PI System Administration: Troubleshooting

OSIsoft, LLC
1600 Alvarado Street
San Leandro, CA 94577 USA
Tel: (01) 510-297-5800
Web: http://www.osisoft.com

Published: October 4, 2021

Table of Contents

Contents

1. Introduction

1.1 Lab Overview

Are you new to the PI System and find yourself in an administrator role? Are you interested in beefing up your troubleshooting skills and learning some tips for administering the PI System? This lab will walk you through troubleshooting some common misconfigurations based on the real-world experiences of OSIsoft Product Support Engineers. You'll also learn time saving tips and other ways to make your life as a PI System Administrator easier. This lab is meant for new administrators but could provide a valuable refresher for seasoned administrators.

The first exercises in this lab will walk you through common user complaints all the way to resolution. Most complaints you will get from users are that data is missing, stale, or displaying bad values.

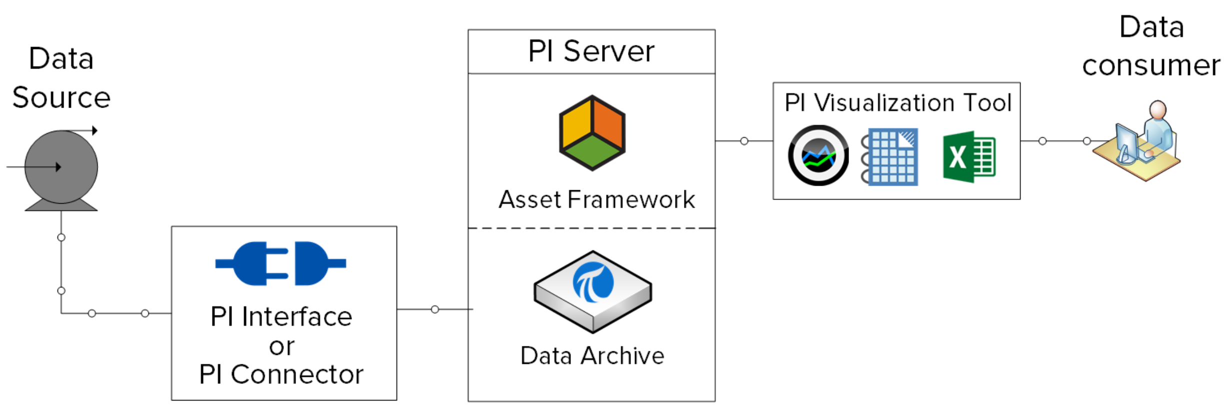
In the last exercise, we will configure some basic Notifications for Stale and Bad data.

In the process we will:

- Access PI Interface logs to find clues as to why tags are not working
- Modify tag configuration to get tags working
- Configure buffering to reduce gaps in data due to network interruptions
- Dispel a common misconception about PI Collectives
- Set up alerts for stale and bad data in order to identify problems before users complain

1.2 PI System Introduction

The PI System is a software suite that collects, stores and enhances data from your plant or process, and delivers it to users who need it. Simply put, the PI System is everything between the data source and the data consumer. The simplest possible PI System is made up of the following software components:

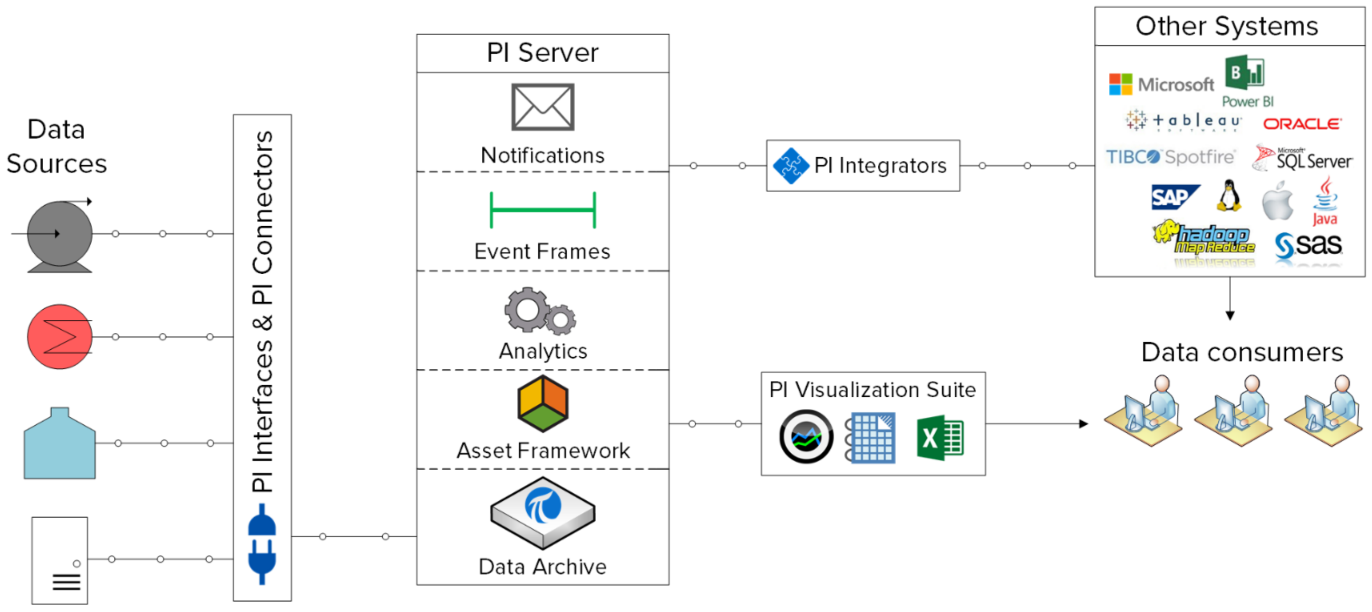


- **PI Interface or PI Connector**: Collects data from a data source

- **PI Server**
  - **Data Archive**: Stores the data
  - **Asset Framework**: Organizes and enhances the data

- **PI Visualization Tool**: Displays the data to the consumer

A more complete PI System would look like this:



OSIsoft developed more than 400 PI Interfaces to connect to different control system data sources, collect real time data and send it to the PI Data Archive to be stored. Some of the most popular interfaces include but are not limited to:

PI Interface for OPC DA:
This is the interface that will be used in this lab to collect real time data from an OPC Simulator (Simulates an OPC DA Server. In this class, the MatrikonOPC Server for Simulation is used.) and send it to the PI Data Archive.

PI Interface for Universal File and Stream Loading (UFL):
This interface reads data from ASCII data sources and writes data to PI Data Archive.

PI Interface for Relational Database (RDBMS via ODBC):
This interface enables you to transfer data between a PI System and any relational database management system (RDBMS) that supports Open Database Connectivity (ODBC) drivers.

PI Interface for Modbus Ethernet PLC:
This interface reads data from Modbus devices, such as PLCs, and writes data to PI Data Archive.

1.3 Understanding PI Points

1.3.1 PI Points

Commonly referred to as PI Tags represent a single named stream of data coming from an instrument, device or sensor. Historical or future data stored sent to the PI Data Archive are stored in PI Points.

In a process, this could be:

- The temperature in a tank
- The volumetric flow through a pump
- The speed of a propeller

1.3.2 Defining key PI Point Attributes

PI Point *attributes* are what define the PI Point. They have multiple different functions, including:

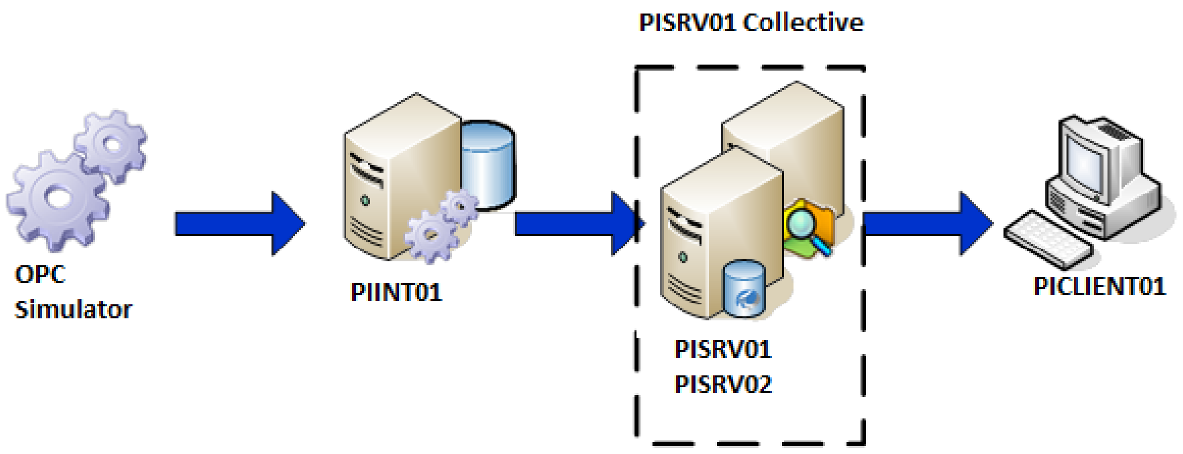
- Specifying how to collect data at the data source
- Defining which PI Interface is responsible for collecting the data
- Describing the data stream so users can search for it

There can be more than 50 different attributes defining a PI Point. Here are a few key attributes:

- **Name:** The name of the PI Point, which must be unique within the Data Archive.
- **Description**: A free text field attached to a PI Point, often used to enter a human friendly description of the PI Point. For example, a temperature point might be TC365674A.pv and the descriptor could be 'Reactor 65 Operating Temp'. Note that PI Points are not required to have a Description.
- **Point Type:** This attribute defines the type of data that is stored in the Data Archive.
- **Point Source**: This attribute commonly specifies which PI Interface is collecting the data for the PI Point.

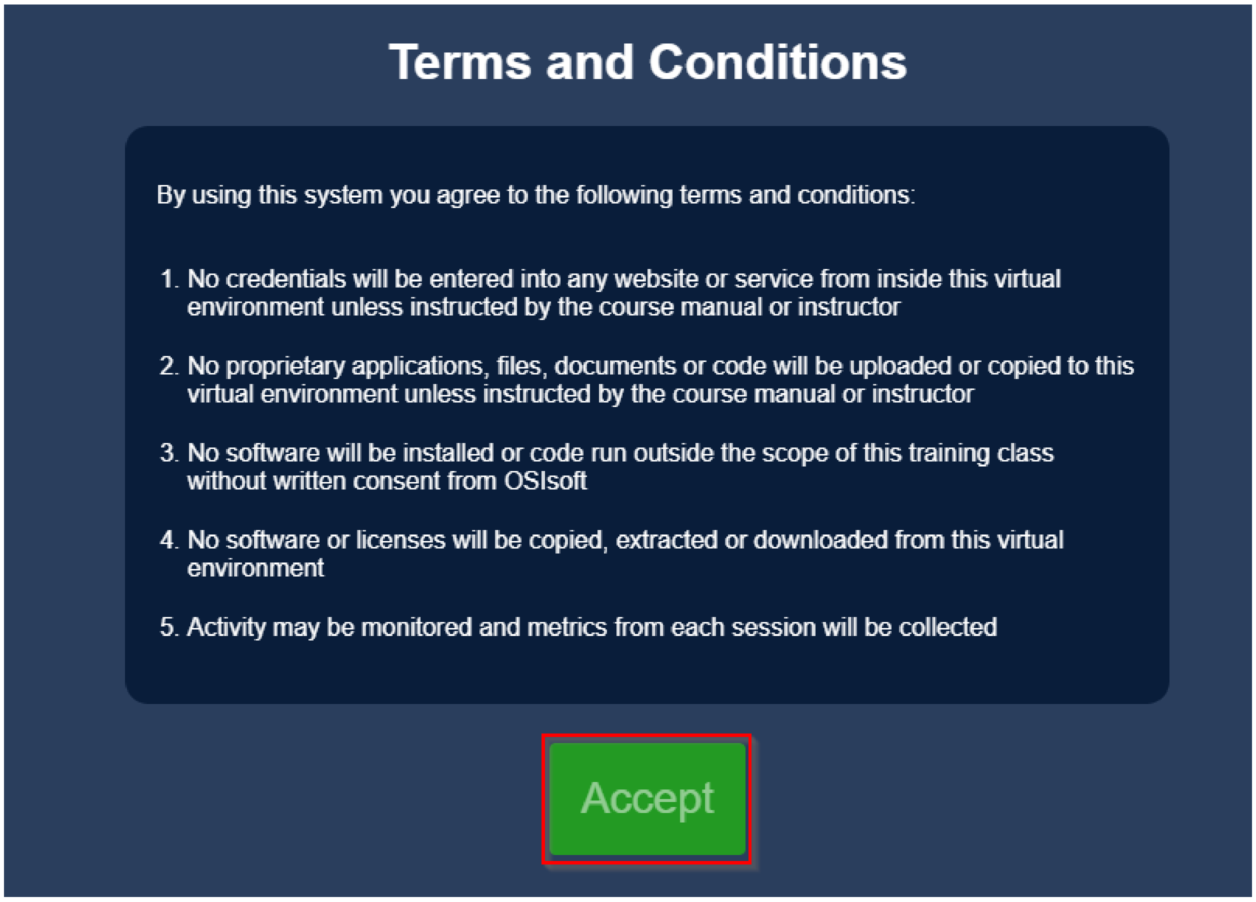
1.4 Lab Environment

In this lab you'll be working with a complete PI System. The diagram shows the flow of data from the OPC Simulator to the PI Data Archives, which form the PISRV01 PI Collective. Data is stored in PI Tags which are consumed by various client applications:



1.5 Connecting to the Virtual Machines

An excel spreadsheet has been sent to your email with the VM connection information. Open a browser of your choice > copy the URL links that are associated with your name. You will be taken to a Terms and Conditions page. Once you click on the "Accept" button, you will be automatically logged into the VM. No usernames or passwords needed!



Terms and Conditions

By using this system you agree to the following terms and conditions:

1. No credentials will be entered into any website or service from inside this virtual environment unless instructed by the course manual or instructor
2. No proprietary applications, files, documents or code will be uploaded or copied to this virtual environment unless instructed by the course manual or instructor
3. No software will be installed or code run outside the scope of this training class without written consent from OSIsoft
4. No software or licenses will be copied, extracted or downloaded from this virtual environment
5. Activity may be monitored and metrics from each session will be collected

2. Directed Activity – Getting Started

2.1 Objective

In this activity we will get acquainted with the environment and begin to respond to a user complaint.

One of your users is complaining about a pair of tags that are showing bad values. The tags in question are named RetiredTag and BrokenTag. The user complains that they were displaying valid data last week, but BrokenTag and RetiredTag are now displaying values of "Scan Off" and "Configure" respectively.

It is of course expected that you will fix the problem, but most of this work will take place in the next exercise. Before diving in, let's get acquainted with a couple basic administration tools.

2.2 Tasks

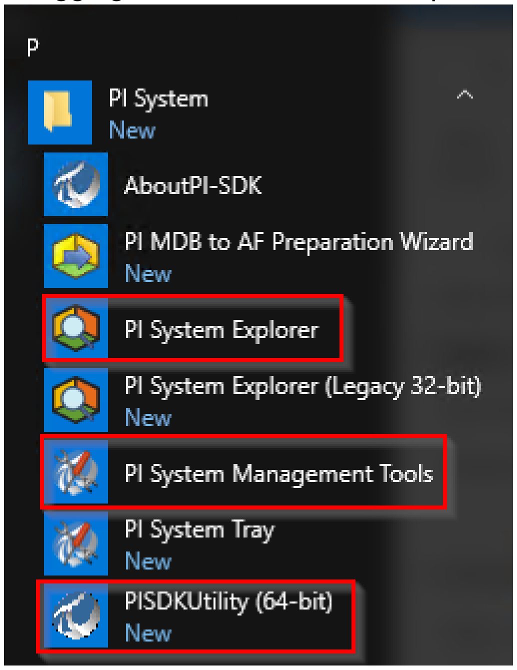
- Use the PI System Management Tools (PI SMT) Current Value plugin

2.3 Step-by-Step Instructions

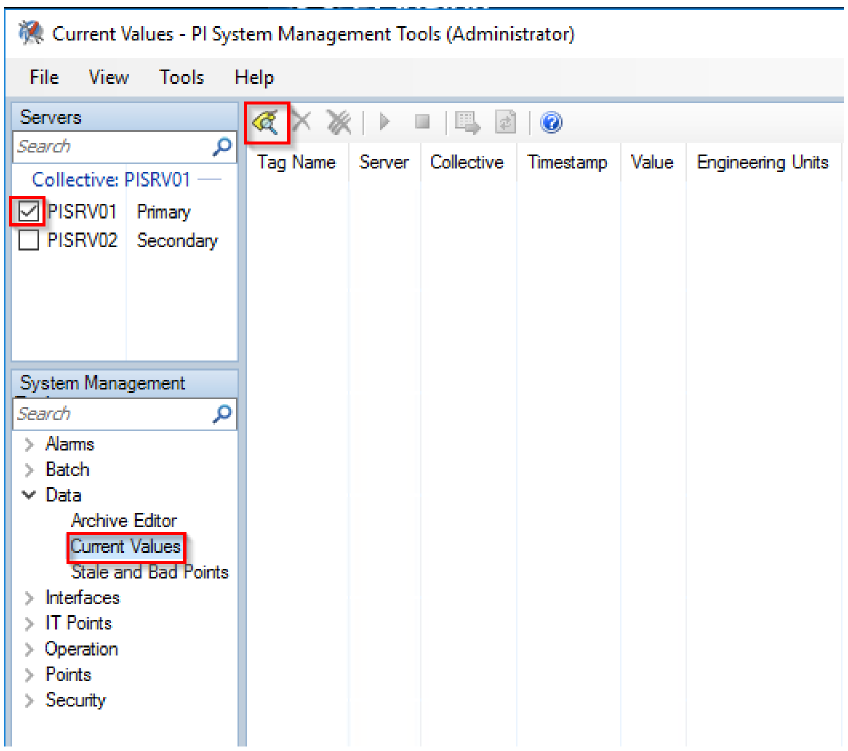
a. Rule out User Error by checking tag values

The first thing to do is rule out user error. From **PICLIENT01**, let's check the current value of the tags in question. There are of course many ways to do this, but a quick way without doing a screen share, leaving your desk, or logging into a server is to fire up PI System Management Tools and use the Current Value Plugin.

