DCOM Config and find *OPC Data Access 2.05a Sample Server*. Right-click and select **Properties**.
5. At **General** tab set Authentication Level to **Connect**.
6. At **Location** tab select **Run application on this computer** (this option may be greyed out, in this case leave it be)




7. At **Security** tab select Customize for Launch and Activation Permissions and Access Permissions. Click **Edit** and add the *Distributed COM Users* local group and allow all permissions.

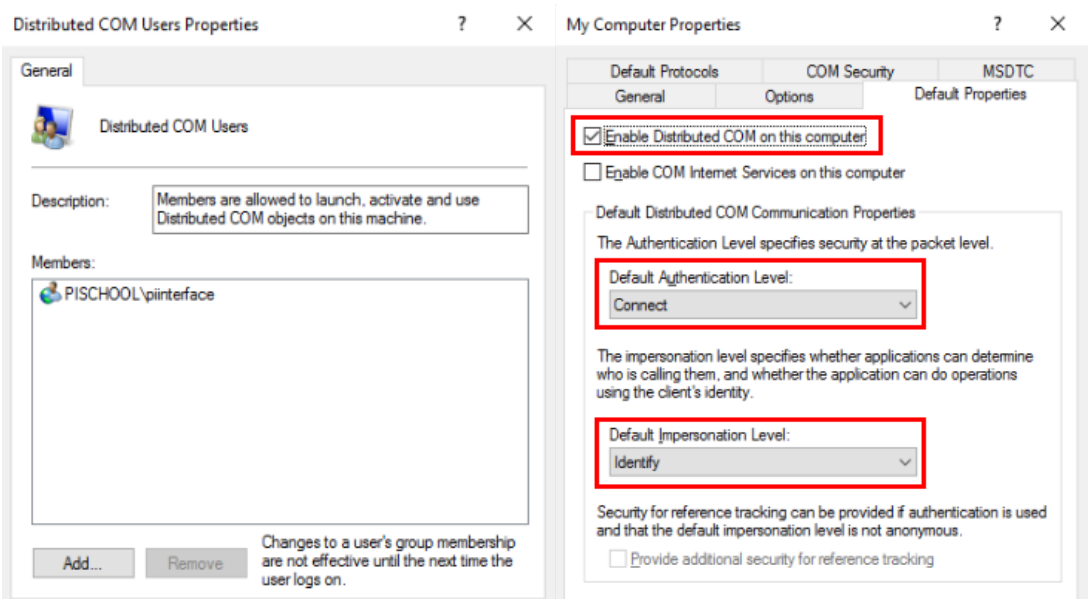



8. At **Endpoints** tab add **Connection-oriented TCP/IP** protocol with Use Default endpoints option.
9. At **Identity** tab select This user and enter **PISCHOOL\PIInterface** (pw: piinterface) account.




10. Apply the settings and go to the server with remote PI Interface for OPC DA (PIINT02).

11. Open Local Users and Groups (l**usrmgr.msc**)
12. Add **PISCHOOL\PIInterface** account into *Distributed COM Users* group
13. Type again **dcomcnfg.exe** into Start Menu. Go to Computers.
14. Right-click on My Computer and select **Properties**.
15. At **Default Properties** tab verify that **Enabled Distributed COM on this computer** is checked and Default Authentication Level is set to **Connect** and Default Impersonation Level is set to **Identify**.




See the "*DCOM Configuration Guide"* for more information on DCOM setup.

**Tip**

Many people feel that DCOM is inherently unsecure. Because of these security concerns, OSIsoft recommends installing the PI Interface for OPC DA on the same computer as the OPC Server. Of course, you cannot avoid this in an HA situation.

### 15.8.2 Testing the remote connection to OPC DA Server

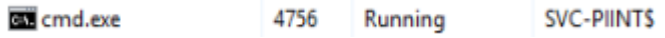
As you already know gMSA is denied interactive logon, meaning it is not possible to simply open application with “Run as…” command.

To open PI OPC Client under gMSA it is necessary to use **PsExec.exe** Windows SysInternal:

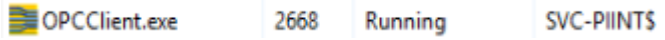
1. On PIINT02 open CMD from Star Menu.
2. Navigate to **C:\Class Documents\PsExec**
3. Type **PsExec.exe -i -u PISCHOOL\SVC-PIINT$ -p ~ cmd.exe**

```
C:\Class Documents\PsExec>PsExec.exe -i -u PISCHOOL\SVC-PIINT$ -p ~ cmd.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

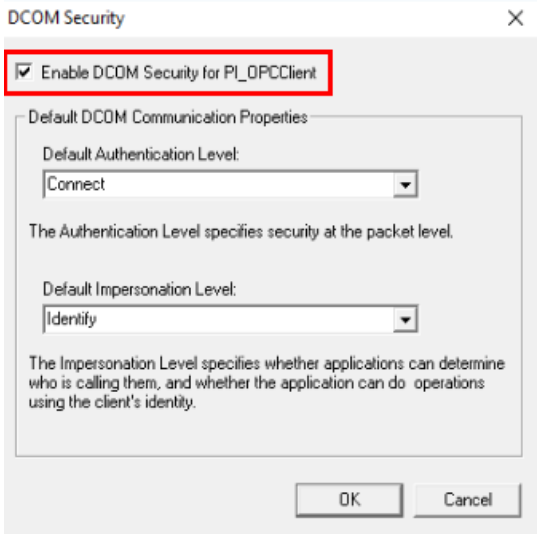
4. This opens a new CMD window that is now running under PISCHOOL\SVC-PIINT$. That you can verify in Task Manager:



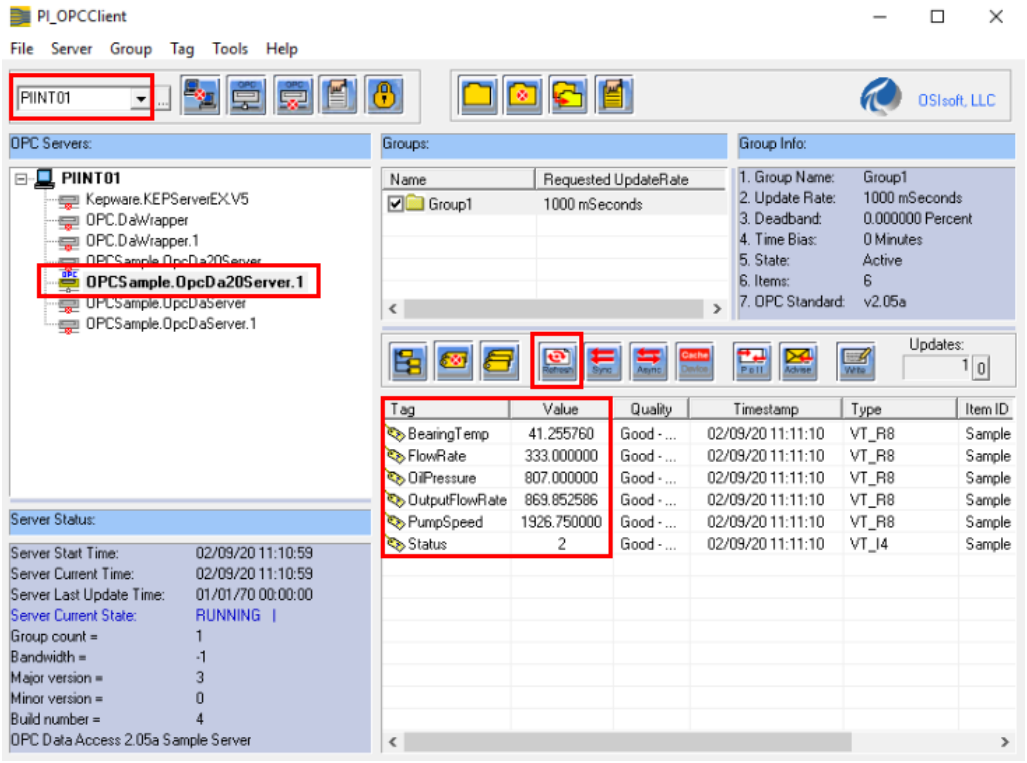

5. In this CMD navigate to **%PIHOME%\PI OPC Tools\PI_OPCClient** directory.
6. Run OPCClient.exe. This will start PI OPC Client under **PISCHOOL\SVC-PIINT$** gMSA. You can verify it in Task Manager:




7. Before connecting to the PIINT01 in PI OPC Client, go to **File → DCOM Security.**
8. Check **Enable DCOM Security for PI_OPCClient** and click **OK**. Keeping the Default Authentication Level to *Connect* and Default Impersonation Level to *Identify*.