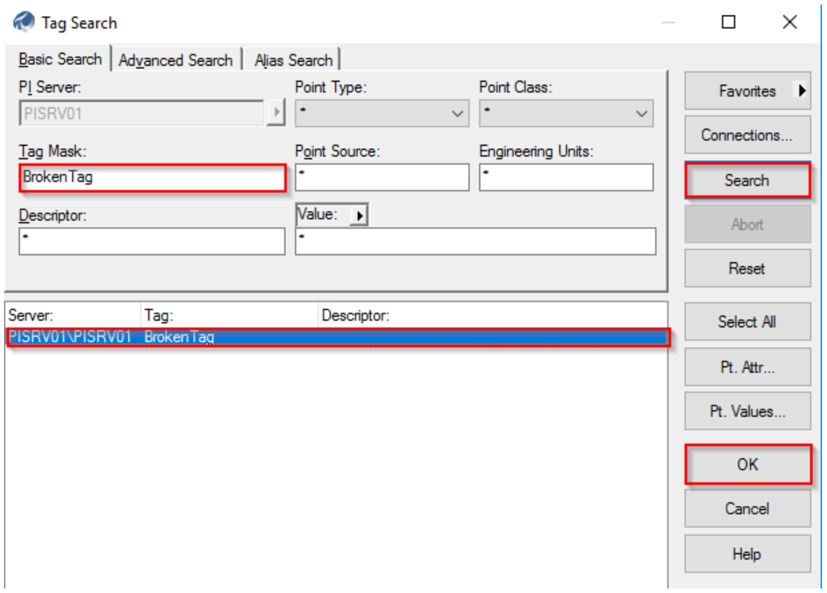
Note: PI System applications used in this lab have all been pinned to the machines they are meant to be used on. The screenshot on the right shows the names and symbols of the applications we will be using


P
PI System
New
AboutPI-SDK
PI MDB to AF Preparation Wizard
New
PI System Explorer
PI System Explorer (Legacy 32-bit)
New
PI System Management Tools
PI System Tray
New
PISDKUtility (64-bit)
New

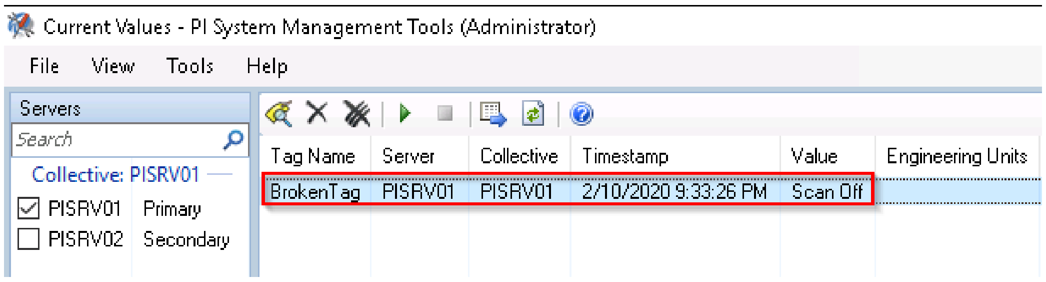
Check the box next to **PISRV1** to connect to the PI Data Archive. Expand Data -> Current Values, then click the icon to search for Tags.



In the search dialog, enter BrokenTag, click search, then select BrokenTag in the result list and click OK.

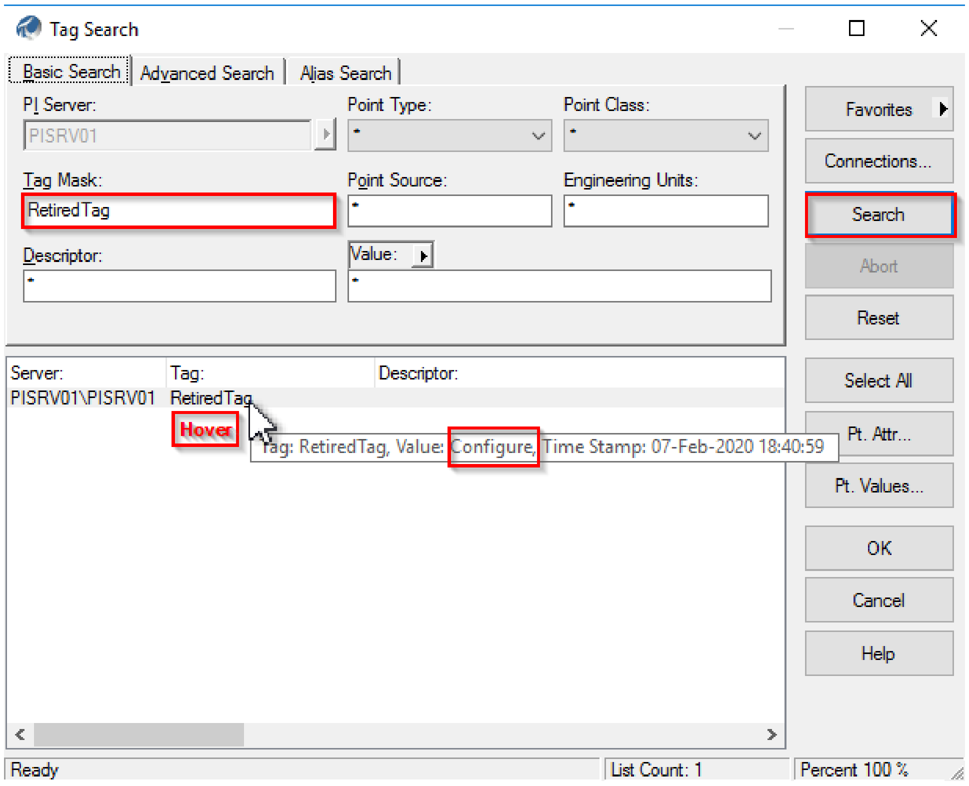


Looks like the value is indeed “Scan Off”. The user informs you it should be a number.



Tag Name	Server	Collective	Timestamp	Value	Engineering Units
BrokenTag	PISRV01	PISRV01	2/10/2020 9:33:26 PM	Scan Off	

Do another tag search (repeat the above steps) for RetiredTag but alternatively, instead of clicking OK, **just hover over RetiredTag, the current value will be shown in the tooltip.**



PI Server: PISRV01
Point Type: *
Point Class: *
Tag Mask: RetiredTag
Point Source: *
Engineering Units: *
Descriptor: *
Value: *

Server:	Tag:	Descriptor:
PISRV01\PISRV01	RetiredTag	

Tag: RetiredTag, Value: Configure, Time Stamp: 07-Feb-2020 18:40:59

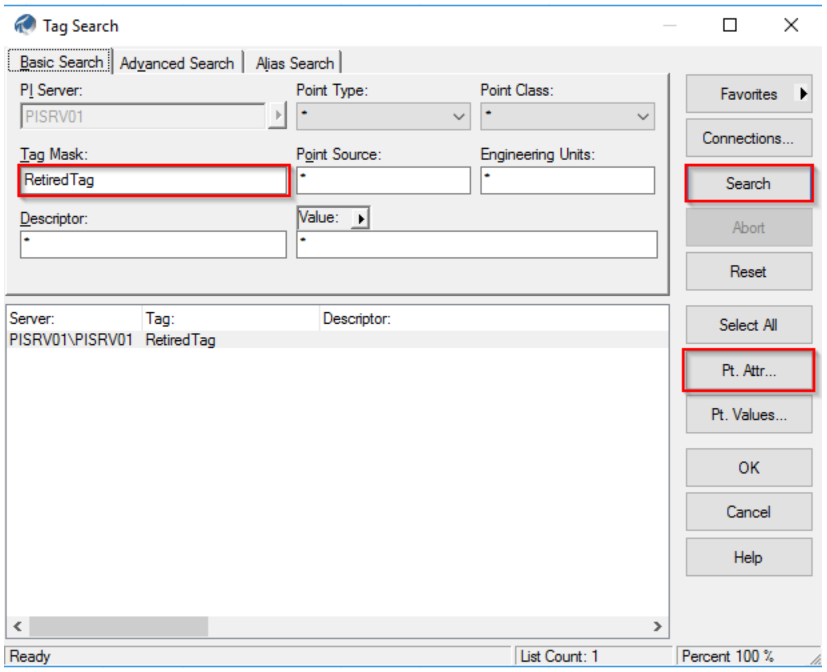
Ready | List Count: 1 | Percent 100 %

b. Identify the PI Interface Node

In time, you'll get a feel for how each tag gets its data, but let's say you know nothing at all about your PI System and have nobody to ask. How do you even get started tackling a bad tag? Generally, the first step is to get close to the source of the data, which often means logging in to the PI Interface machine. In many cases, it can be shown that the PI Interface is doing its job, but there is a problem with the data source, at which point it's the instrumentation person's problem.

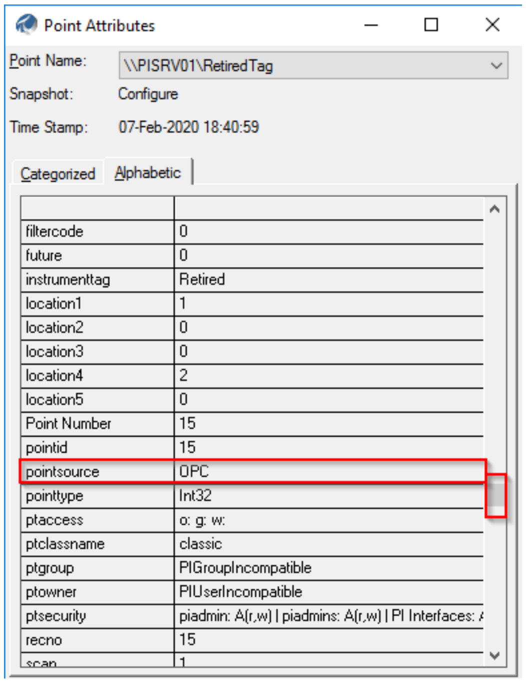
Logging into the PI Interface node is what we want to do now, but how do we know which server the PI Interface that writes to RetiredTag and BrokenTag is installed on?

Go back to the Tag Search dialog, and search for one of the bad tags again. **Enter RetiredTag for the Tag Mask, Hit search, Click on RetiredTag, and Click Pt Attr….**



Server:	Tag:	Descriptor:
PISRV01\PISRV01	RetiredTag	

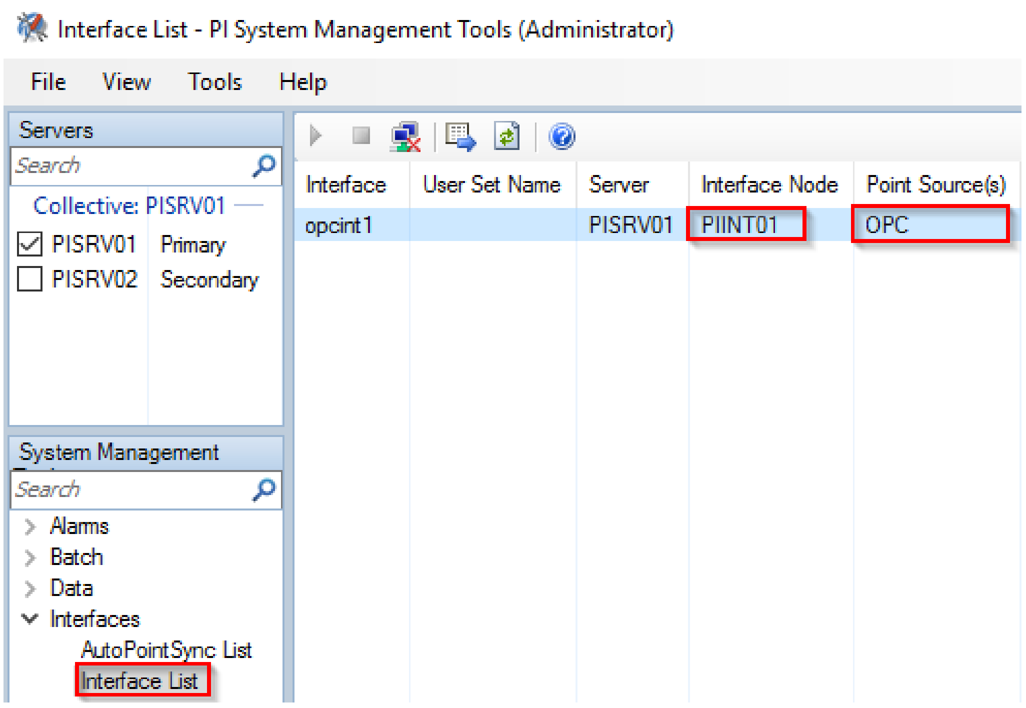
This will list the tag configuration attributes. Scroll down the alphabetically listed attributes until you find **pointsource**. The Point Source can be used to look up the PI Interface. In this case the Point Source is **OPC**.



Point Name: \\PISRV01\RetiredTag
Snapshot: Configure
Time Stamp: 07-Feb-2020 18:40:59

filtercode	0
future	0
instrumenttag	Retired
location1	1
location2	0
location3	0
location4	2
location5	0
Point Number	15
pointid	15
pointsource	OPC
pointtype	Int32
ptaccess	o: g: w:
ptclassname	classic
ptgroup	PIGroupIncompatible
ptowner	PIUserIncompatible
ptsecurity	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces:
recno	15
scan	1

Now in **PI SMT**, go to the **Interfaces -> Interface List** plugin and find **OPC** in the list. We can see that the interface that writes to these tags is on a machine named **PIINT01** and it's an OPC Interface.



PISRV01	Primary
PISRV02	Secondary

Interface	User Set Name	Server	Interface Node	Point Source(s)
opcint1		PISRV01	PIINT01	OPC

3. Directed Activity – Troubleshooting BrokenTag and RetiredTag

3.1 Objective

In this activity, we'll continue to investigate issues occurring with tags **BrokenTag** and **RetiredTag** by logging into the PI Interface Node (**PIINT01**) and inspecting the logs.

3.2 Tasks

- View PI Interface for OPC DA Startup logs to identify the issues with BrokenTag and RetiredTag
- Fix BrokenTag and RetiredTag by editing the tag attributes

3.3 Step-by-Step Instructions

a. View the PI Interface startup logs

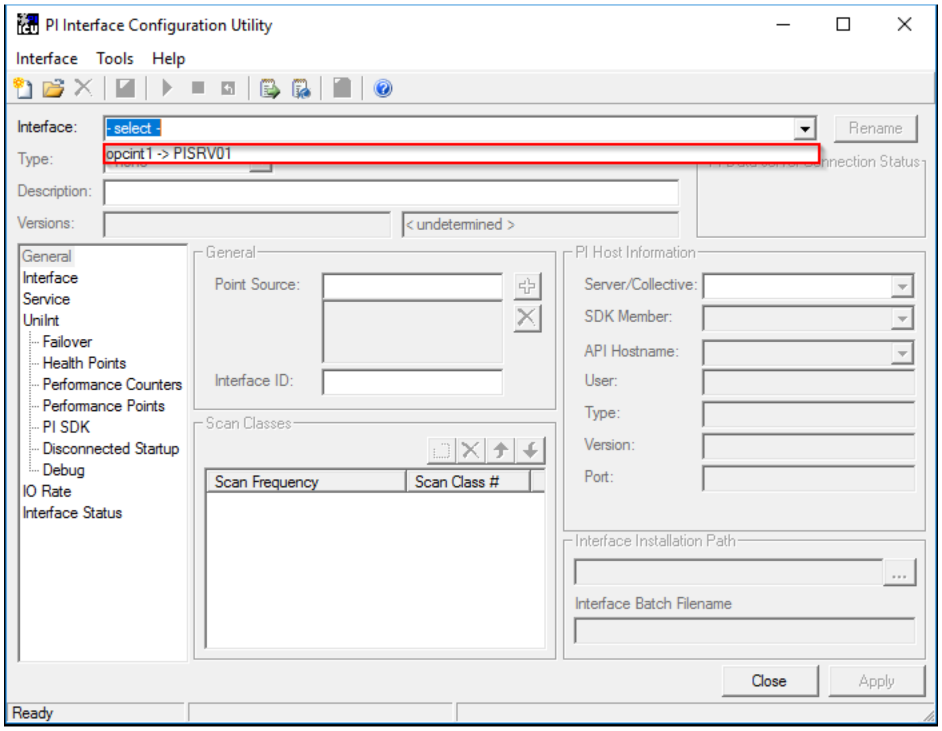
Make sure you're connected to **PIINT01** for the next few steps.

Often when tags are misbehaving, the issue can be identified by looking at the messages logged during the interface startup routine. The least intrusive method would technically be to figure out when the PI Interface last started and look at the logs from that time period.

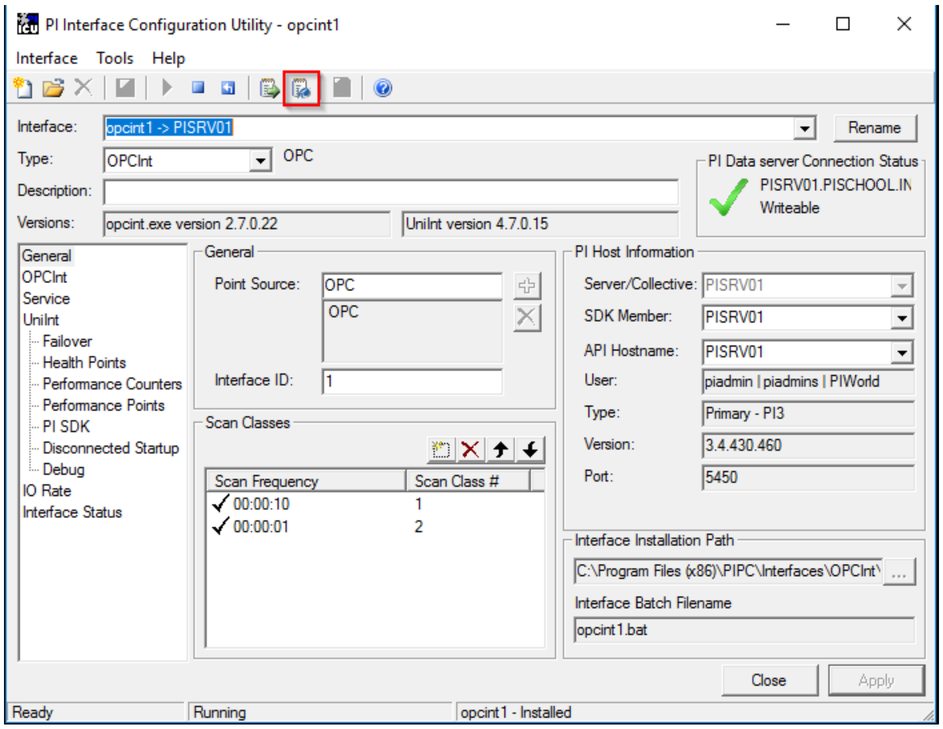
Note: Although not a best practice, some may choose to restart the PI Interface service and watch the logs in real-time. Restarting the PI Interface service is generally safe and can be done during troubleshooting and when adding new tags. Please note that during a restart, interfaces will not be able to collect data. For non-history recovery interfaces, there will be data loss. OSIsoft recommends setting up interface level redundancy to avoid this issue.

Perhaps the simplest way to view the logs is using the 'View current PI Message log continuously' button in the PI Interface Configuration Utility (PI ICU).

Launch PI ICU from the taskbar and **select opcint1** from the drop-down.



Click the ‘View current PI Message log continuously’ button.



Interface: opcint1 -> PISRV01
Type: OPCInt OPC
Description:
Versions: opcint.exe version 2.7.0.22 UniInt version 4.7.0.15

PI Data server Connection Status
PISRV01.PISCHOOL.IN
Writeable

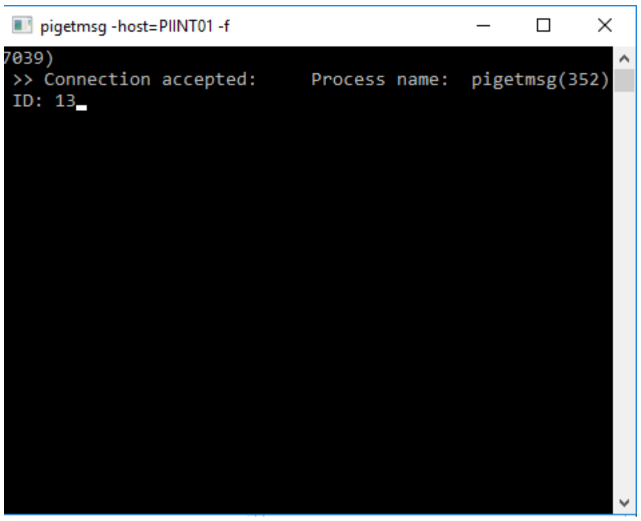
Point Source: OPC
OPC
Interface ID: 1

Scan Frequency	Scan Class #
✓ 00:00:10	1
✓ 00:00:01	2

Server/Collective: PISRV01
SDK Member: PISRV01
API Hostname: PISRV01
User: piadmin | piadmins | PIWorld
Type: Primary - PI3
Version: 3.4.430.460
Port: 5450

C:\Program Files (x86)\PIPC\Interfaces\OPCInt\
opcint1.bat

Ready | Running | opcint1 - Installed

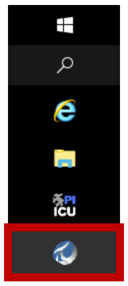


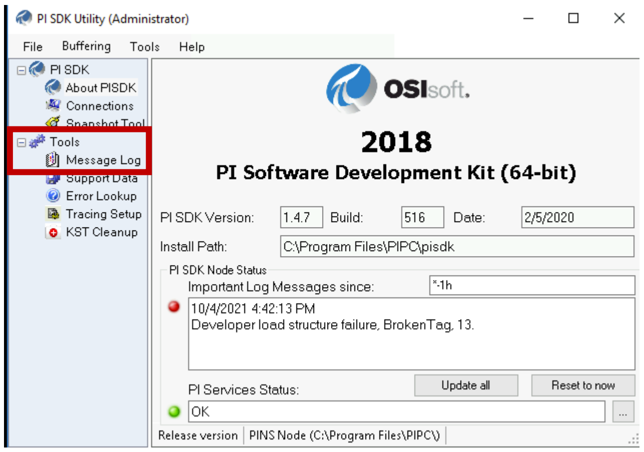
pigetmsg -host=PIINT01 -f

```
7039)
 >> Connection accepted:     Process name:  pigetmsg(352)
 ID: 13_
```

Keep the log running while we explore the errors and correct the set-up of the tags.

From the PIINT01 server, open the PI SDK Utility – the shortcut is pinned on the task bar.





PI SDK Version: 1.4.7 Build: 516 Date: 2/5/2020
Install Path: C:\Program Files\PIPC\pisdk
PI SDK Node Status
Important Log Messages since: *-1h

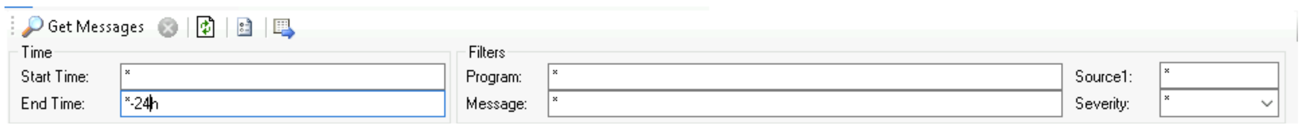
10/4/2021 4:42:13 PM
Developer load structure failure, BrokenTag, 13.

PI Services Status:
OK

Release version | PINS Node (C:\Program Files\PIPC\)

Select Message Log. Update the Severity to an *, so all potential messages can be viewed. The severity defaults to Error.

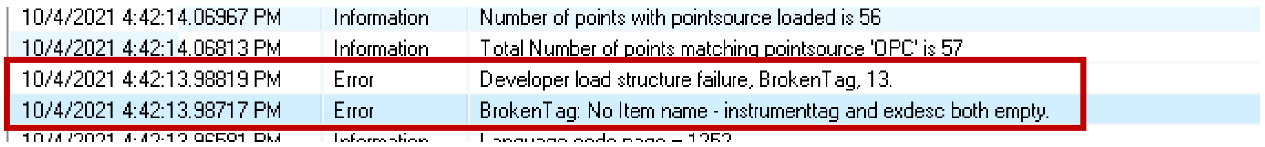
Change the end time to *-24h, most of our machines have not been running more than 24 hours.



Time
Start Time: *
End Time: *-24h

Filters
Program: *
Message: *
Source1: *
Severity: *

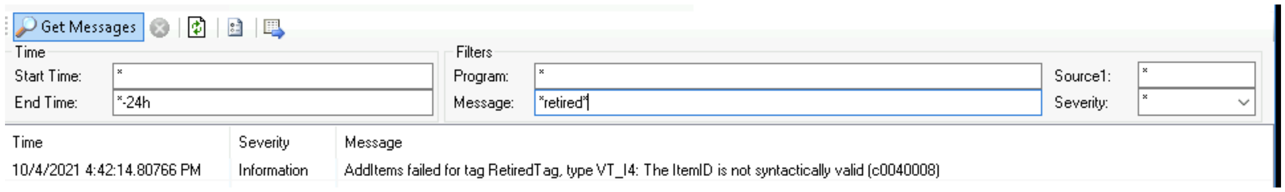
The errors below will indicate the issue with BrokenTag:



```
10/4/2021 4:42:14.06967 PM   Information   Number of points with pointsource loaded is 56
10/4/2021 4:42:14.06813 PM   Information   Total Number of points matching pointsource 'OPC' is 57
10/4/2021 4:42:13.98819 PM   Error         Developer load structure failure, BrokenTag, 13.
10/4/2021 4:42:13.98717 PM   Error         BrokenTag: No Item name - instrumenttag and exdesc both empty.
```

The above errors indicate that the instrument tag and extended descriptor fields of the PI Point “BrokenTag” are both empty. These field being empty is a problem because how the interface maps the PI Point to its corresponding Tag on the OPC server side. Since the tag was receiving values before, this likely happened due to human error when making changes to the PI Point.

The error below indicates the issue with RetiredTag. We simplified the message outputs by adding a filter of *retired* in the Message filter.



Time
Start Time: *
End Time: *-24h

Filters
Program: *
Message: *retired*
Source1: *
Severity: *

```
Time                         Severity      Message
10/4/2021 4:42:14.80766 PM   Information   AddItems failed for tag RetiredTag, type VT_I4: The ItemID is not syntactically valid (c0040008)
```

We can see that the interface is complaining about the ItemID, which is how the PI Tag maps to the OPC Tag. In a production environment, you might see different messages related to the ItemID.

The error message for tag “RetiredTag” means that the ItemID listed in the PI Tag configuration is incorrect. This can happen when the instrumentation changes, PLCs are removed, or configurations to the OPC Server are made.

At this point you would work with your instrumentation or control system team to determine the correct ItemID.

The instrumentation person informs you that indeed they swapped a PLC, and **the correct itemID for BrokenTag is Random.Int1**.

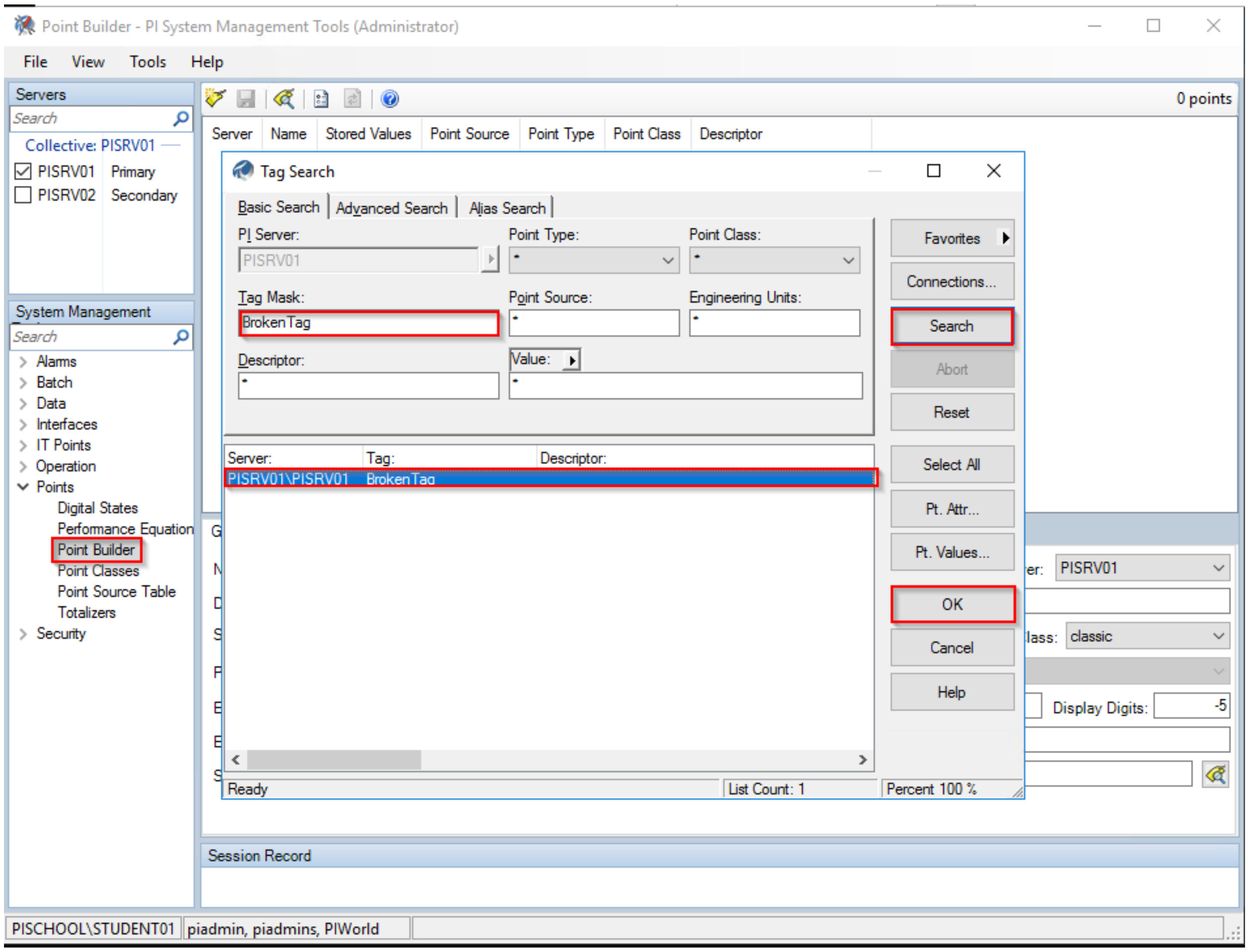
The instrumentation person also informs you that **RetiredTag no longer has any corresponding instrumentation. The equipment has been removed.** You now have a few options:

- **Leave it:** It's not really hurting anything, however it would be nice to stop logging the error message so that the logs are easier to work with.
- **Delete it:** If this data is never going to be needed again why not free up a tag in your license? You might have a policy to never delete data or tags however.
- **Keep it, but stop the interface from loading it:** You can keep the history just in case, but also prevent the error message from cluttering the logs.

b. Fix BrokenTag

Let’s fix BrokenTag first. **Go back to PI SMT on PICLIENT01**.

Go to the **Points -> Point Builder** plugin, **Search for BrokenTag**, select it, and click **OK**.



PI Server: PISRV01
Point Type: *
Point Class: *
Tag Mask: BrokenTag
Point Source: *
Engineering Units: *
Descriptor: *
Value: *

Server:	Tag:	Descriptor:
PISRV01\PISRV01	BrokenTag	

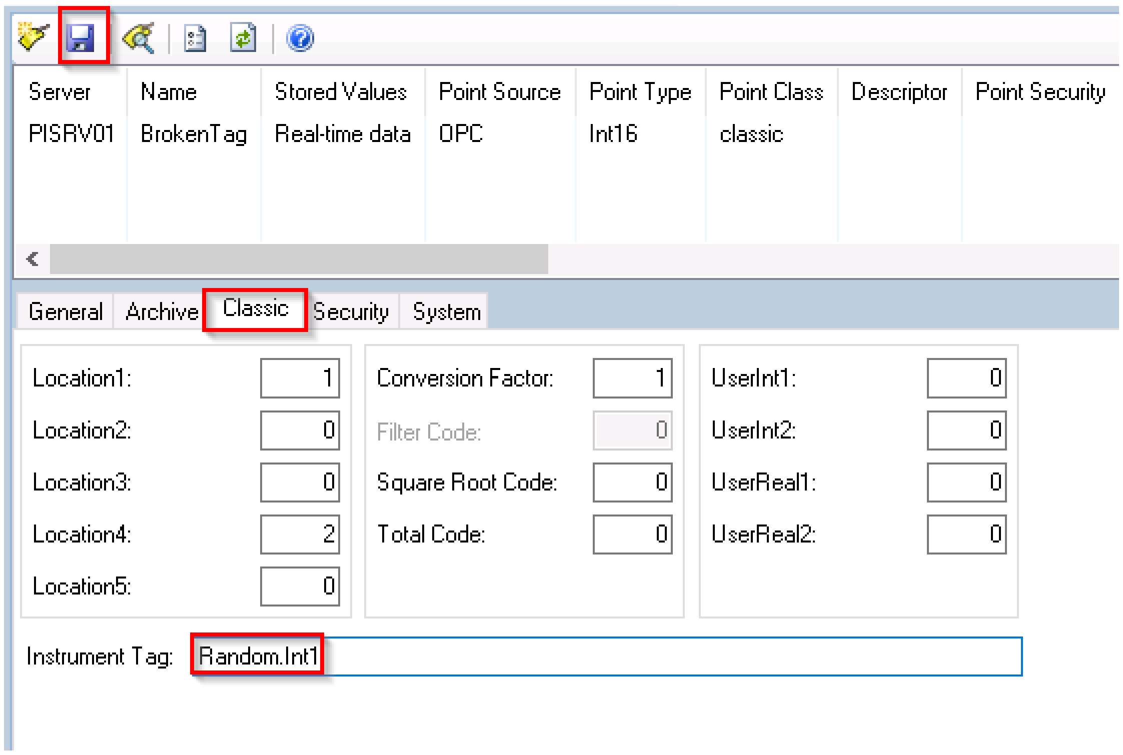
Ready | List Count: 1 | Percent 100 %

PISCHOOL\STUDENT01 | piadmin, piadmins, PIWorld

The ItemID is stored in the instrument tag point attribute.