9. Type PIINT01 instead of Localhost and repeat all the steps until you verify you can read values.

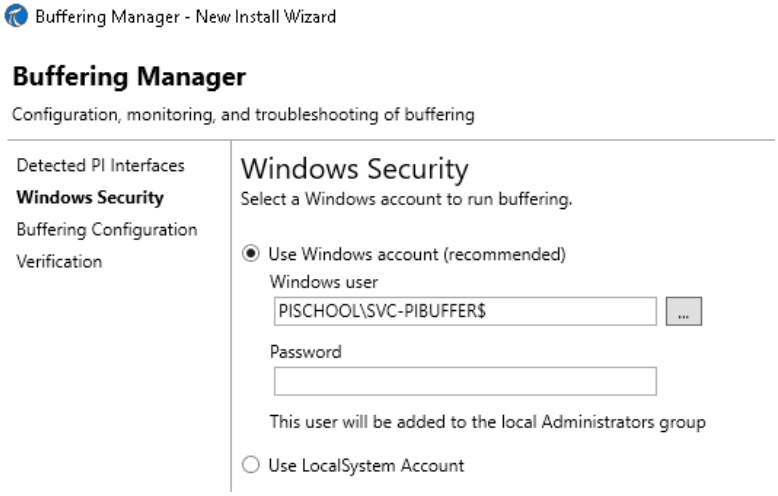



### 15.8.3 Buffering and PI Interface setup

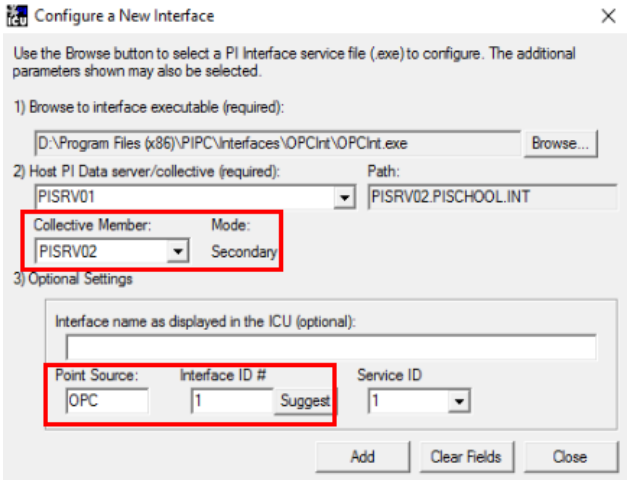
PI ICU, PI Interface for OPC DA and PI Buffer Subsystem are already installed on PIINT02.

**PISCHOOL\SVC-PIBUFFER$** has already been assigned to the service and it is a member of local *PI Buffering Administrators* group.

1. Finnish the buffering configuration. Open PI ICU, go to Tools → Buffering to open Buffering Manager. Continue with configuration.
2. The **PISCHOOL\SVC-PIBUFFER$** gMSA is pre-filled. Click **Next**.



3. Select the location for buffer queue file the same as on PIINT01 **D:\PI Buffer**. Then **Next**.
4. The buffering configuration is complete but as there is no PI Interface configured yet, there is no PI Data Archive server in the list in Buffering Manager. Go to **File** → **Add Data server…** and insert PISRV01 and confirm.
5. After authentication test our PI Collective is added to the list.
6. In PI ICU add new interface instance. Point it now to PISRV02. Keep the same Point Source and Interface ID and **Add**




7. Configure the interface the same as on PIINT01. This means same scan classes in General tab and settings in different sections. In OPC Server definition it is now **PIINT01::OPCSample.OpcDa20Server.1**.
8. Again, create the interface windows service under default *NT Service\OPCInt1* account and switch it for **PISCHOOL\SVC-PIINT$** gMSA in Services Manager. Restart PI ICU.
9. Stop the interface on PIINT01 and start the interface on PIINT02.
10. Verify the PI Points are updating. This means the DCOM connection to remote OPC DA Sever is working properly.

Keep only one interface running as failover has not been configured, yet. Otherwise there will be **DUPLICATED** values in PI Data Archive!

# 16. Failover Defined

## 16.1 PI Interface Failover

To minimize data loss during a single point of failure within a system, many PI Interfaces provide two failover schemes:

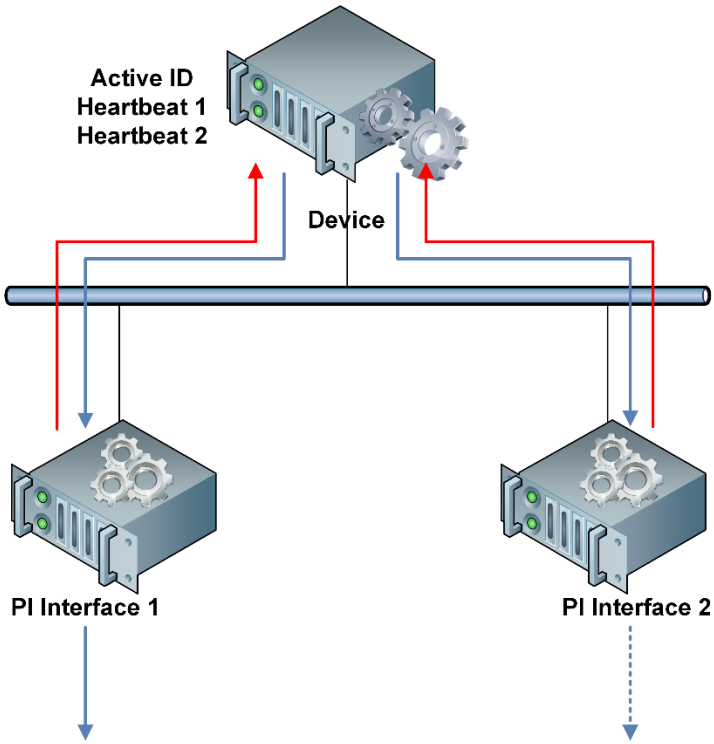
- **Phase One**: heartbeat synchronization through the data source
- **Phase Two**: heartbeat synchronization through a shared file.

Tip

Phase 1 is appropriate in two situations: (1) if performance degradation occurs using the shared file or (2) read/write permissions for the shared file cannot be granted to both interfaces.

### 16.1.1 Phase 1

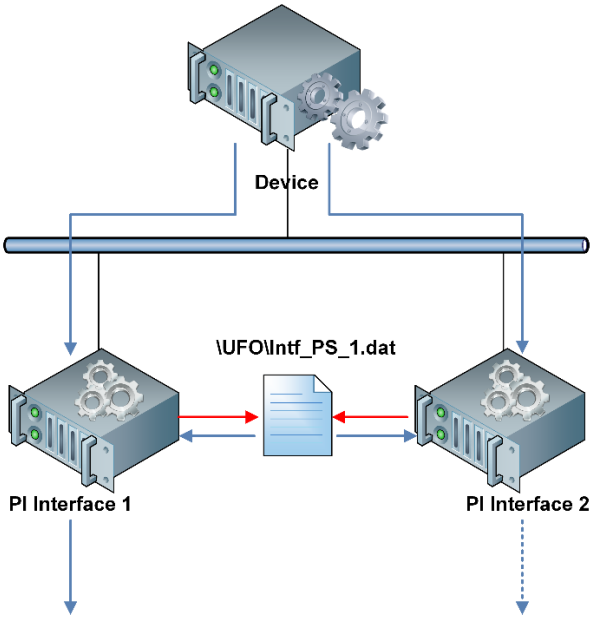
Phase 1 UniInt Failover uses the data source itself to synchronize failover operations and provides a hot failover, no data loss solution when a single point of failure occurs.



For this option, the data source must be able to communicate with and provide data to two interfaces simultaneously. Additionally, the failover configuration requires the interface to support outputs – data is *written* back to the data source. *This is something that many SCADA engineers view with apprehension, and as such is not recommended.*

### 16.1.2 Phase 2

Phase 2 UniInt Failover uses a shared file to synchronize failover operations and provides for *hot, warm,* or *cold* failover.




The Phase 2 hot failover configuration provides a no data loss solution for a single point of failure like Phase 1. However, in warm and cold failover configurations, *you can expect a small period of data loss during a single point of failure transition*.

## 16.2 PI Connector (1st Gen) Failover

Only a few first-generation PI Connectors currently support failover. The ones that do support failover do not need to communicate with the data source or a shared file like PI Interfaces do, but they communicate directly with each other. If the communication is successful, the connector communicates its own status to the other connector. If both connector instances have reliable data source communication, one of the connectors assumes the active role and other takes a backup role. The active connector sends data to both PI Data Archive and PI AF and the backup connector discards the collected data. PI Connectors (1st Gen) support currently **only HOT** failover.