To find this out you would have to consult the PI Interface manual or have this memorized from experience. Different point attributes are used for different things depending on the interface type. The interface manual will be a .pdf or word document somewhere in the PIPC folder. In our case you would **log into PIINT01 and look in C:\Program Files (x86)\PIPC\Interfaces\OPCInt1 for PI_OPCInt.pdf**. The important section to search for is “Configure point attributes”, which is a common section to all PI Interface manuals.

Select the Classic tab, enter **Random.Int1** as the instrument tag, and **save**.

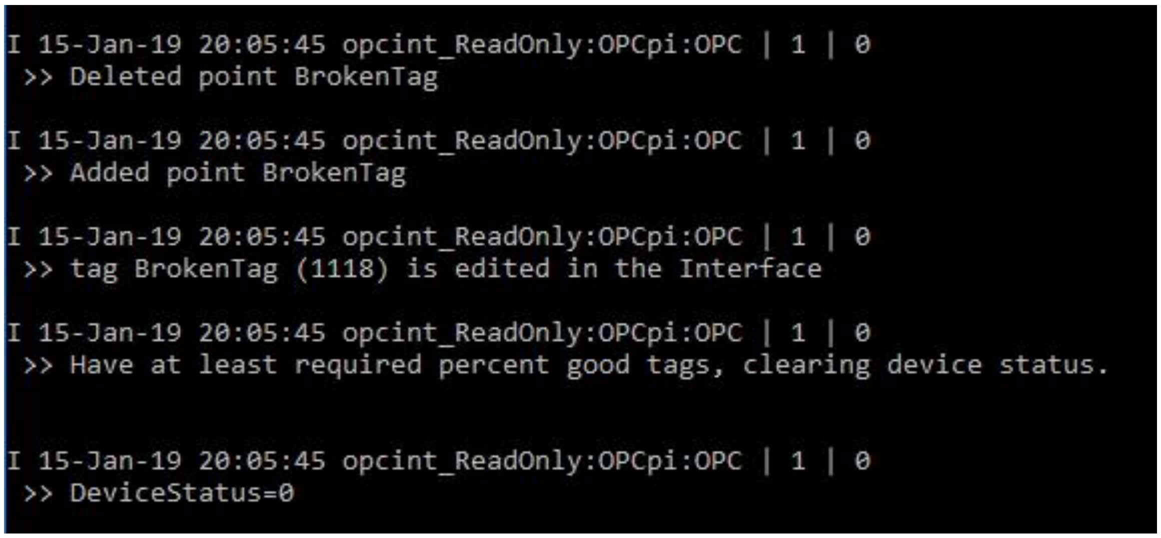


Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor	Point Security
PISRV01	BrokenTag	Real-time data	OPC	Int16	classic		

Location1:	1	Conversion Factor:	1	UserInt1:	0
Location2:	0	Filter Code:	0	UserInt2:	0
Location3:	0	Square Root Code:	0	UserReal1:	0
Location4:	2	Total Code:	0	UserReal2:	0
Location5:	0				

Instrument Tag: Random.Int1

If you wait a few minutes, the PI Interface will naturally pick up the change and start writing to the tag. **However, there is no need to wait. Move to the next page.**

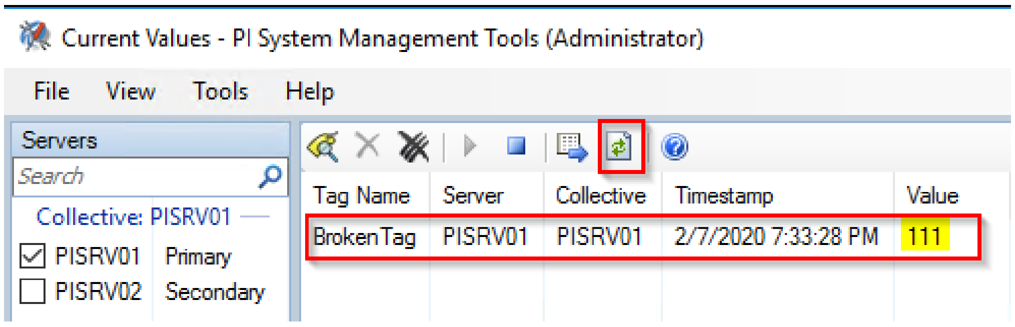


```
I 15-Jan-19 20:05:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
 >> Deleted point BrokenTag

I 15-Jan-19 20:05:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
 >> Added point BrokenTag

I 15-Jan-19 20:05:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
 >> tag BrokenTag (1118) is edited in the Interface

I 15-Jan-19 20:05:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
 >> Have at least required percent good tags, clearing device status.


I 15-Jan-19 20:05:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
 >> DeviceStatus=0
```

If you want to see the change right away instead of waiting, restart the PI Interface service, but you won't see the message in the previous screenshot during an interface startup.

From **PICLIENT01**, go back to the **Current Values plugin** and refresh to confirm that the tag is now getting data:

Everyone's values will be different and will not match the workbook because this is randomly generated data!



Tag Name	Server	Collective	Timestamp	Value
BrokenTag	PISRV01	PISRV01	2/7/2020 7:33:28 PM	111

c. Retire RetiredTag

Now to retire RetiredTag.

One way to do this is to set the scan attribute to off. The PI Interface will load the tag at startup, but then set the value to Scan Off and offload the tag.

Another method is to assign an unused Point Source dedicated to retired tags, such as ZZZ or ZZZ_OPC.

The best practice is actually to do both:

- **Scan Off** gives a strong signal to users that the tag has been turned off since they will see "Scan Off" when they query the value. Not everyone will look at the Point Source when searching for tags.
- **Assigning a retired Point Source** will prevent the interface from loading the tag at all. This means unnecessary messages will not be logged, and interface startup times will be improved.

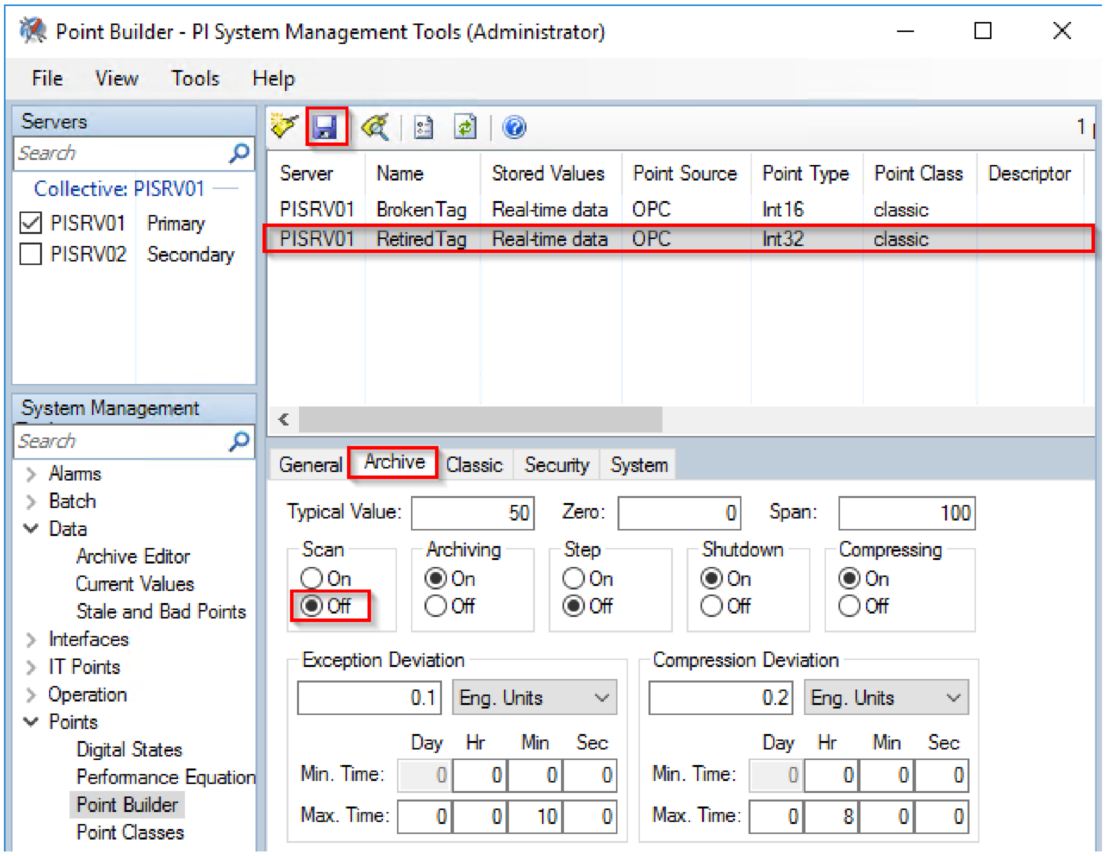
✓

Best Practice

Retire tags by assigning an unused point source in addition to setting the scan attribute to off. This will prevent the PI Interface from loading and evaluating the tag upon startup.

From **PICLIENT01**, go back to the **Point Builder plugin** and search for **RetiredTag**, and add it to the list.

Select **RetiredTag**, go to the Archive tab, click the radio button to set Scan to Off, then save.



Collective: PISRV01

☑	PISRV01	Primary
☐	PISRV02	Secondary

Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor
PISRV01	BrokenTag	Real-time data	OPC	Int16	classic	
PISRV01	RetiredTag	Real-time data	OPC	Int32	classic	

Typical Value: 50 Zero: 0 Span: 100

Scan: On [✓ Off]
Archiving: [✓ On] Off
Step: On [✓ Off]
Shutdown: [✓ On] Off
Compressing: [✓ On] Off

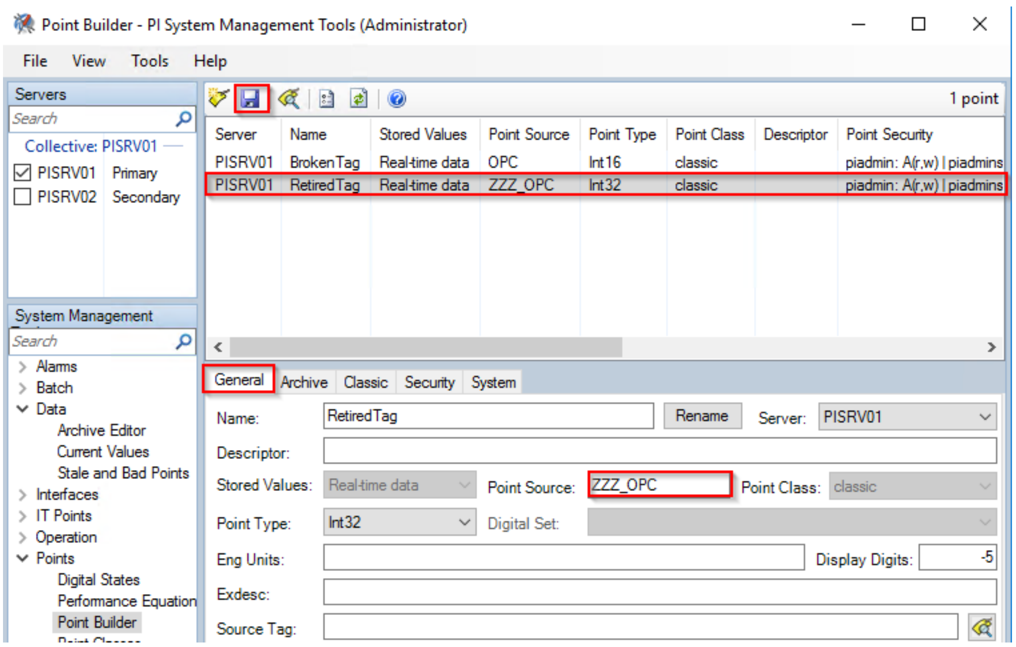
Exception Deviation: 0.1 Eng. Units

	Day	Hr	Min	Sec
Min. Time:	0	0	0	0
Max. Time:	0	0	10	0

Compression Deviation: 0.2 Eng. Units

	Day	Hr	Min	Sec
Min. Time:	0	0	0	0
Max. Time:	0	8	0	0

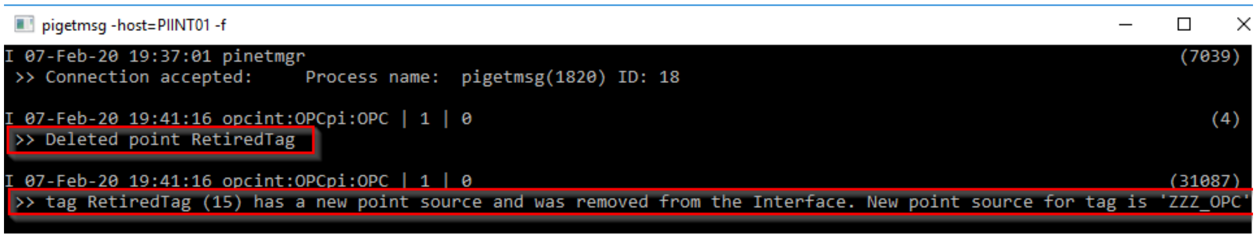
Now set the Point Source in the General tab to ZZZ_OPC and save



Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor	Point Security
PISRV01	BrokenTag	Real-time data	OPC	Int16	classic		piadmin: A(r,w) | piadmins
PISRV01	RetiredTag	Real-time data	ZZZ_OPC	Int32	classic		piadmin: A(r,w) | piadmins

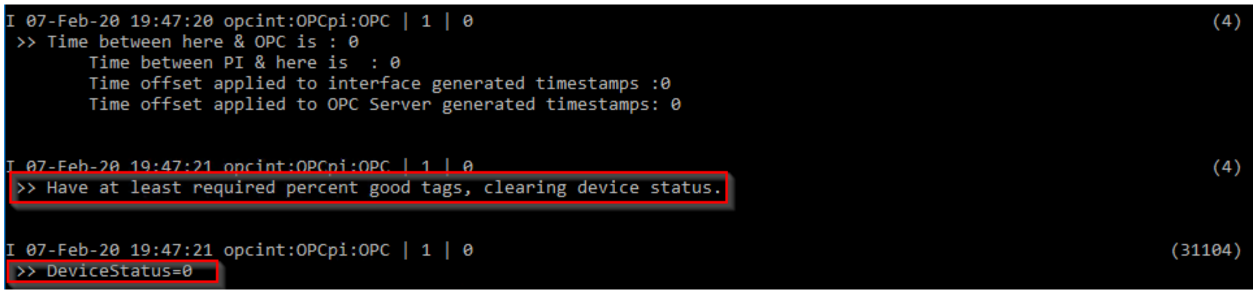
Name: RetiredTag
Server: PISRV01
Descriptor:
Stored Values: Real-time data
Point Source: ZZZ_OPC
Point Class: classic
Point Type: Int32
Digital Set:
Eng Units:
Display Digits: -5
Exdesc:
Source Tag:

If you look at the logs on **PIINT01** after a few minutes, you'll see this:



```
I 07-Feb-20 19:37:01 pinetmgr                                                                          (7039)
 >> Connection accepted:     Process name:  pigetmsg(1820) ID: 18

I 07-Feb-20 19:41:16 opcint:OPCpi:OPC | 1 | 0                                                             (4)
 >> Deleted point RetiredTag

I 07-Feb-20 19:41:16 opcint:OPCpi:OPC | 1 | 0                                                         (31087)
 >> tag RetiredTag (15) has a new point source and was removed from the Interface. New point source for tag is 'ZZZ_OPC'
```

Go back to PIINT01 and restart the interface again. Confirm that there are no complaints about the ItemID in the logs.

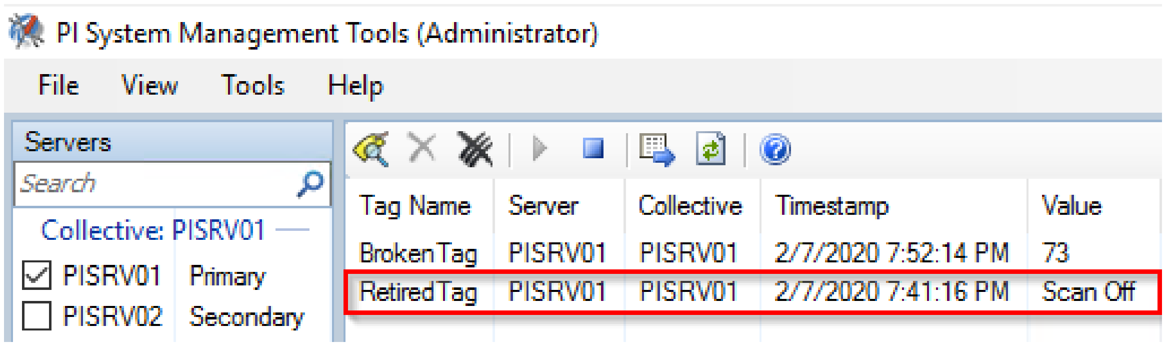


```
I 07-Feb-20 19:47:20 opcint:OPCpi:OPC | 1 | 0                                                             (4)
 >> Time between here & OPC is : 0
        Time between PI & here is  : 0
        Time offset applied to interface generated timestamps :0
        Time offset applied to OPC Server generated timestamps: 0

I 07-Feb-20 19:47:21 opcint:OPCpi:OPC | 1 | 0                                                             (4)
 >> Have at least required percent good tags, clearing device status.

I 07-Feb-20 19:47:21 opcint:OPCpi:OPC | 1 | 0                                                         (31104)
 >> DeviceStatus=0
```

DeviceStatus=0 is not a guarantee that all tags are working, but it's the best possible Device Status and a good sign that the main checks passed.

Finally, check the Current Value of RetiredTag.

Yours may still show Configure depending on whether you waited for the PI interface to detect the Scan off change.


PI System Management Tools (Administrator)

File View Tools Help

Servers

Collective: PISRV01

☑ PISRV01	Primary
☐ PISRV02	Secondary

Tag Name	Server	Collective	Timestamp	Value
BrokenTag	PISRV01	PISRV01	2/7/2020 7:52:14 PM	73
RetiredTag	PISRV01	PISRV01	2/7/2020 7:41:16 PM	Scan Off

Success!

d. Define the relationship between PI Point Attribute and PI Interface configuration

We began our discussion of PI Point Attributes in the previous section. As we saw in the previous exercise, there is a direct relationship between specific PI Point attributes and the PI Interface instance that is collecting the PI Point data.

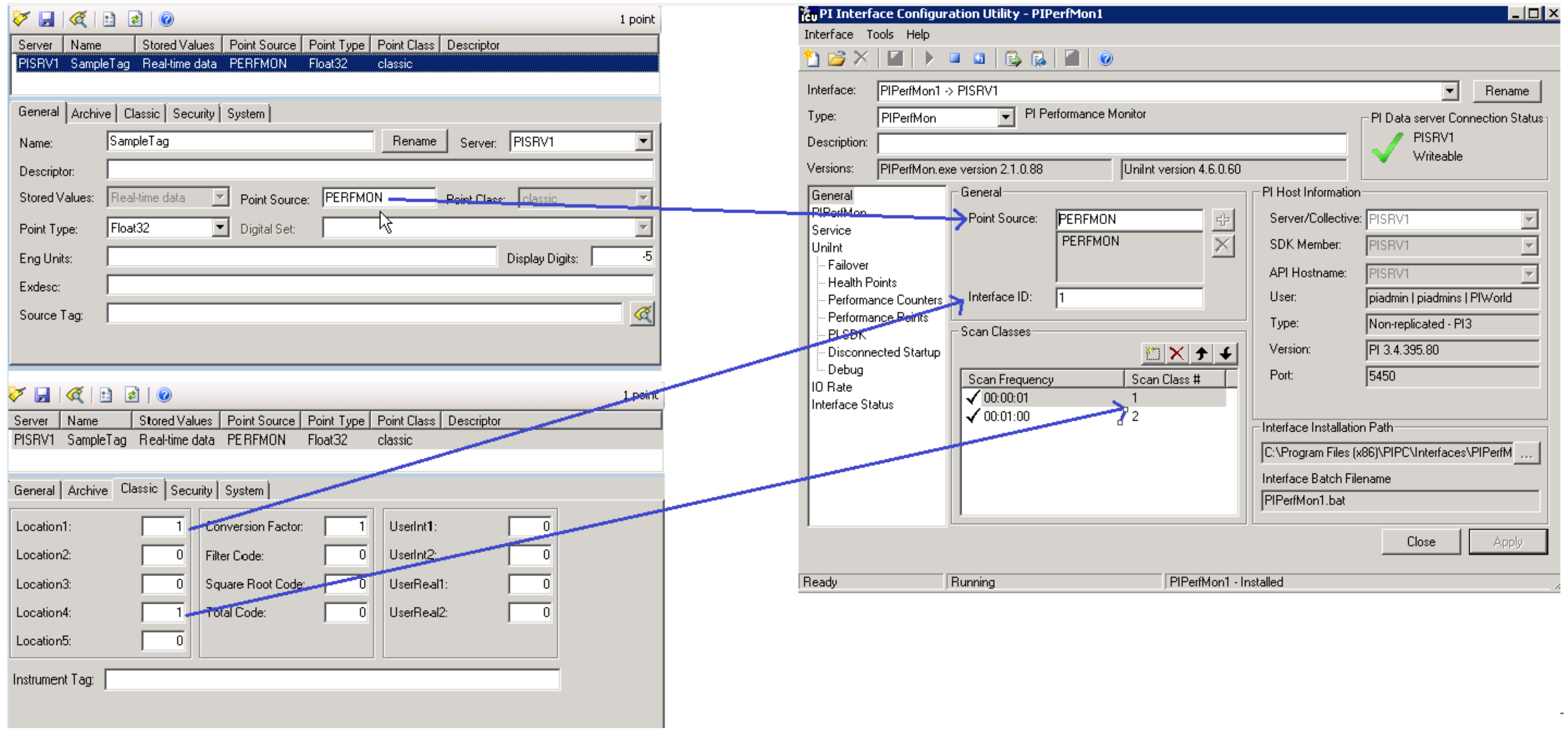
The exact relationship is unique to each different PI Interface. Listed below are the common PI Point attributes and how they are ***typically*** used. ***ALWAYS consult the interface manual when creating PI Points.***

Instrument Tag	Name of the point/location in the source data system. *Often it is case sensitive and must match the data source exactly!*
Extended Descriptor	Place for detailed query instructions (uncommon).
Point Source	Must match the point source of the interface.
Location 1	*Typically*, this field is used for the interface instance ID. In this case, the unique combination of the point source + interface id is what links a PI Point to its PI Interface instance.
Location 4	*Typically,* this field is the scan class number.
Scan	Include the PI point in the list of points to scan (always set to ON)

The most common cause of new PI Points not receiving data is an incorrect configuration of the PI Point attributes according to the data source of PI Interface instance configuration. In exercise 3a, we saw that the PI Point “BrokenTag” was not updating because its instrument tag was not correct. In exercise 3b, we saw that we could prevent the PI Point “RetiredTag” from being picked up by the interface at startup by changing its point source. The following image shows the relationship between some PI Point attributes in PI SMT and PI Interface parameters in PI ICU.

PI SMT--> Points--> Point Builder

PI ICU



1 point

Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor
PISRV1	SampleTag	Real-time data	PERFMON	Float32	classic	

General | Archive | Classic | Security | System

Name: SampleTag
Server: PISRV1
Descriptor:
Stored Values: Real-time data
Point Source: PERFMON
Point Class: classic
Point Type: Float32
Digital Set:
Eng Units:
Display Digits: -5
Exdesc:
Source Tag:

1 point

Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor
PISRV1	SampleTag	Real-time data	PERFMON	Float32	classic	

General | Archive | Classic | Security | System

Location1:	1	Conversion Factor:	1	UserInt1:	0
Location2:	0	Filter Code:	0	UserInt2:	0
Location3:	0	Square Root Code:	0	UserReal1:	0
Location4:	1	Total Code:	0	UserReal2:	0
Location5:	0				

Instrument Tag:

PI Interface Configuration Utility - PIPerfMon1

Interface: PIPerfMon1 -> PISRV1
Type: PIPerfMon PI Performance Monitor
Description:
Versions: PIPerfMon.exe version 2.1.0.88 UniInt version 4.6.0.60
PI Data server Connection Status: PISRV1 Writeable

General
Point Source: PERFMON
PERFMON
Interface ID: 1

Scan Classes

Scan Frequency	Scan Class #
00:00:01	1
00:01:00	2

PI Host Information
Server/Collective: PISRV1
SDK Member: PISRV1
API Hostname: PISRV1
User: piadmin | piadmins | PIWorld
Type: Non-replicated - PI3
Version: PI 3.4.395.80
Port: 5450

Interface Installation Path: C:\Program Files (x86)\PIPC\Interfaces\PIPerfM
Interface Batch Filename: PIPerfMon1.bat

Ready | Running | PIPerfMon1 - Installed

4. Directed Activity – The PI Collective Misconception

4.1 Objective

In this activity, we'll troubleshoot another common misconfiguration and get a better understanding of how PI Collectives work.

One of your users is now complaining that the tag "**ExampleTag**" has flatlined in their PI Vision display. He claims that the tag should be getting a new value every second. He sends you the PI Vision display and asks for your assistance with figuring out the issue.

4.2 Tasks

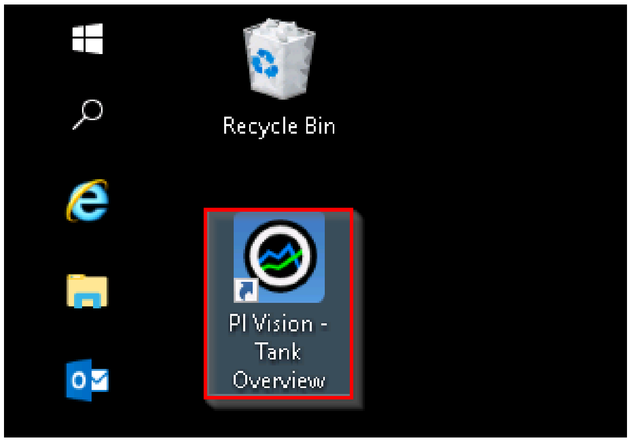
- Manually switch connections between PI Collective members
- Compare Current Values between PI Collective members
- Configure PI Buffer Subsystem
- Do bulk tag edits using PI Builder
- Reinitialize a PI Collective

4.3 Step-by-Step Instructions

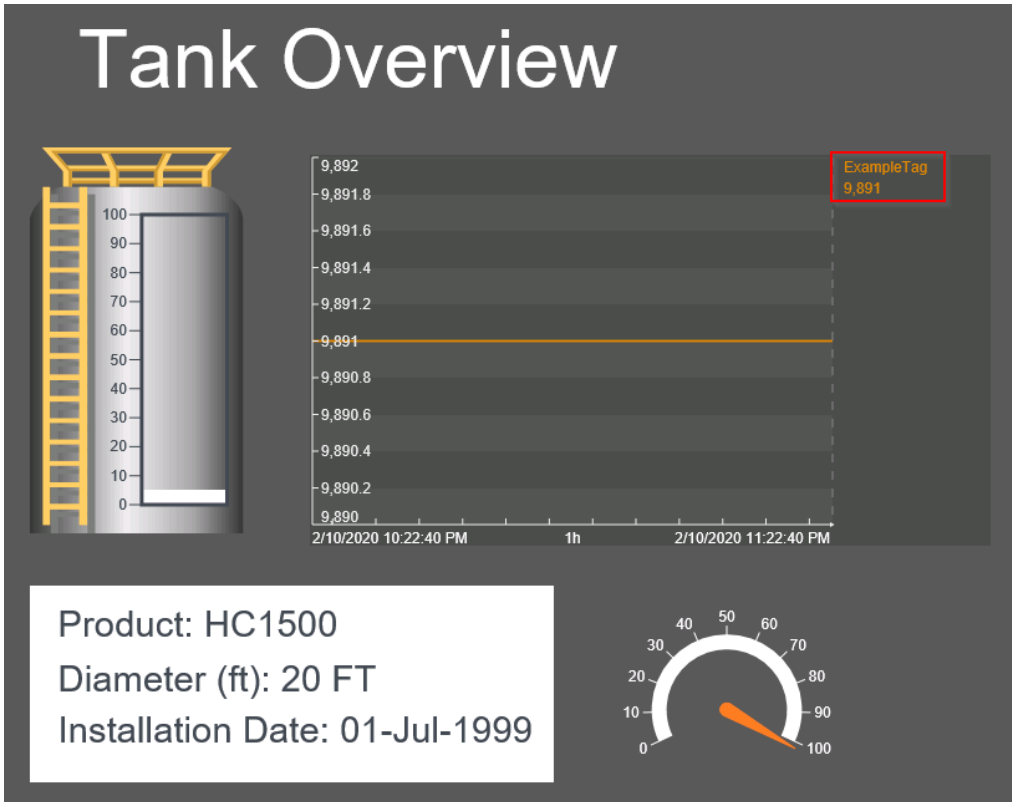
a. Open the display to verify the described behavior

Note: Everyone's values will be different and will not match the workbook because this is randomly generated data!

You want to check the PI Vision display to confirm the behavior. From **PICLIENT01**, open the PI Vision Shortcut on the desktop.



You see that the data has indeed flatlined.

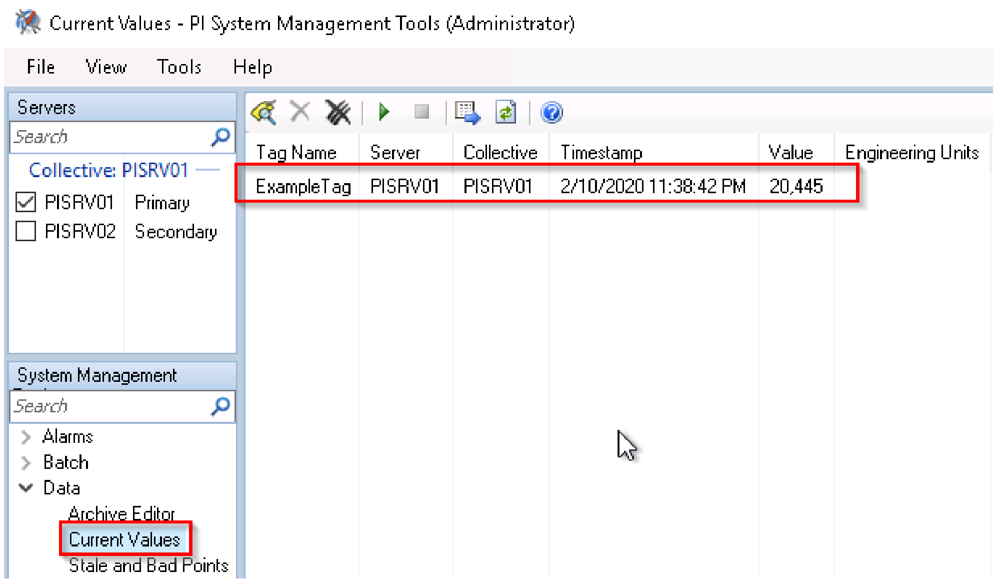


Product: HC1500
Diameter (ft): 20 FT
Installation Date: 01-Jul-1999

b. Begin Troubleshooting

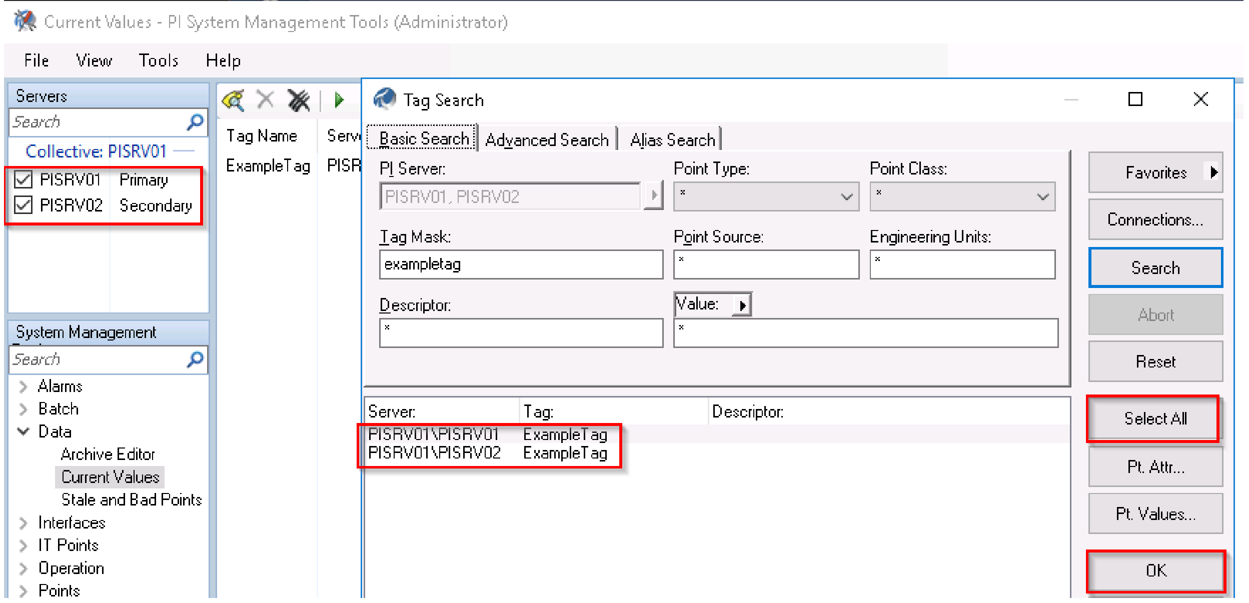
You don't really know what to do, so you start flailing around with the buttons and menus. You then decide to use another client tool to verify the data.