Failover may occur in the following scenarios:

### Starting and stopping the connector

When one connector is stopped or shut down, it sends a stopped state to its peer, and the peer connector assumes the active role. When the stopped connector is started again, it takes the backup role. The connectors do not support automatic fail-back to the connector that was previously active. In other words, when the original primary comes back online, it will not initiate another failover scenario

### Data Source connection loss

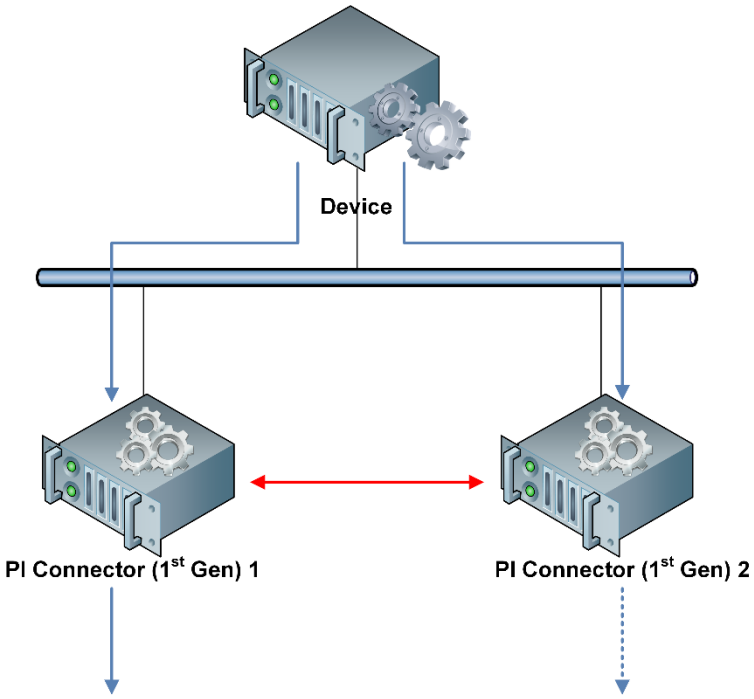
When an active connector loses its data source connection, it sends the information about its own data source status to the backup connector. Upon receiving the data source disconnection status from the active connector, the backup connector assumes the active role provided it has a good data source connection. If the backup connector does not have a reliable data source connection, both connectors continue in the same role until one of them resumes a reliable data source connection.

### Force transition to active role

You can transition the backup connector into an active role by clicking the *Made node active* button at the Failover section of PI Connector UI.

### Connection loss between connectors

When connector peers cannot communicate with each other, both become active connectors. In this scenario, duplicate data might be sent to PI System. When the connection resumes, both connectors negotiate their data source status, eventually one connector relinquishes the active role, and the other one becomes the backup.



## 16.3 Group Exercise - HA Step 4: Implement Failover

This activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

### Exercise Objectives

Implement PI Interface-level failover.

### Exercise Description

You have a working PI Interfaces on PIINT01 and PIINT02. You need to make them work together so when one fails the other takes over. For PI Interfaces configure **PHASE 2 HOT** Failover.

### Approach

### Configure the Shared File

Choose a location for the shared file. The file can reside on one of the interface nodes but OSIsoft strongly recommends that you put the file on a dedicated file server that has no other role in data collection. In this classroom, it will be simplest to put it on one of the interface computers.

1. Create a folder **D:\Failover** on PIINT01.

Setup a file share folder and assign the permissions so that both interfaces have read and write access to the file:

2. In folder properties **Security** tab add **PISCHOOL\SVC-PIINT$** gMSA and assign Full Control.
3. In **Sharing** tab click on Share. Make sure Student01 and SVC-PIINT$ are in the list and confirm. The UNC path now is **\\PIINT01\Failover**

### Configure the PI Interface Parameters

4. Start on PIINT01. Use the **Failover** section of the PI ICU to enable failover and assign two parameters for each interface:
    a. **Failover ID** number for the interface: **1** for \\PIINT01\OPCInt1
    b. **Failover ID** number for its backup interface: **2** for \\PIINT02\OPCInt1

**Failover ID** for each interface must be unique and each interface must know the **Failover ID** of its backup interface.

### Select the Synchronization File Path and File to Use for Failover.

5. Select the **HOT** type of failover.

Ensure that the username assigned in the **Log on as:** parameter in the **Service** section of the PI ICU is a user that has read and write access to the folder where the shared file will reside.

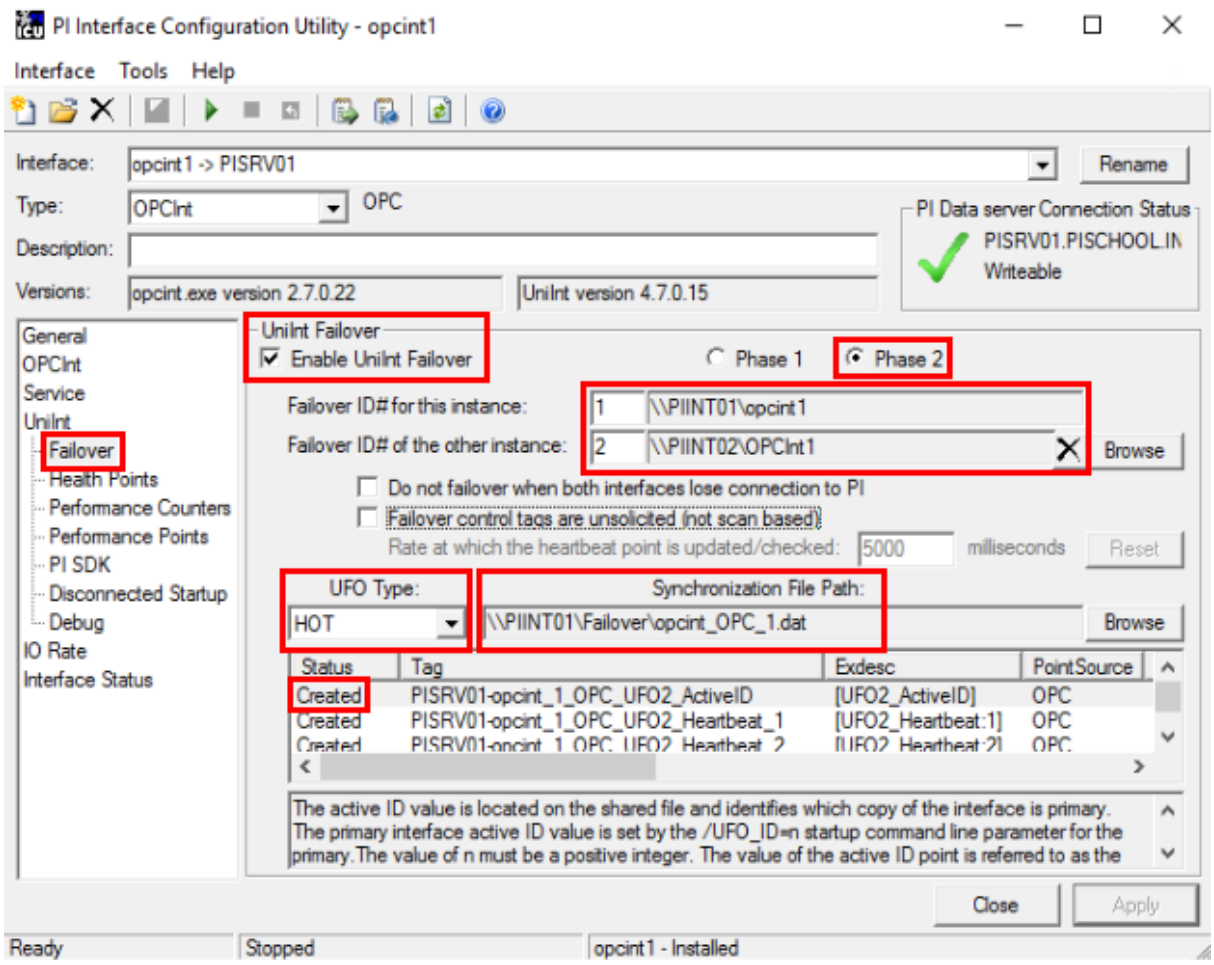
6. Enter the UNC \\PIINT01\Failover into the Synchronization File Path field

**Note:** The first interface to start will create the file.

All other command line parameters (beside the /HOST) for the primary and secondary interfaces must be identical.

### Configure the Digital State Set / PI points

7. Right-click on the Digital State point and select **Create UFO_State Digital State Set on Server …** to create the required digital state set.
8. Right-click on the points and select **Create all points (UFO Phase 2)** to create seven (7) PI points for the interface: **Active ID**, **Heartbeat_1**, **Heartbeat_2**, **DeviceStatus_1**, **DeviceStatus_2**, **State_1**, and **State_2**.



### Modify UFO Points Security

When the UFO PI Points are created, their security ACL in Point Security and Data Security attributes will copy the ACL of PIPOINT database where PI Interfaces and PI Buffer identities only have *Read* access. We need to provide the *Write* access to PI Buffer identity on Data Security. And remove unnecessary PI Identities from the ACL.