From **PICLIENT01**, you go to **PI SMT > Data > Current Values** and search for "**ExampleTag**"



Tag Name	Server	Collective	Timestamp	Value	Engineering Units
ExampleTag	PISRV01	PISRV01	2/10/2020 11:38:42 PM	20.445	

You then realize that the current value of **ExampleTag** is different than the one on the display. You remember that the server is in a PI collective and decide to check the current value of ExampleTag on the secondary member of the collective. You check the box of the secondary server to connect to it and do the same search again

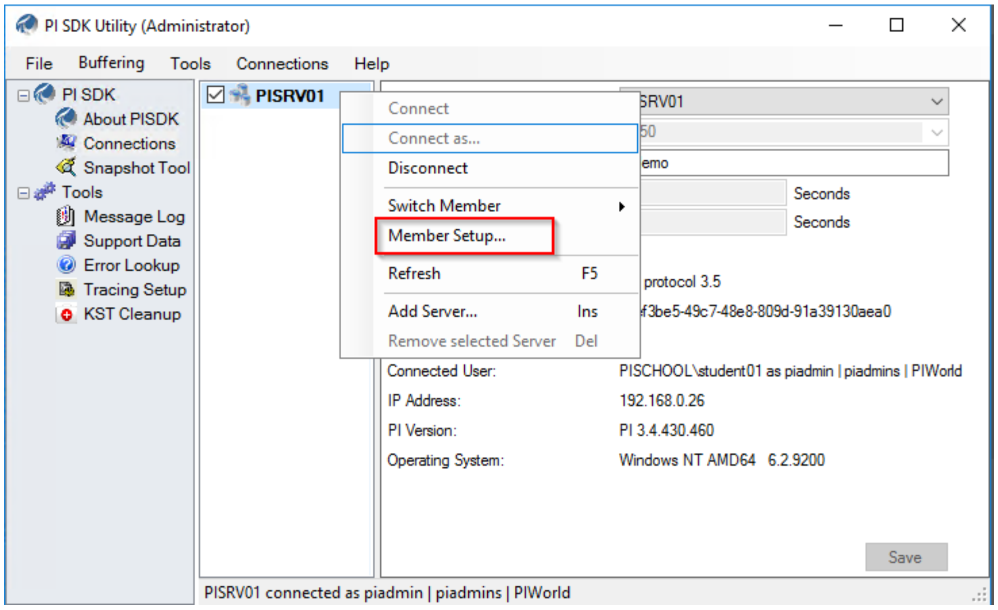


PI Server: PISRV01, PISRV02
Point Type: *
Point Class: *
Tag Mask: exampletag
Point Source: *
Engineering Units: *
Descriptor: *
Value: *

Server:	Tag:	Descriptor:
PISRV01\PISRV01	ExampleTag	
PISRV01\PISRV02	ExampleTag	

The value on the secondary member matches the flatlined value in PI Vision. This means that we are pointing to the secondary member of the collective on the PI Vision Server.

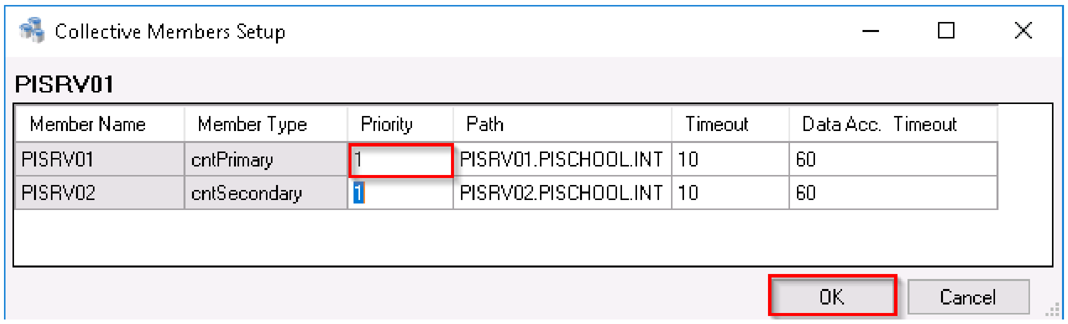
You log into **PISRV01**, open the **PI SDK Utility > Connections > right-click on the PI Data Archive collective > Member Setup…**



SRV01
50
emo
protocol 3.5
f3be5-49c7-48e8-809d-91a39130aea0
Connected User: PISCHOOL\student01 as piadmin | piadmins | PIWorld
IP Address: 192.168.0.26
PI Version: PI 3.4.430.460
Operating System: Windows NT AMD64 6.2.9200

PISRV01 connected as piadmin | piadmins | PIWorld

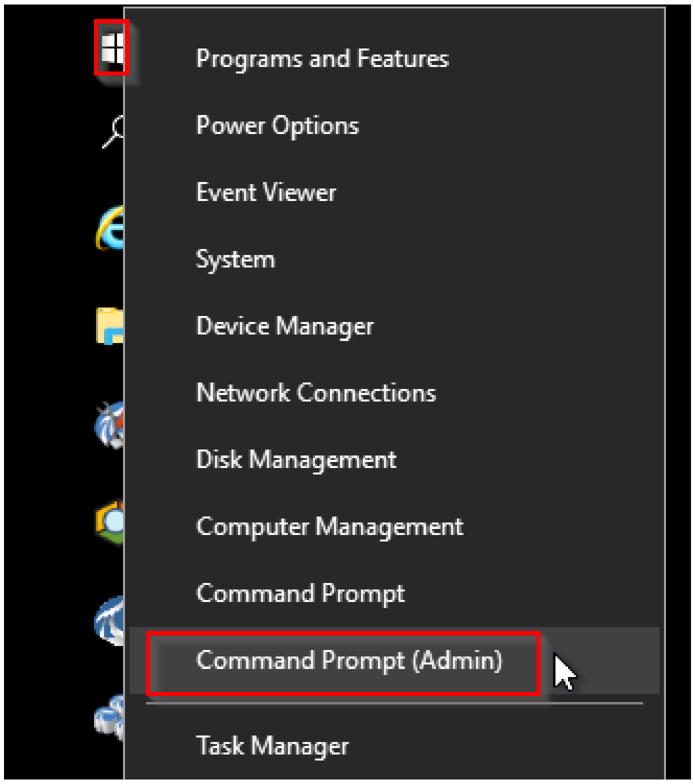
You notice that the priority of the primary member of the collective is **-1** which means that the secondary member will always be connected to first. **Change the priority of the primary to 1 > hit enter on your keyboard > click OK.**



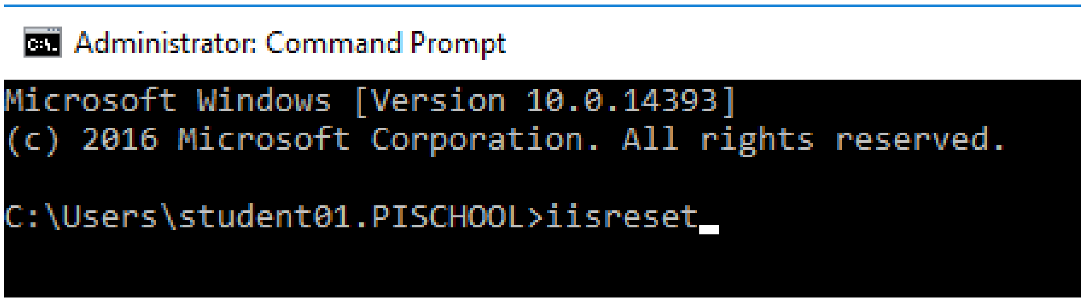
PISRV01

Member Name	Member Type	Priority	Path	Timeout	Data Acc. Timeout
PISRV01	cntPrimary	1	PISRV01.PISCHOOL.INT	10	60
PISRV02	cntSecondary	1	PISRV02.PISCHOOL.INT	10	60

Right-click on the start menu > select "Command prompt (Admin)"



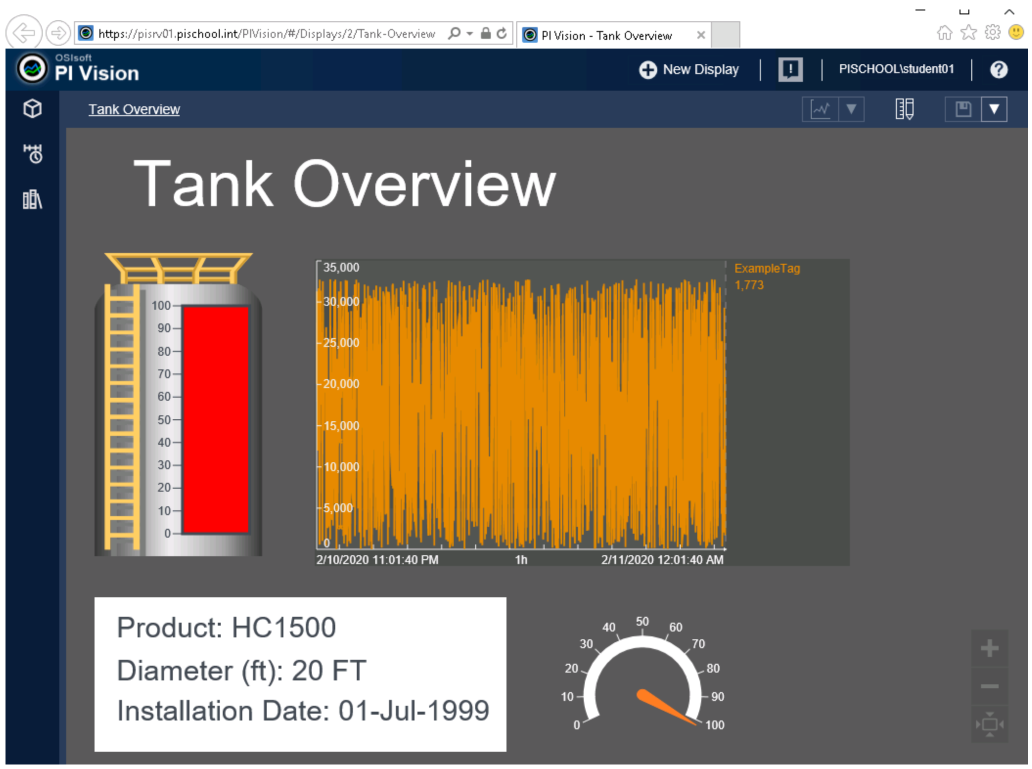
In the command prompt, type in **iisreset**:



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\student01.PISCHOOL>iisreset_
```

You will see the message **"Internet services successfully restarted"** once the operation is completed.

Go back to **PICLIENT01**, close the internet explorer browser page with PI Vision open, then double click on the display shortcut on your desktop one more time to reopen it.



Product: HC1500
Diameter (ft): 20 FT
Installation Date: 01-Jul-1999

Woo! We've got data flowing again. Well, not really. We merely switched collective priority and forced PI Vision to point to the primary member of the collective which is getting data for ExampleTag every second. The fact that the display is showing the data is a good first step because the end user can keep working while you troubleshoot the issue with the secondary member of the collective.

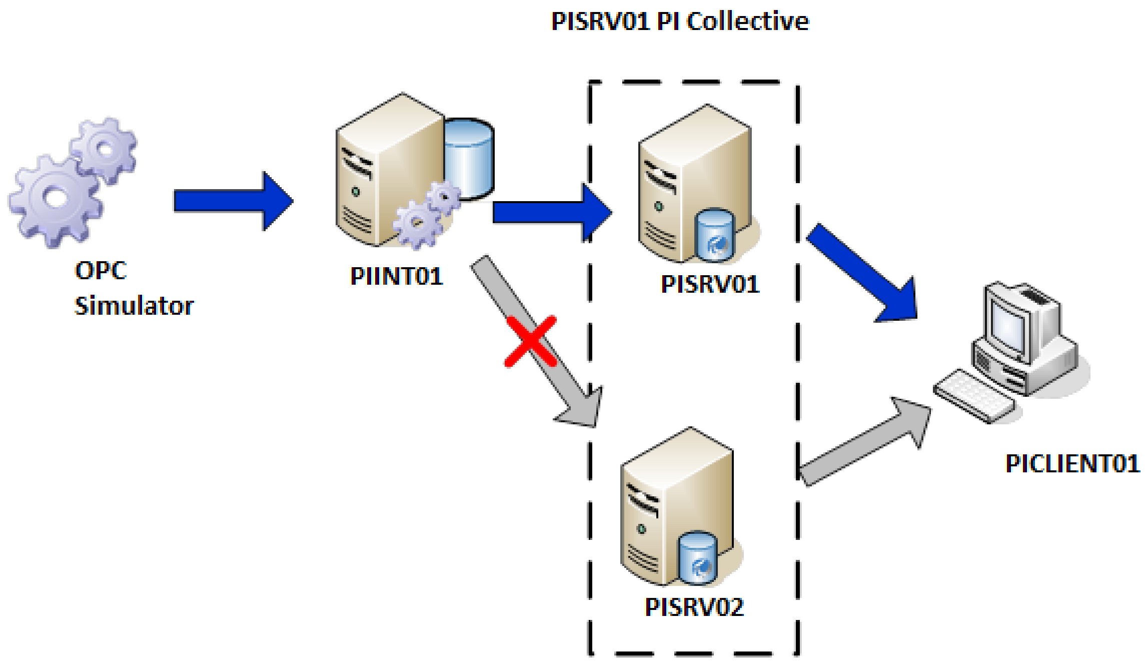
Tip

A very common misconception about PI Collectives is that there is some server-side mechanism keeping all the data synchronized. Many PI System users and administrators assume that any data written to the Primary is replicated to the Secondary, similar to SQL Server replication, or that PI Collectives work like a Windows Cluster with shared storage. Neither of these assumptions are true.

There are 2 main ways that data gets to the Secondary:

- The first is during a process called **Reinitialization**, where the archive files are copied from the Primary to the Secondary when requested by the Administrator using the PI Collective Manager utility. This operation is performed when the PI Collective is formed initially, but otherwise typically is only done on-demand when it is discovered that data is missing. **Data is missing, so we will need to do this, but we need to address another issue first.**
- The second is through **the use of PI Buffer Subsystem**, which takes data collected by PI Interfaces, PI Analysis Service, and other data entry applications and sends a copy to all PI Collective members. This is how data remains "in-sync" (for lack of a better term) during continuous operation.

The behavior we have just observed is almost always an issue with buffering. Either PI Buffer Subsystem is misconfigured, not configured at all, or there is a network or security issue preventing PI Buffer Subsystem from writing to tags on the Secondary.

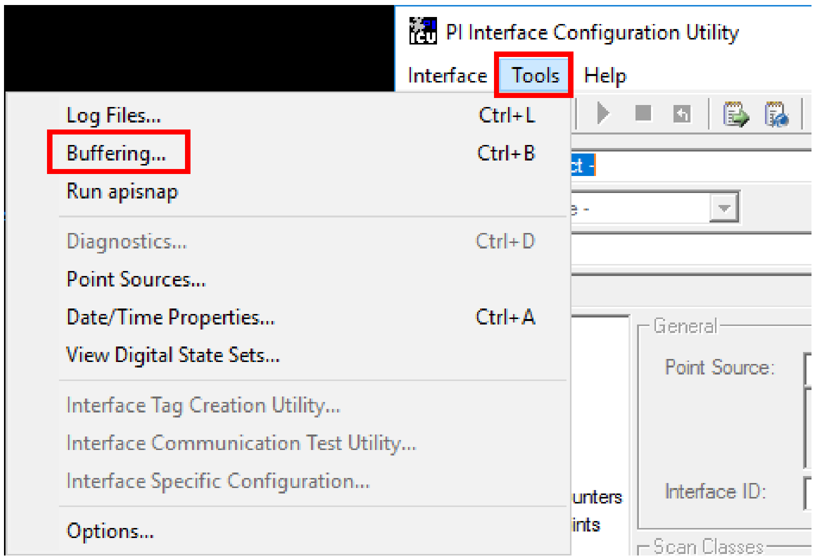


PI Buffer Subsystem must be configured on every node with an application that writes to the PI Data Archives. This of course includes PI Interfaces.

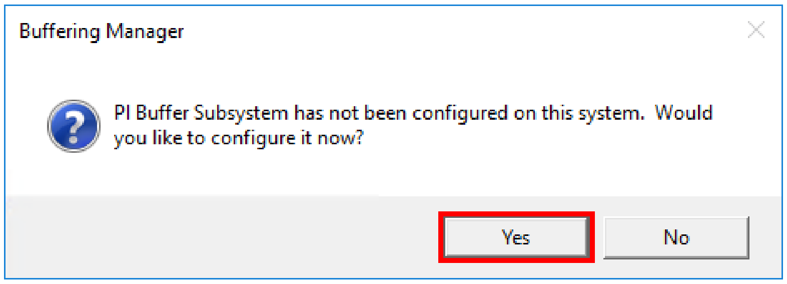
c. Check the PI Interface Node and configure PI Buffer Subsystem

Let's log into **PIINT01** and see what's wrong.

On **PIINT01**, open **PI ICU and do Tools -> Buffering**:

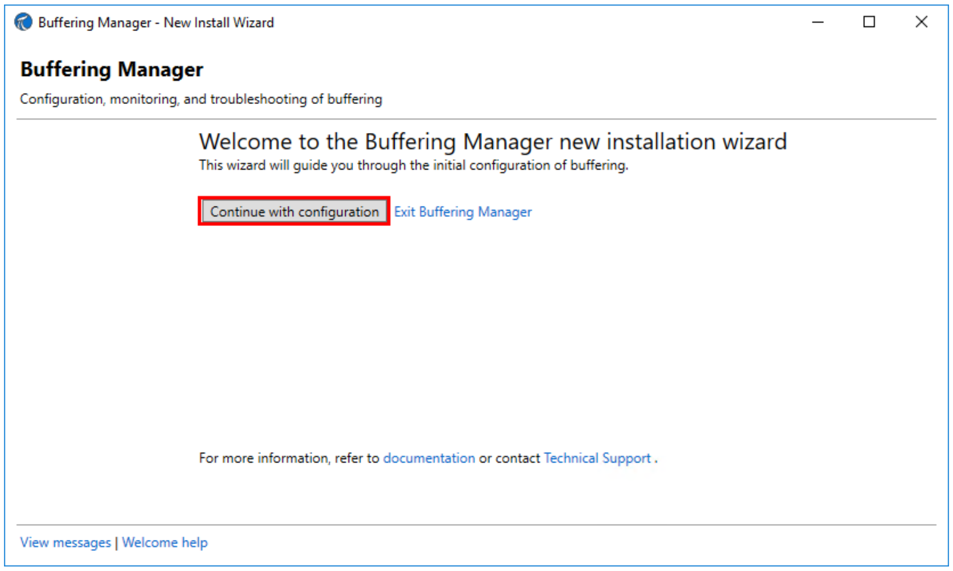

PI Interface Configuration Utility
Interface
Tools
Help
Log Files... Ctrl+L
Buffering... Ctrl+B
Run apisnap
Diagnostics... Ctrl+D
Point Sources...
Date/Time Properties... Ctrl+A
View Digital State Sets...
Interface Tag Creation Utility...
Interface Communication Test Utility...
Interface Specific Configuration...
Options...
General
Point Source:
Interface ID:
Scan Classes

Well that would explain it. PI Buffer Subsystem is not configured. Let's configure it:


Buffering Manager

PI Buffer Subsystem has not been configured on this system. Would you like to configure it now?

Click Continue with configuration:


Buffering Manager - New Install Wizard

Buffering Manager

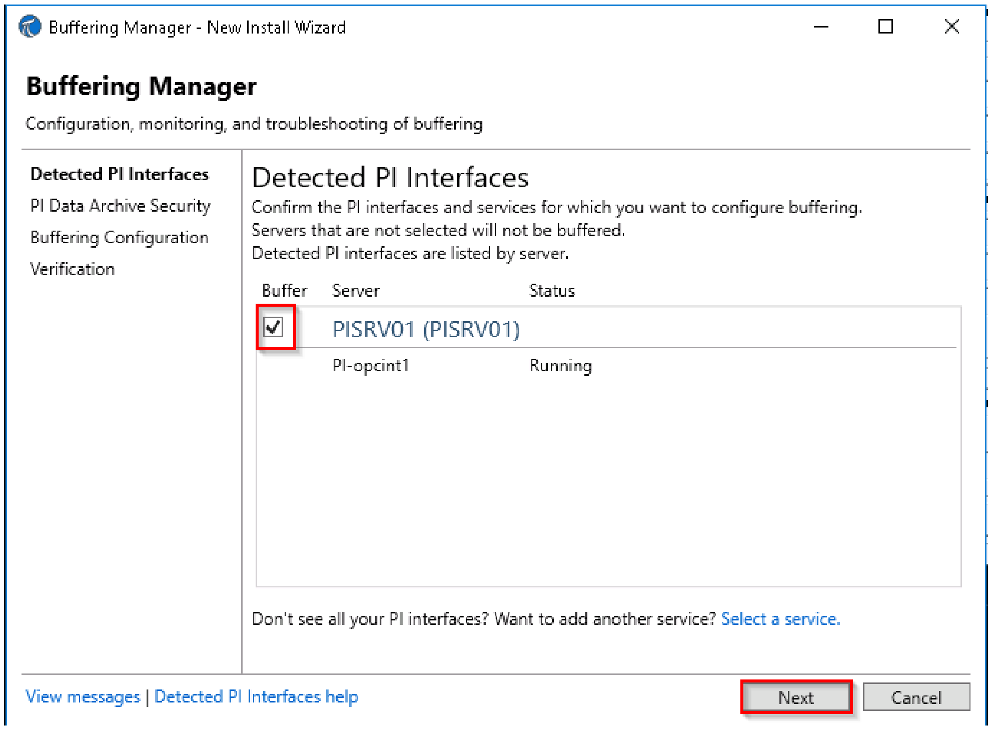
Configuration, monitoring, and troubleshooting of buffering

Welcome to the Buffering Manager new installation wizard

This wizard will guide you through the initial configuration of buffering.

For more information, refer to documentation or contact Technical Support .

Select the PI Data Archive / PI Collective you wish to buffer to, Next:


Buffering Manager - New Install Wizard

Buffering Manager

Configuration, monitoring, and troubleshooting of buffering

Detected PI Interfaces
PI Data Archive Security
Buffering Configuration
Verification

Detected PI Interfaces

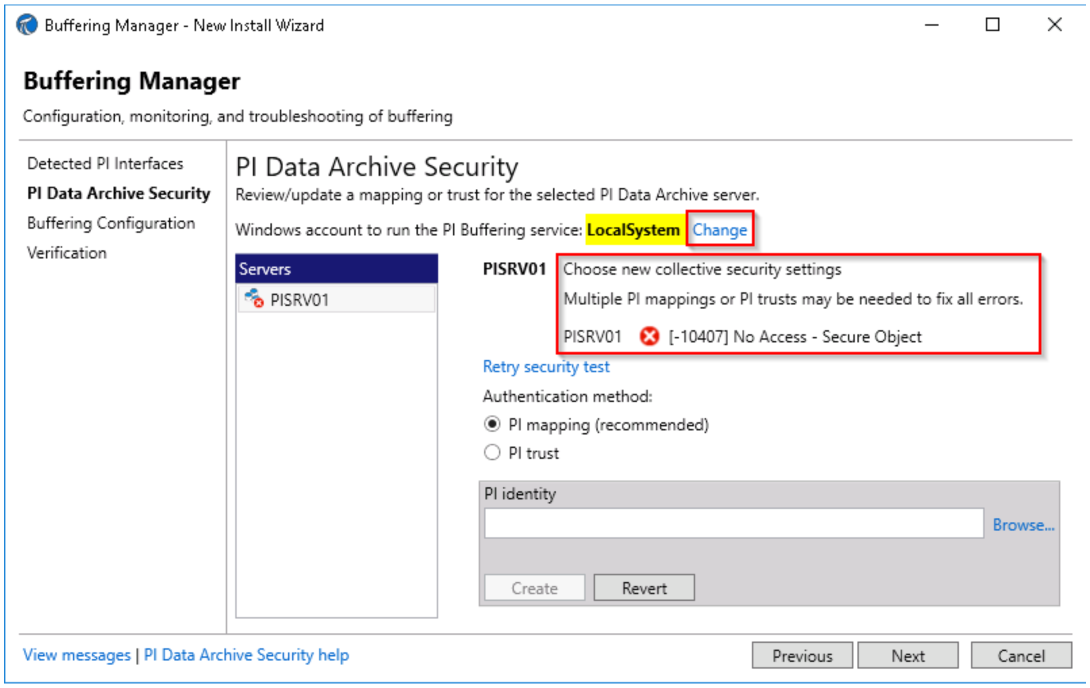
Confirm the PI interfaces and services for which you want to configure buffering.
Servers that are not selected will not be buffered.
Detected PI interfaces are listed by server.

Buffer	Server	Status
✓	PISRV01 (PISRV01)	
	PI-opcint1	Running

Don't see all your PI interfaces? Want to add another service? Select a service.

View messages | Detected PI Interfaces help

Stop at the below screen, there is some configuration required. There is no PI Mapping or PI Trust for the buffer subsystem.


Buffering Manager - New Install Wizard

Buffering Manager

Configuration, monitoring, and troubleshooting of buffering

Detected PI Interfaces
PI Data Archive Security
Buffering Configuration
Verification

PI Data Archive Security

Review/update a mapping or trust for the selected PI Data Archive server.

Windows account to run the PI Buffering service: **LocalSystem** Change

Servers
PISRV01

PISRV01 Choose new collective security settings
Multiple PI mappings or PI trusts may be needed to fix all errors.
PISRV01 [-10407] No Access - Secure Object

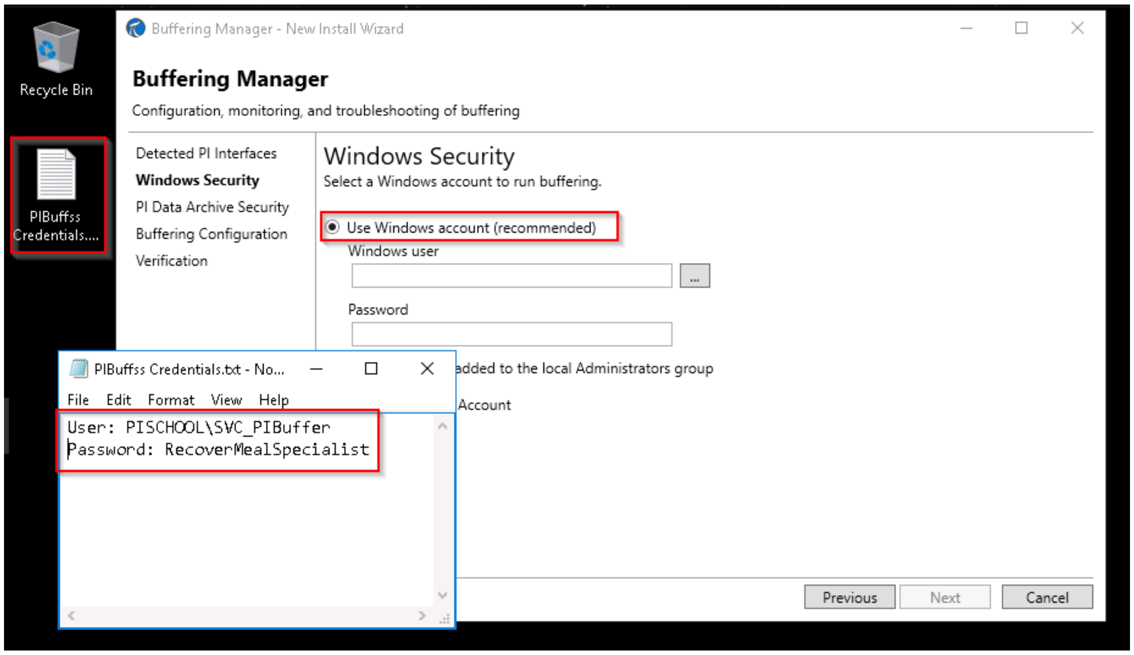
Retry security test

Authentication method:
(●) PI mapping (recommended)
( ) PI trust

PI identity

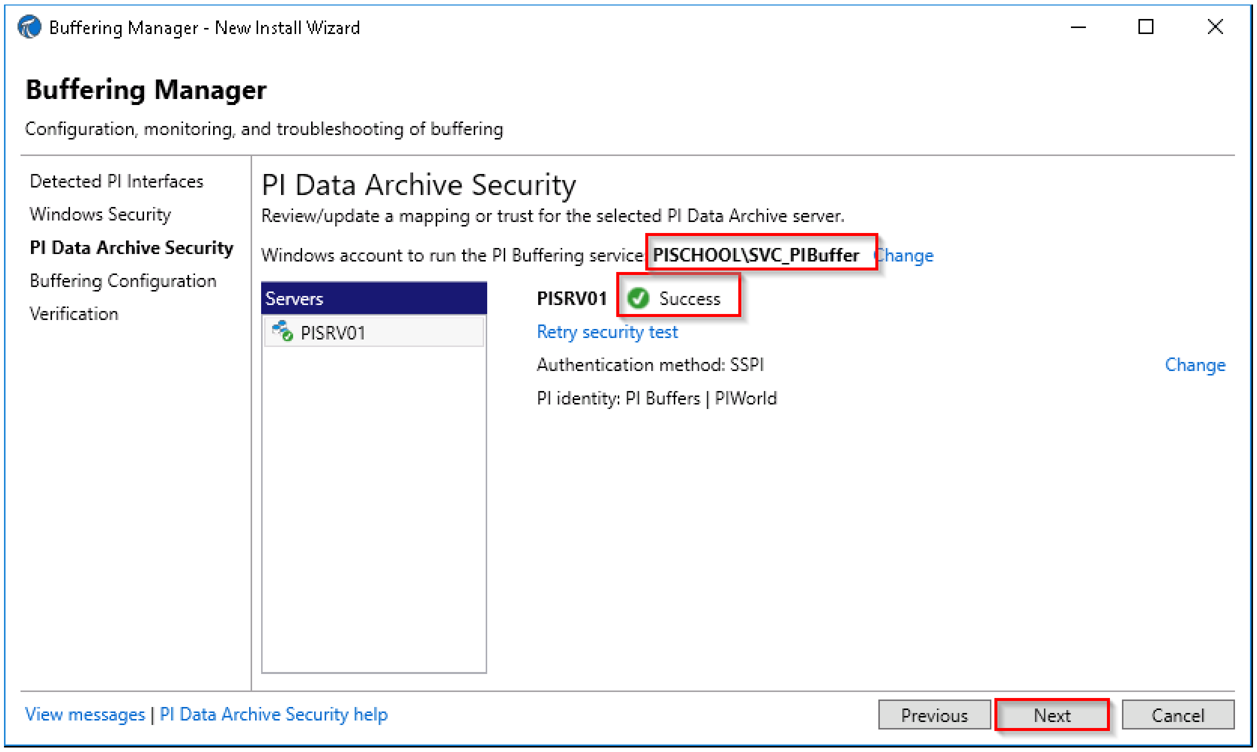
View messages | PI Data Archive Security help

We want to change the service account from LocalSystem to a domain service account. Click on the “Change” option highlighted above. Select “Windows account” and enter the credentials stored in the **PIBufss Credentials text document stored on your desktop**:


Recycle Bin
PIBuffss Credentials....
Buffering Manager - New Install Wizard
Buffering Manager
Configuration, monitoring, and troubleshooting of buffering
Detected PI Interfaces
Windows Security
PI Data Archive Security
Buffering Configuration
Verification
Windows Security
Select a Windows account to run buffering.
Use Windows account (recommended)
Windows user
Password
added to the local Administrators group
Account
Previous
Next
Cancel
PIBuffss Credentials.txt - No...
File Edit Format View Help