9. Use PI Builder MS Excel add-in. Search for points using tag mask: ***UFO***. Load only the security attributes.
10. Modify the Data Security ACL as it is below:

    piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r) | PI Buffer: A(r,w) | PI Vision: A(r)

11. Modify the Point Security ACL as it is below:

    piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r) | PI Buffer: A(r) | PI Vision: A(r)

12. Replicate to all UFO points and publish the changes in Edit Mode.

### Configure Failover on second PI Interface

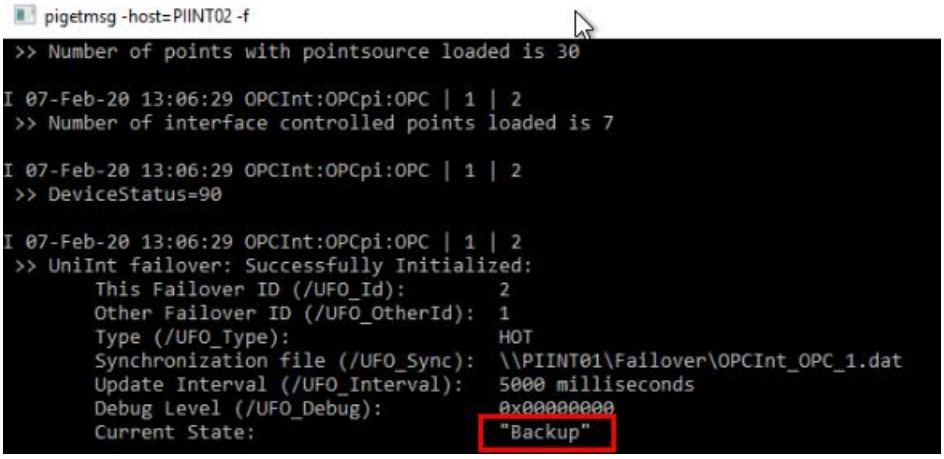
13. Repeat for the second interface. You will not have to recreate the digital state set or failover points. Of course, the Failover ID are switched for this interface and other instance:

| | | |
|---|---|---|
| Failover ID# for this instance: | 2 | \\PIINT02\OPCInt1 |
| Failover ID# of the other instance: | 1 | \\PIINT01\opcint1 |

Before you start testing, start Interactive message logging:

```
pigetmsg –f
```

In order to test that failover is working correctly you will have to have a properly running interface and then start the second. You should see that interface start and become the “Backup.”




Then you can stop the “primary” interface and you should see the change reflected in each log file

## 16.4 Group Exercise - HA Final Step: The Road Test

This group activity is designed to maximize learning in a specific topic area. Your instructor will have instructions and will coach you if you need assistance during the activity.

**Exercise Description**

Test the system to make sure all the failover and redundancy solutions are working.

**Approach**

Perform any or all the following actions:

1. Build a new display in PI Vision with objects displaying the point updated by PI Interface for OPC DA.

2. Using the PI SMT tool, add a random point to the collective by copying *CDT158* to *CDT69* on the primary server. Returning to PI Vision, failover as necessary until you are connected to the secondary server. Use **PI Search** to verify that *CDT69* exists on your secondary server.

3. Shut down the Primary PI Data Archive (using **%PISERVER%adm\pisrvstop.bat**)

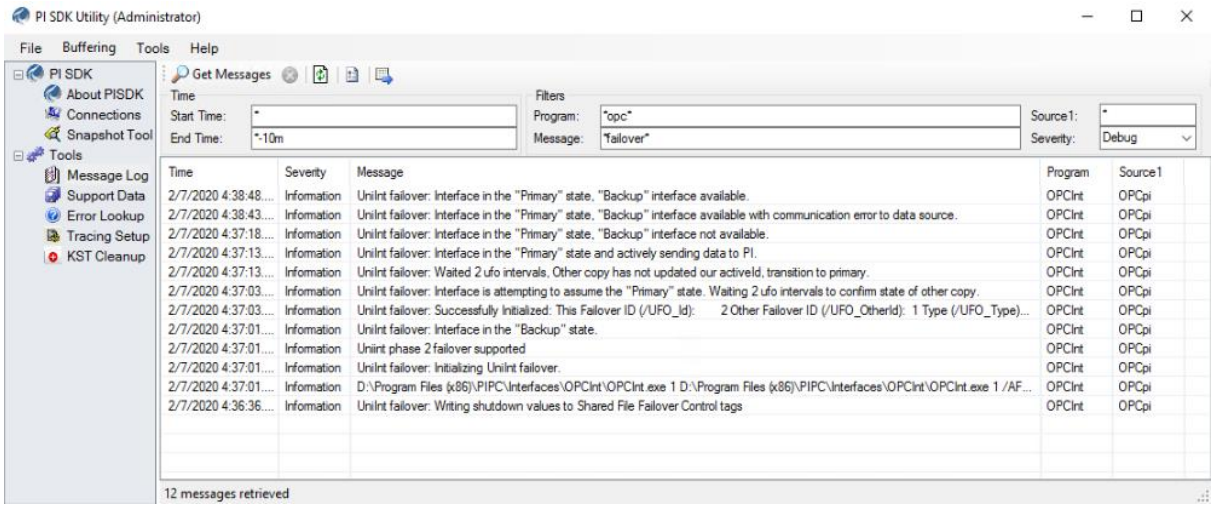
   - Is your PI Vision trend active? Did it switch connections?
   - Check %PIHOME%\bin\pibufss -qs on the primary data collection computer (the one that has that status in the active points) – is the data for the Primary Server queuing?
   - Try to add another Random point. Can you do it?
   - Check your **PI Collective Manager** on the secondary server – what does it show?
   - Start the Primary Data Archive.

4. Shut down the Primary PI Data Archive (using **%PISERVER%adm\pisrvstop.bat**)

   - Is your PI Vision trend active? Did it switch connections again? Did the Primary PI Data Archive lose data while it was down, or did the buffer work?
   - Check %PIHOME%\bin\pibufss -qs on the primary interface computer (the one that has that status in the active points) – is the data for the Secondary Server queuing?
   - Try to add another Random point. Can you do it?
   - Check the PI Collective Manager on the primary server – what does it show?
   - Restart the Secondary PI Data Archive
   - Use PI SMT – did the Secondary Server pickup your new Random point? Did the buffer work for the Secondary PI Data Archive such that you did not lose data?

5. Stop the primary interface (with the PI ICU or Services Manager)

- Did the other interface pick up data collection properly? Do you have updates in PI Vision?
- Do the Interface Status and Heartbeat points look appropriate?
- Check the PI Message logs on the interface computers using the PISDKUtility




- Check pibufss -qs on the secondary interface (now with the primary status). Is it processing data for both servers?
- Restart the primary interface and check for a good start.

6. Stop the “Backup” interface (with the PI ICU or Services Manager)
   - Did the other interface pick up data collection properly?
   - Do you have updates in PI Vision?
   - Do the Interface Status and Heartbeat points look appropriate?

Lastly, restart the Backup Interface. Validate that both PI Data Archives and both PI Interfaces are running correctly.

# 17. Final Exercise (Optional)

This is a solo activity designed to test your understanding of the material in this course. Ask for assistance if needed. Don't waste your time just wondering.

**Exercise Objectives**

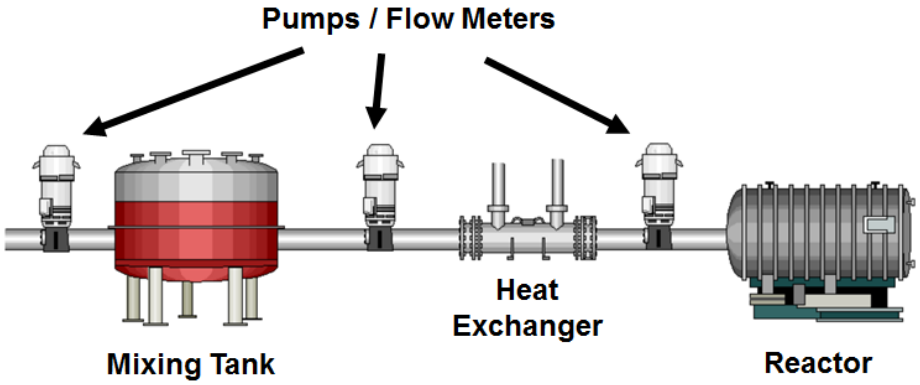
- Demonstrate your new skills
- Configure new instance of PI Interface for OPC DA for another OPC DA server
- Build and test PI points for the new interface instance
- Add new data source for PI Connector for OPC UA
- Build PI AF elements and create templates
- Connect data archive points to PI AF element attributes
- Show element data in PI Vision display

**Problem Description**

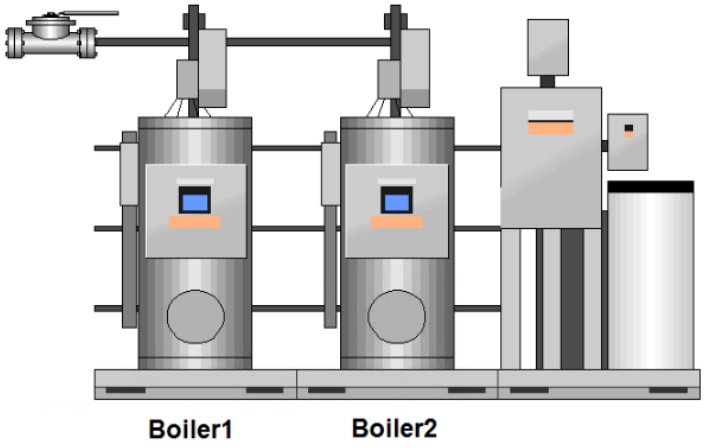
Your company had added a new production line in your plant and installed a new OPC DA Server on PIINT02 that collects data about the new line from DCS.

Your success in working with the PI System at your facility has put you right at the top of the list of people to implement the new system. Use the skills learned this week to implement this PI System, from interface to displaying the data to your users.

Fortunately, the facility is small. There are 2 units, each with a few elements as can be seen from the diagram below. Unfortunately, the process control people were less than helpful when naming the instrument tags and they may not match your PI AF and point nomenclature.



After your success, company added another line with 2 boilers. Boilers measurements are stored in the newly implemented OPC UA Server on PIINT01. With the skills you have just learned about PI Connectors, you should be able to add this new data source to the PI Connector configuration and let it build the PI Points and structure in PI AF for you.




### Approach

Alone or in collaboration with others:

- Explore the new OPC DA Server on PIINT02 using PI OPC Client
- Configure new instance of PI Interface for OPC DA on PIINT02
- Build and test the PI points for PI Interface for OPC DA
- Explore the new OPC UA Server on PIINT01 using UA Expert
- Add a new Data Source for PI Connector for OPC UA and let it create new PI Points and AF structure via PI Data Collection Manager
- Build PI AF elements for OPC DA points
- Build a simple PI Vision display

## 17.1 Preparation

Record the machine information in the following table:

| *Machine* | *IP Address* | *Hostname* |
|---|---|---|
| **PI Server** | | PISRV01 |
| **PI Interface for OPC DA** | | PIINT02 |
| **OPC DA Server** | | PIINT02 |
| **PI Connector for OPC UA** | | PIINT01 |
| **OPC UA Server** | | PIINT01 |

OPC DA Server on PIINT02 is already running. Server address for PI Interface for OPC DA configuration is **PIINT02::OPCSample.OpcDa20Server.1**

OPC UA Server on PIINT01 is running only interactively and it must be started manually. In the Start Menu click on **UaDemoServer** tile which starts the server and display endpoint URL: **opc.tcp://PIINT01:4841**


## 17.2 Configuration

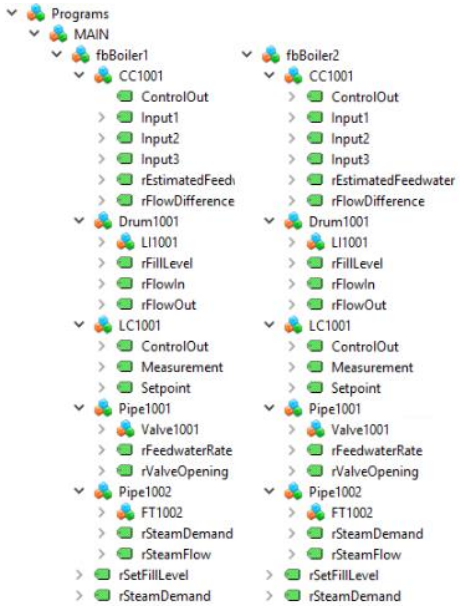
### 17.2.1 PI Interface for OPC DA

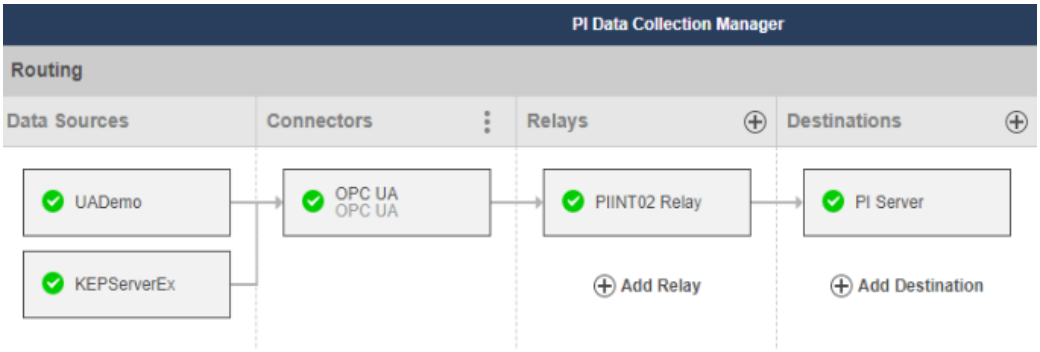
Create a new instance of PI Interface for OPC DA on PIINT02 to connect locally to OPC DA server to avoid DCOM.

### 17.2.2 PI Connector for OPC UA

Connect to the OPC UA server using UA Expert client tool. Use [None]:[None]:[Binary] security and browse the structure and drill down to reach the **fbBoiler1** and **fbBoiler2** items.

Write down the NodeID of MAIN: _____ to be used as Root Node in Data Source configuration in PI Data Collection Manager.

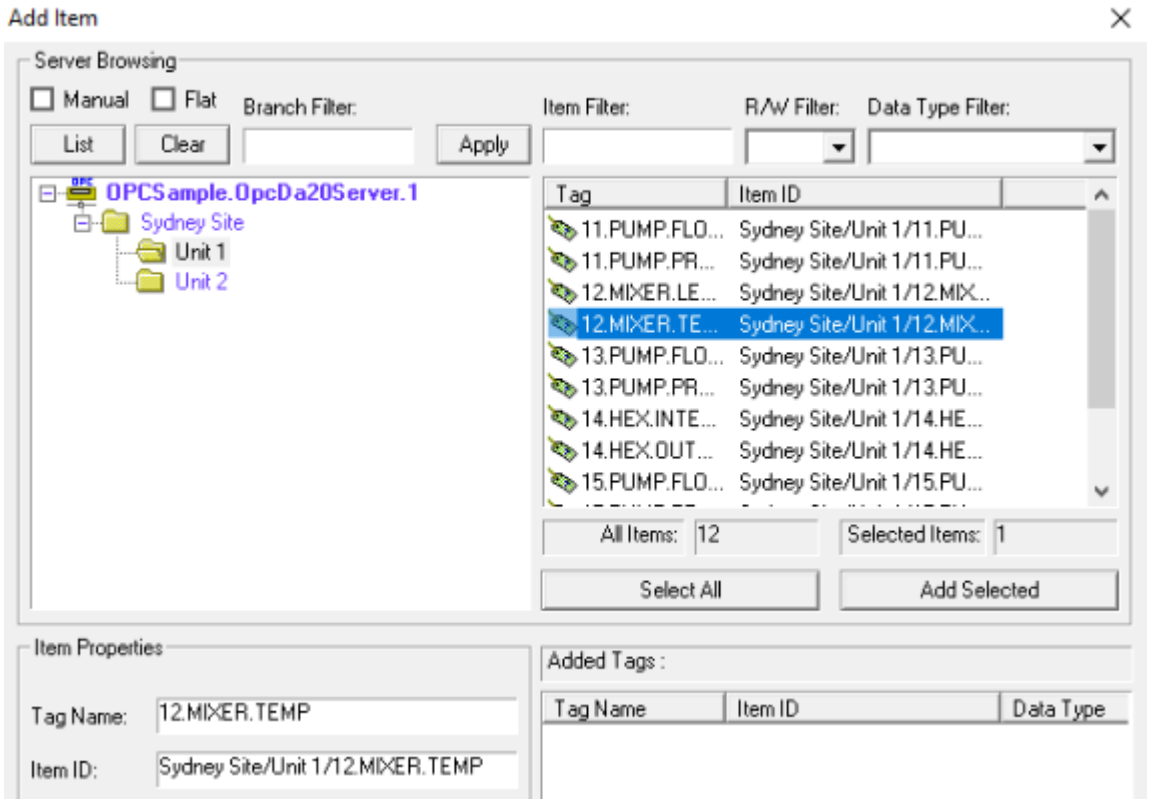




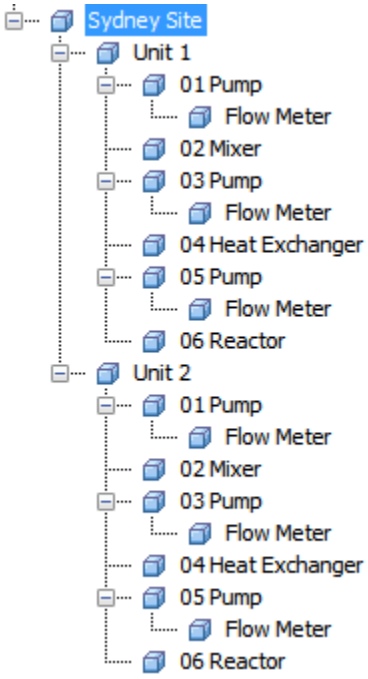

## 17.3 Build PI Points

PI Connector will create the points automatically, but for the PI Interface use the ***PI OPC Client*** to output a list of points on the OPC DA Server for use in PI Builder.

**DO NOT FORGET to define the Point and Data Security for the PI Points. Remember the PIPOINT database ACL setting!!!**



## 17.4 Build and Test AF Assets




- Using AF templates build the AF structure for the assets in your facility. It may look like the picture to the left.
- Construct an analysis rule that calculates the difference between Pump 1 flow and the Pump 5 flow such that the history is available. **Hint:** You will need to create a writable PI point that receives the output of the calculation.
- The flow units for the flows are l/min for the points and L/s in AF. Resolve this discrepancy, similarly for hPa and kPa on the pressure measurements.
- You may want to show all measurements in the AF using US customary units.
- Note the form of the Item ID below.

Answer the following questions:

- Do your asset attribute values show correctly in PI System Explorer?
- What is the authentication method?

## 17.5 Visualizing Data

Build a PI Vision display showing your equipment.

Answer the following questions:

- Can you switch between the assets seamlessly when a new display created?
- Do your element values show correctly in PI Vision?

Finished!

Plenty of drawing space below…

# 18. Appendix A: Complex Architectures

The recommended standard architecture to support PI implementation must be robust and highly available to ensure data availability to the end users that make decision based on PI System information.

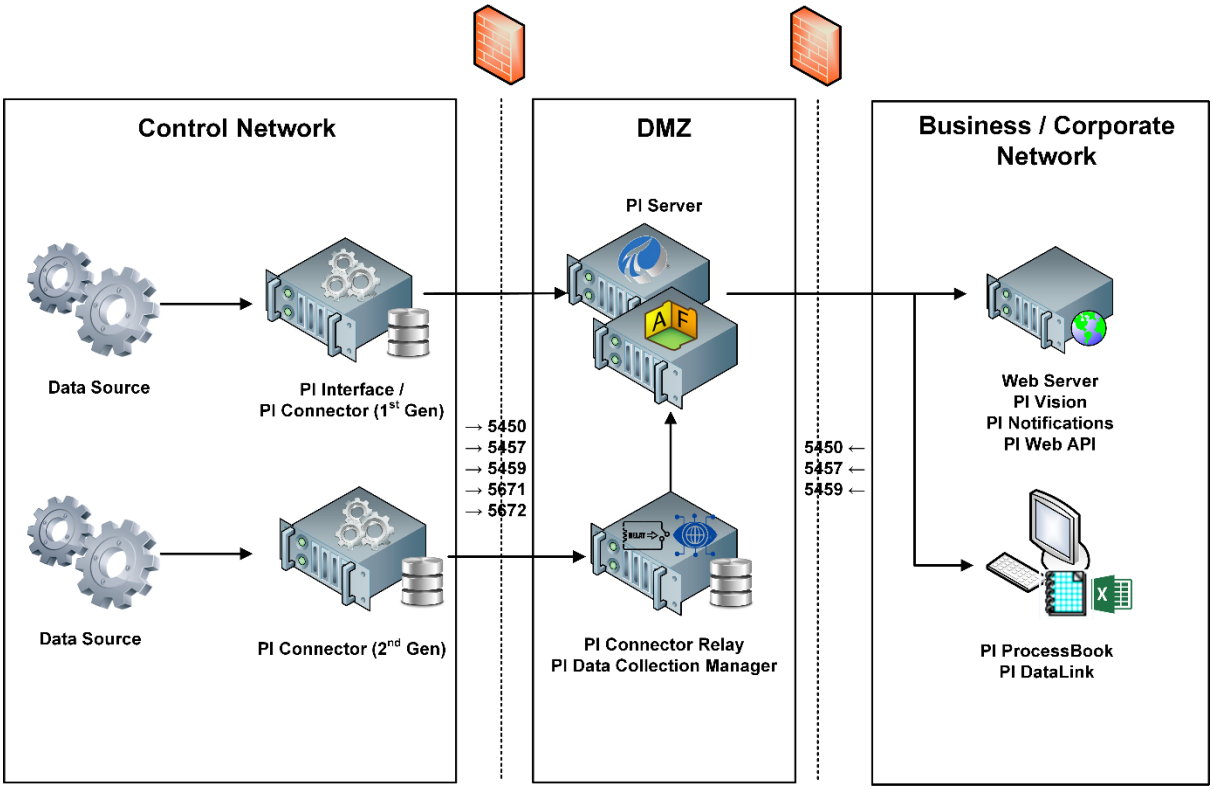
OSIsoft recommends utilizing at least one DMZ layer within your business network to host the PI Data Archive Collectives, PI AF servers, PI Analytics and SQL Server environment. This can be done by using virtual Local Area Networks (vLAN) or similar technologies.

OSIsoft recommends that PI Interfaces and PI Connectors are implemented as close to the data sources as possible, within the Process Control Network (PCN), if possible.

The most common PI System architecture patterns are listed below:

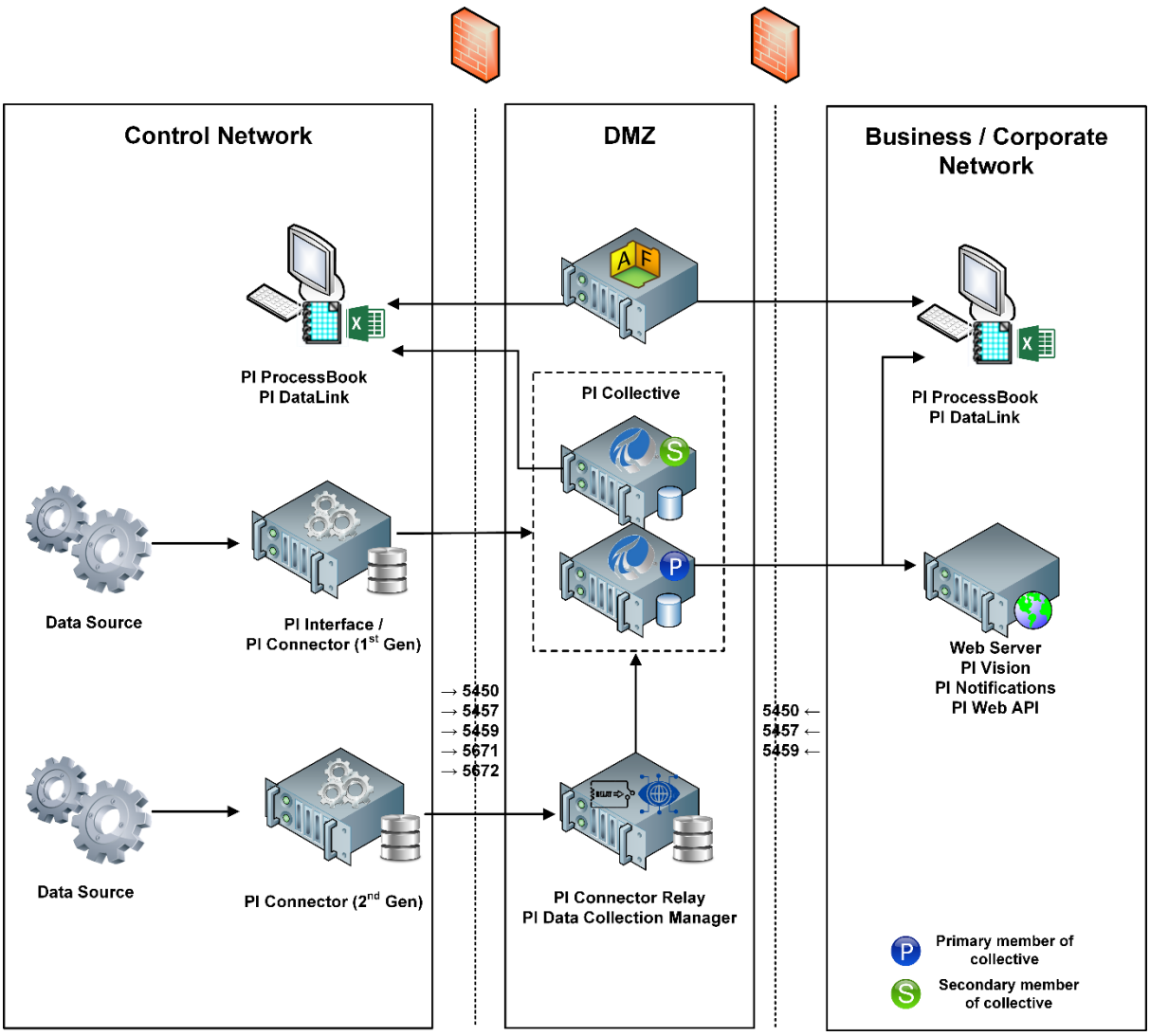
## 18.1 Pattern 1: PI Server in DMZ

This pattern segregates the control network and the corresponding PI Interfaces / PI Connectors from the less secure corporate network. This is accomplished by utilizing a DMZ to host the PI Server, including the PI Data Archive, AF Server, Analytics Server (Analysis and Notifications) and supporting systems. The DMZ should be configured with the appropriate inbound and outbound firewall rules to ensure that corporate network users cannot gain access past the DMZ layer into the control network.



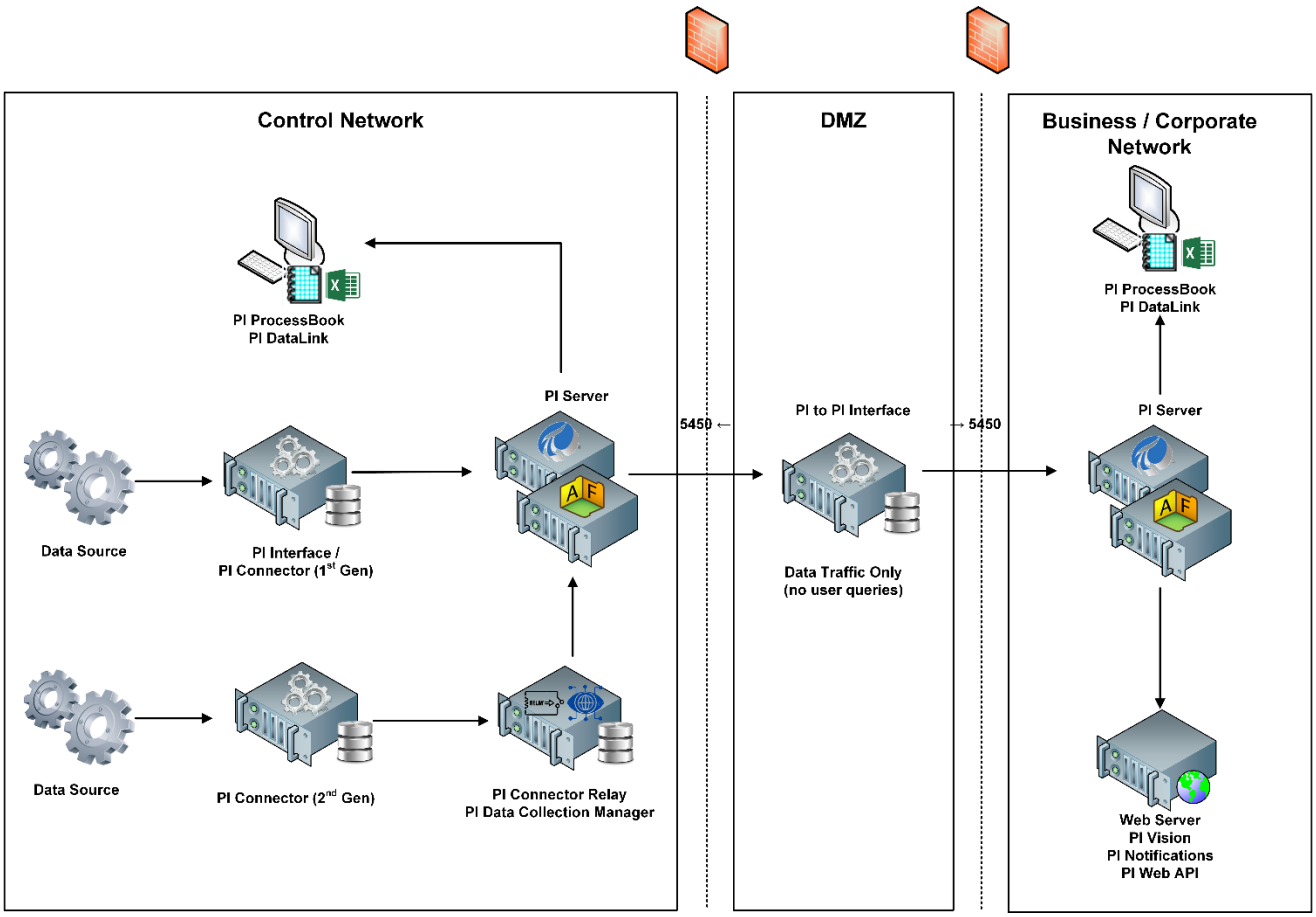
## 18.2 Pattern 2: PI Server High Availability in DMZ

This pattern resembles Pattern 1, using a DMZ layer to host the PI Server and supporting systems. However, this pattern includes the use of High Availability for the PI Data Archive. The PI Clients should be configured to preferably connect to a specific collective member thereby preventing any protected network users in the control network from connecting to the same PI Data Archive as users in the less secure corporate network. PI AF server is shared.



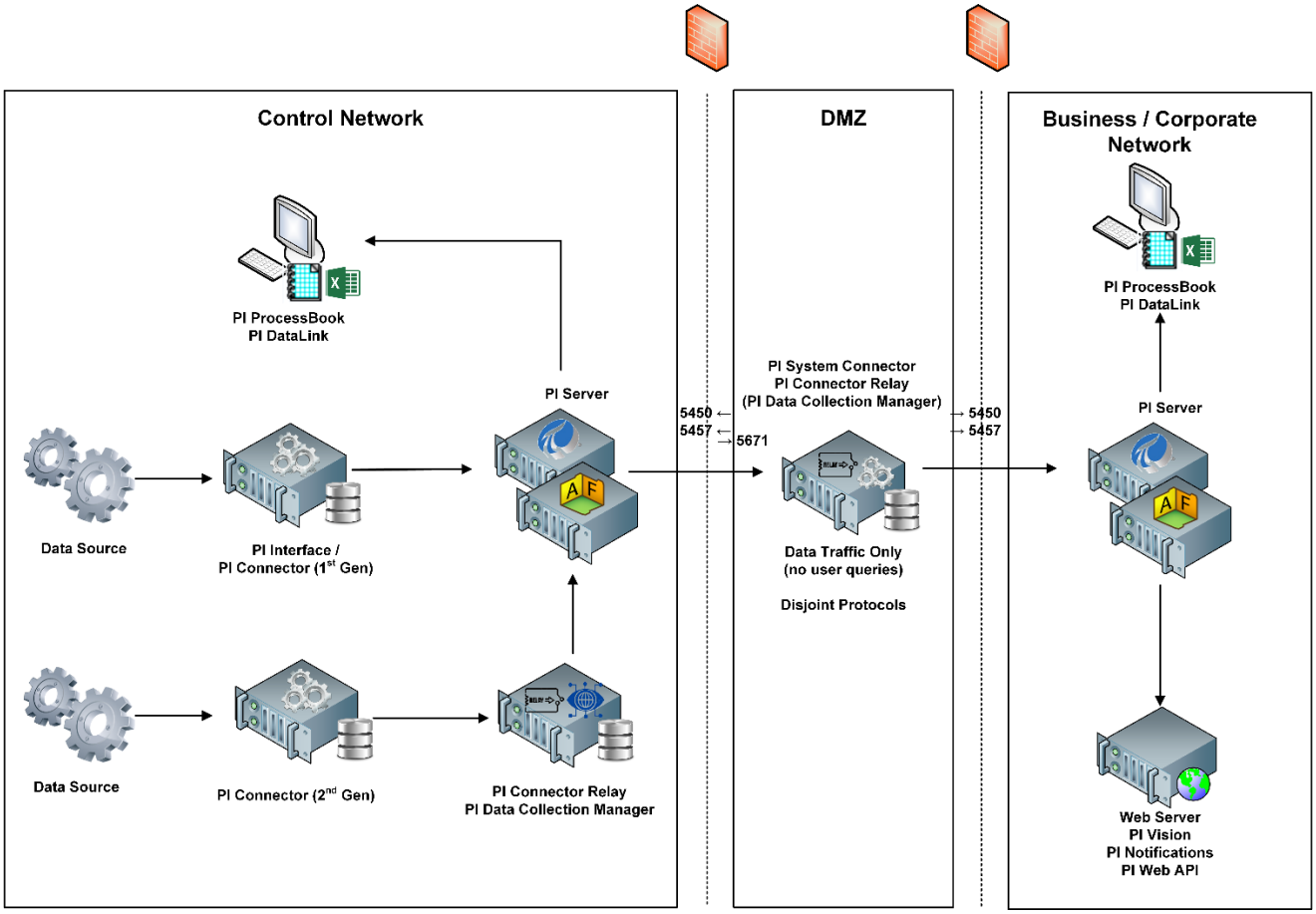
## 18.3 Pattern 3: PI to PI Interface in DMZ

This pattern further secures access to the control network from the DMZ by introducing a separate PI Server within the control network. The end users in this secure network layer utilize this PI Server for their purposes within the Process Control Network. The data can then be replicated from the control network to the corporate network with the use of a PI-to-PI interface located in the DMZ layer. This minimizes the number of connections into and out of the DMZ from both the control and corporate layers. With PI-to-PI interface only PI Data Archive values are replicated. PI AF servers' hierarchies are independent.



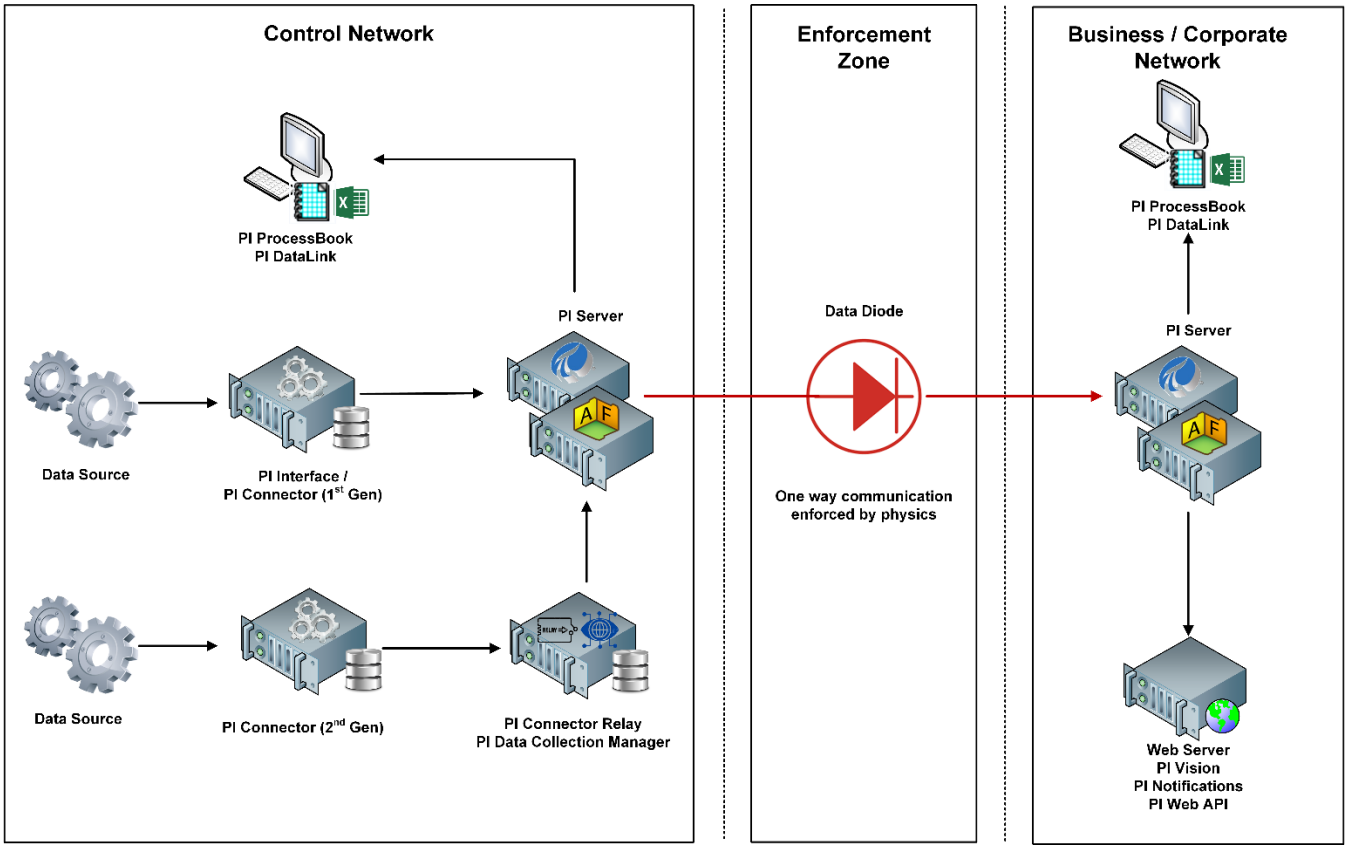
## 18.4 Pattern 4: PI System Connector in DMZ

This pattern resembles Pattern 3, the difference is that the PI Connector Relay technology replaces the PI-to-PI Interface for an even more secured method for sending data from the PI System in the Control network to the PI System in the Corporate Network. With PI System Connector not only replicates values from PI Data Archive, but also PI AF server hierarchies and event frames. If there is no PI Data Collection Manager utilized in Control Network, then PI Data Collection Manager should be installed inside DMZ on the same machine as PI System Connector and PI Connector Relay or on a separate server.



## 18.5 Pattern 5: Absolute Enforcement

This pattern further secures access to the control network from the DMZ by utilizing a Data Diode or equivalent device that enforces one-way communication between the Control Network's PI System and the Corporate Network's PI System.



# 19. Appendix B – Ports

You will have to open certain network ports for the applications to communicate.

The following ports may need to be opened on a firewall to allow access to the PI System or other associated services. Taken from Firewall Port Requirements knowledge article on Customer Portal

| Port | Type | Application | Comment |
|---|---|---|---|
| 25 | TCP | SMTP Server | Email Delivery Channel for PI Notifications |
| 53 | UDP | DNS | IP/Host Lookup (PI Trust) |
| 80 | TCP | HTTP | Connection to applications via HTTP in web browser |
| 88 | TCP/UDP | KDC | Kerberos (PI Mappings) |
| 135 | TCP/UDP | Domain Controller | SPN Registration (PI Mappings) |
| 389 | TCP/UDP | LDAP | Cross-domain authentication |
| 443 | TCP | HTTPS | Connection to applications via HTTPS in web browser |
| 445 | TCP | SMB | Search for local accounts to manage mappings remotely through PI AF Client |
| 1433 | TCP | SQL Server | Hosting the PI databases |
| 1434 | UDP | SQL Server Browser | Remotely identify SQL instances |
| 5450 | TCP | PI Data Archive | Client connections to PI Data Archive |
| 5456 | TCP | PI ACE | PI ACE calculations exposed via a Web Service |
| 5457 | TCP | PI AF Server | Client connections to PI AF server |
| 5458 | TCP | PI Notifications 1.x | Client connections to PI Notifications Scheduler |
| 5459 | TCP | PI AF Server | PI SQL for AF Client connections to PI AF server (e.g. PI OLEDB Enterprise) |
| 5460 | TCP | PI SQL DAS | HTTP explicit login endpoint for PI SQL DAS 1.3.3 and earlier |

| | | | |
|---|---|---|---|
| 5461 | TCP | PI SQL DAS | HTTPS endpoint |
| 5462 | TCP | PI SQL DAS | NET.TCP endpoint |
| 5463 | TCP | PI Analysis Service | Client connections to PI Analysis Service |
| 5464 | TCP | PI SQL DAS (RTQP) | HTTPS endpoint |
| 5465 | TCP | PI SQL DAS (RTQP) | NET.TCP endpoint |
| 5468 | TCP | PI Notifications 2.x | Client connections to PI Notifications Service |
| 5671 | TCP | PI Integrator for BA | PI Integrator for Business Analytics outgoing data (required only for Microsoft Azure IoT Hub or Microsoft Azure Event Hub) |
| 5671 | TCP | PI Connector Relay | AMQPS using X.509 certificate. Data is sent from the connector to the relay on this port. All communication is encrypted. |
| 5672 | TCP | PI Data Collection Manager | AMQPS using X.509 certificate. Connectors and Relays use this port for communication to the PI Data Collection Manager. This port must remain open. |

# 20. Appendix C – Abbreviations

| | |
|---|---|
| ACE | Advanced Computing Engine |
| ACL | Access Control List |
| AF | Asset Framework |
| AMQPS | Advanced Message Queuing Protocol Secure |
| API | Application Programming Interface |
| BA | Business Analytics |
| CMD | Command Prompt |
| DA | Data Archive |
| DAS | Data Access Server |
| DB | Database |
| DCM | Data Collection Manager |
| DCOM | Distributed Component Object Model |
| DCS | Distributed Control System |
| DHCP | Dynamic Host Configuration Protocol |
| DMZ | Demilitarized Zone |
| DNS | Domain Name System |
| EMS | Enterprise Management System |
| ERP | Enterprise Resource Planning |
| FQDN | Fully Qualified Domain Name |
| gMSA | Group Managed Service Account |
| HA | High Availability |
| HTTPS | Hyper Text Transfer Protocol Secure |
| ICU | Interface Configuration Utility |
| IIS | Internet Information Services |
| IP | Internet Protocol |
| JDBC | Java Database Connectivity |
| LAN | Local Area Network |
| LDAP | Lightweight Directory Access Protocol |
| LIMS | Laboratory Information Management System |
| MDB | Module Database |

| | |
|---|---|
| MS | Microsoft |
| MSA | Managed Service Account |
| ODBC | Open Database Connectivity |
| OLE | Object Linking and Embedding |
| OLEDB | Object Linking and Embedding Database |
| OPC A&E | OPC Alarms & Events |
| OPC DA | OPC Data Access |
| OPC HDA | OPC Historical Data Access |
| OPC UA | OPC Unified Architecture |
| OPC | Open Platform Communication (or OLE for Process Control) |
| OS | Operating System |
| OSI | Oil Systems Incorporated (former name of OSIsoft) |
| PCN | Process Control Network |
| PI | Plant Information |
| PIFD | PI Foundation Database |
| PLC | Programmable Logic Controller |
| RDBMS | Relational Database Management System |
| RTQP | Real Time Query Processor |
| SCADA | Supervisory Control and Data Acquisition |
| SDK | Software Development Kit |
| SMB | Server Message Block |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| SPN | Service Principal Name |
| SQL | Structured Query Language |
| SSL | Secure Socket Layer |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| UDP | User Datagram Protocol |
| UFL | Universal File Loader |
| UFO | UniInt Failover (Universal Interface Failover), not the  |
| UOM | Unit of Measurement |