```
User: PISCHOOL\SVC_PIBuffer
Password: RecoverMealSpecialist
```

Once the credentials have been entered, select Next. The security will be successful at this point because a mapping has already been created for the PI buffer subsystem on PISRV01.



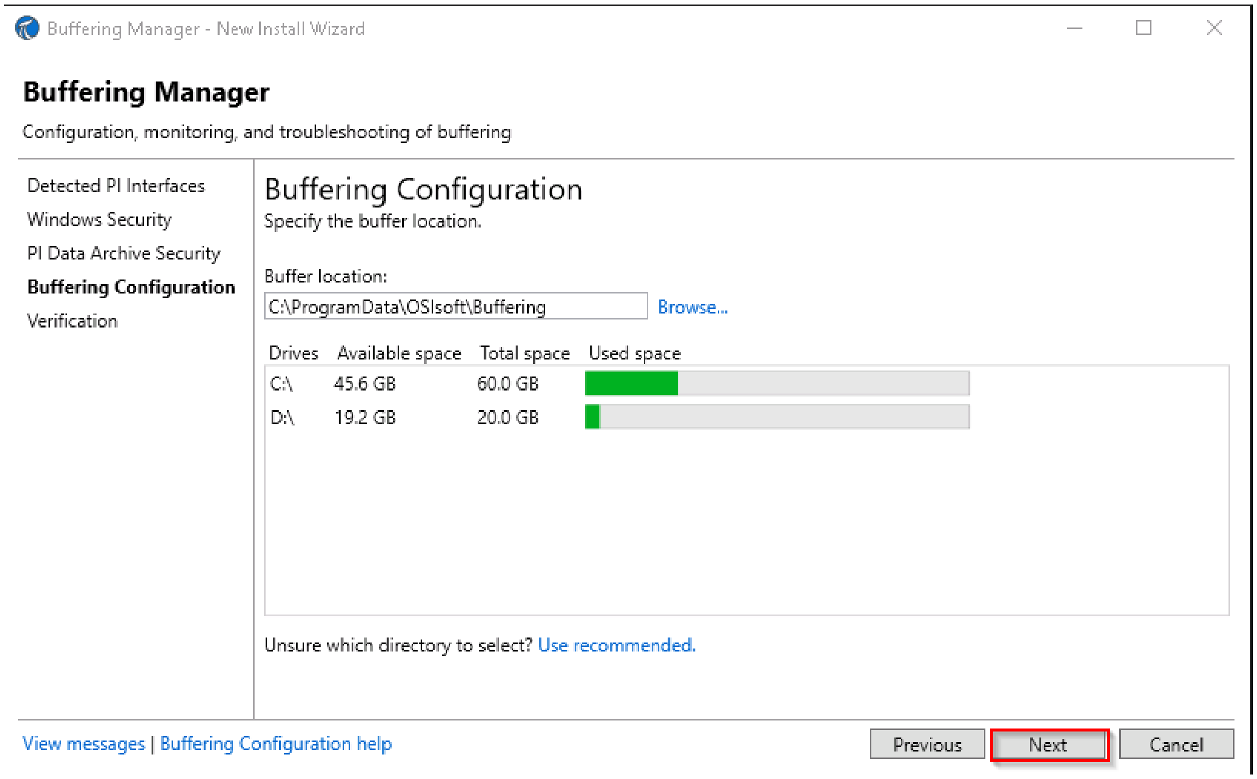
Review/update a mapping or trust for the selected PI Data Archive server.

Windows account to run the PI Buffering service: PISCHOOL\SVC_PIBuffer

PISRV01 ✓ Success

Authentication method: SSPI

PI identity: PI Buffers | PIWorld


Buffering Manager - New Install Wizard

Buffering Manager

Configuration, monitoring, and troubleshooting of buffering

Detected PI Interfaces
Windows Security
PI Data Archive Security
Buffering Configuration
Verification

Buffering Configuration

Specify the buffer location.

Buffer location:
C:\ProgramData\OSIsoft\Buffering

Drives	Available space	Total space	Used space
C:\	45.6 GB	60.0 GB	
D:\	19.2 GB	20.0 GB	

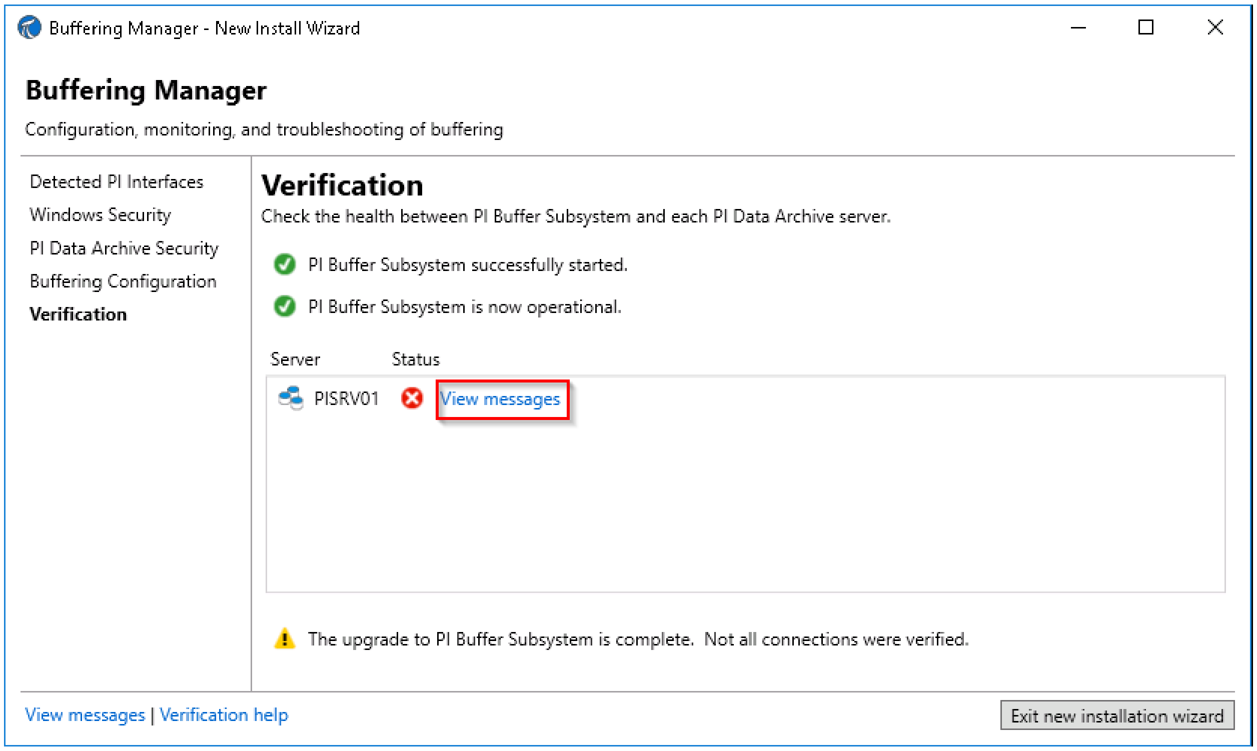
Unsure which directory to select? Use recommended.

After this, the buffer subsystem will be restarted

Tip

We won’t get into all the configuration involved in PI System Security. That’s an entire lab in itself. However, we will touch on security with a few examples. Consider watching this YouTube playlist (at a later time) to get a better understanding of PI Data Archive Security.

You'll definitely see the errors here. Click PI messages.



Buffering Manager - New Install Wizard

Buffering Manager

Configuration, monitoring, and troubleshooting of buffering

Detected PI Interfaces
Windows Security
PI Data Archive Security
Buffering Configuration
Verification

Verification

Check the health between PI Buffer Subsystem and each PI Data Archive server.

✓ PI Buffer Subsystem successfully started.

✓ PI Buffer Subsystem is now operational.

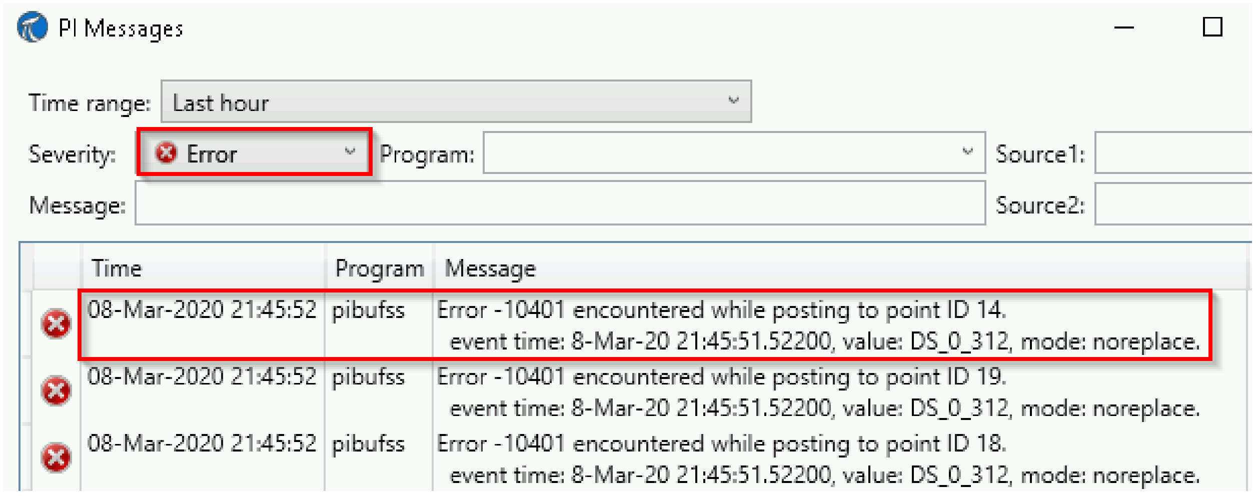
Server	Status
PISRV01	✗ View messages

⚠ The upgrade to PI Buffer Subsystem is complete. Not all connections were verified.

View messages | Verification help

Exit new installation wizard

Filter for Errors:



PI Messages

Time range: Last hour

Severity: Error

Program:

Source1:

Message:

Source2:

	Time	Program	Message
✗	08-Mar-2020 21:45:52	pibufss	Error -10401 encountered while posting to point ID 14. event time: 8-Mar-20 21:45:51.52200, value: DS_0_312, mode: noreplace.
✗	08-Mar-2020 21:45:52	pibufss	Error -10401 encountered while posting to point ID 19. event time: 8-Mar-20 21:45:51.52200, value: DS_0_312, mode: noreplace.
✗	08-Mar-2020 21:45:52	pibufss	Error -10401 encountered while posting to point ID 18. event time: 8-Mar-20 21:45:51.52200, value: DS_0_312, mode: noreplace.

d. Troubleshoot Errors

Error -10401 is not giving much detail, but we can at least look up point ID 14 later on to see which tag is having a problem.

If we google "pi error 10401" and do a little digging, we can deduce that the full error is **[-10401] No write access – secure object**.

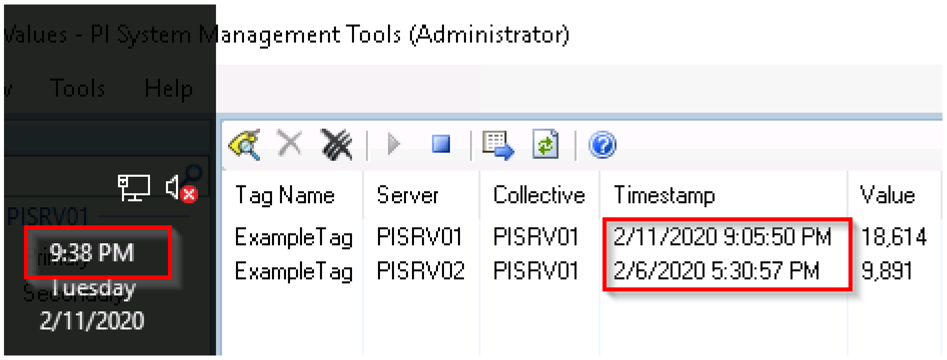
You could search for the error codes on the customer portal to get articles on our knowledge base with details about this error and how to resolve it. You will need to sign up for an OSIsoft Single sign on (SSO) account. The required SSO account is free and highly recommended but we don't want to spend any time on this during the lab.

As for the other errors, looks like we maybe didn't have permission to restart the OPC Interface service through Buffering Manager. Let's see if they persist after we address the -10401 errors and restart PI Buffer Subsystem.

We've deduced that something doesn't have write access to a tag with pointid 14. It stands to reason that it's probably PI Buffer Subsystem, since the tag was working before we configured buffering.

Which tag is pointid 14? We can find out by doing a tag search, and since we anticipate the possibility of tag configuration changes, let's do it from the **Point Builder plugin in PI SMT**.

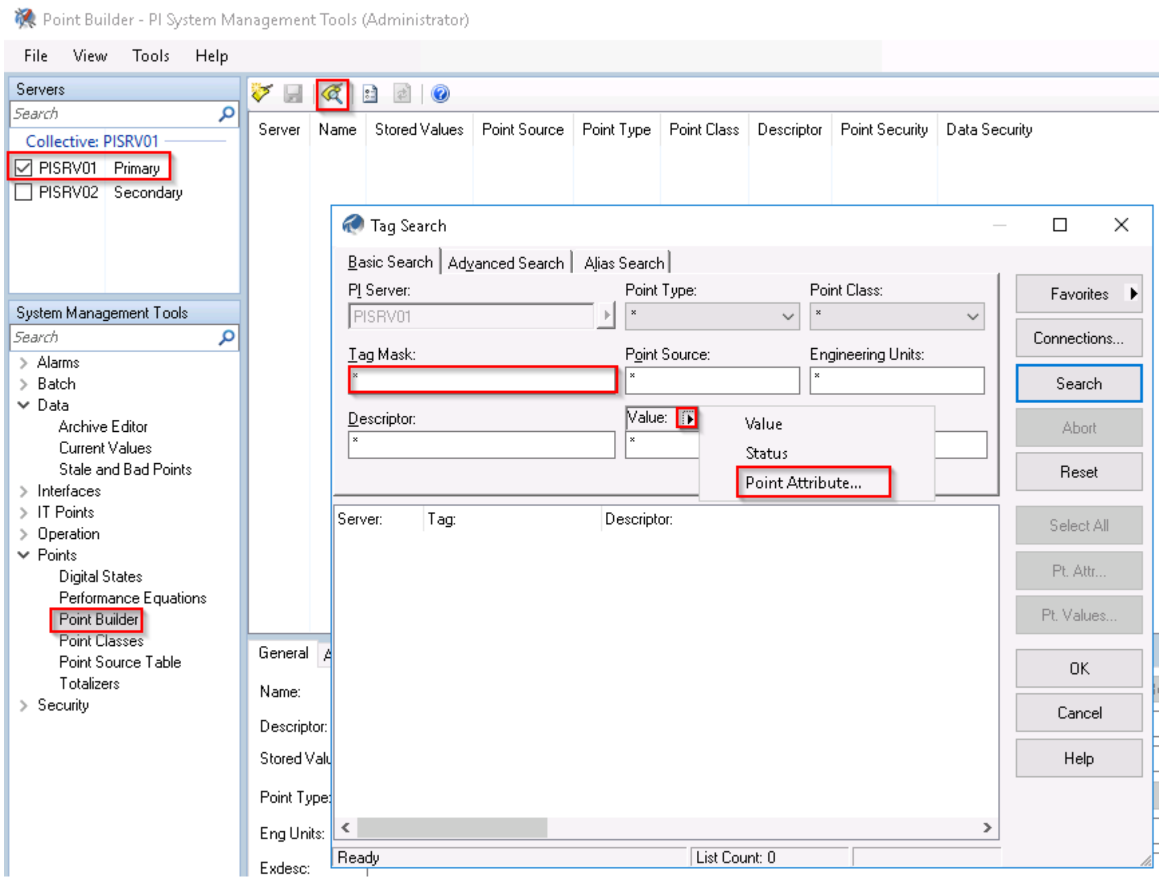
From **PICLIENT01**, Open **PI SMT > Data > Current values**. Verify the values of ExampleTag on PISRV01 collective and confirm that both are not getting new data by comparing the system timestamp with the timestamp of the last value.



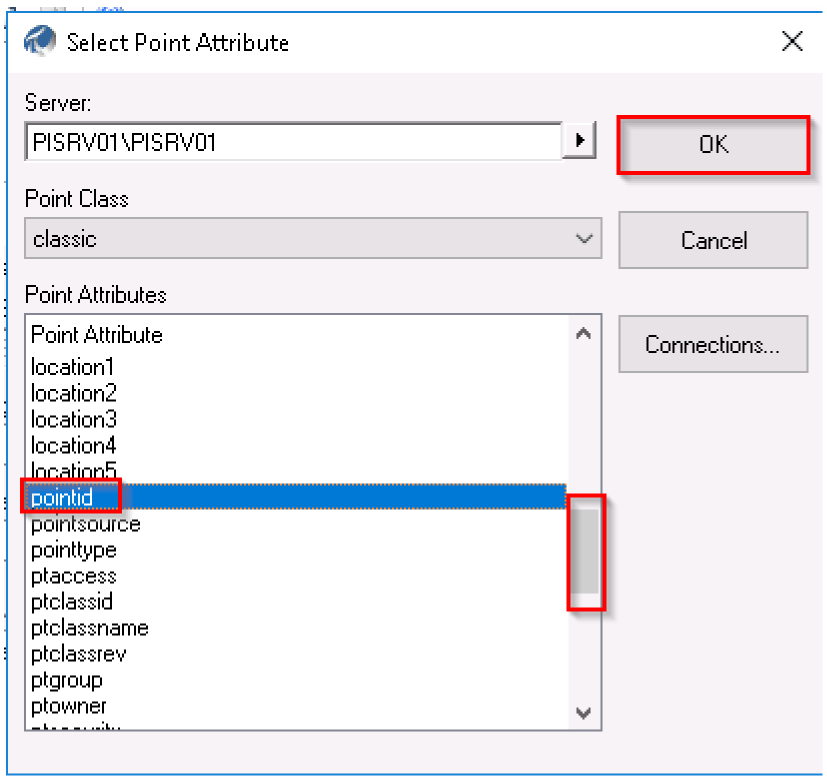
Tag Name	Server	Collective	Timestamp	Value
ExampleTag	PISRV01	PISRV01	2/11/2020 9:05:50 PM	18,614
ExampleTag	PISRV02	PISRV01	2/6/2020 5:30:57 PM	9,891

Now, switch to Point Builder in SMT to update the security (**PI SMT > Data > Point Builder).**

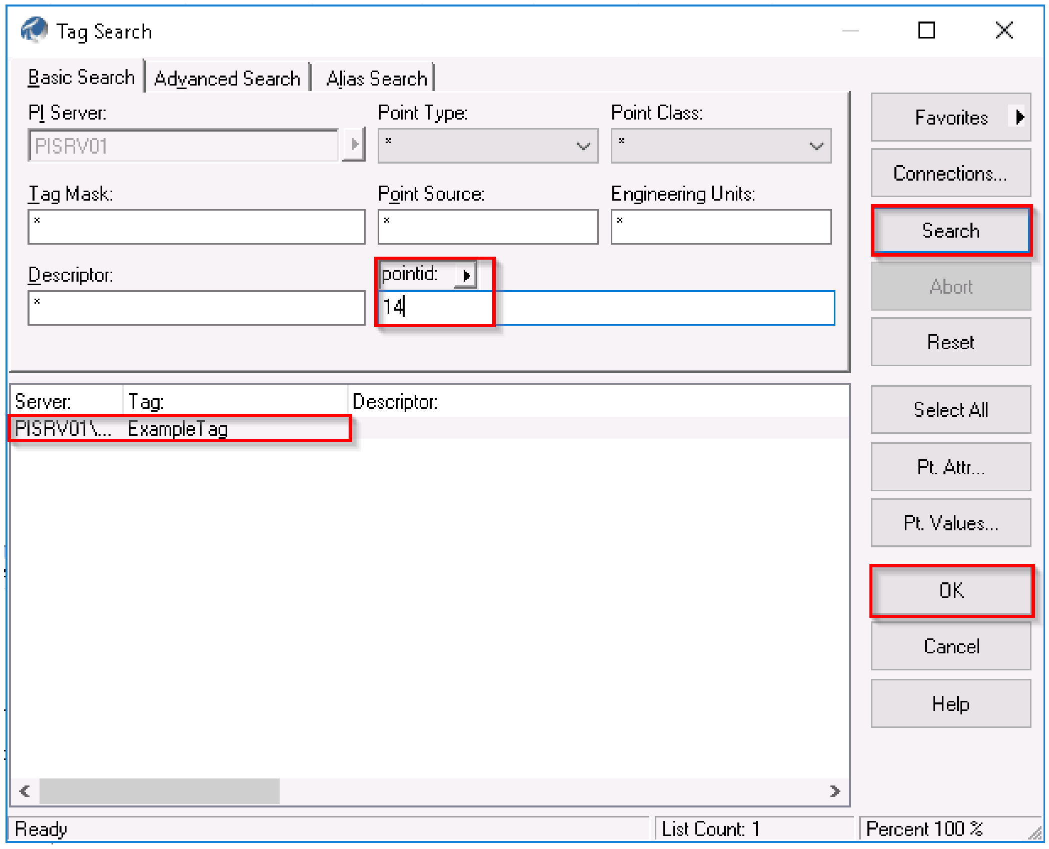
Do a tag search, ensure * is set for the Tag Mask, and change the customizable field to Point Attribute…



Scroll down to pointid, select pointid, click OK.



Search for pointid 14, select tag ExampleTag, click OK



PI Server: PISRV01

Point Type: *

Point Class: *

Tag Mask: *

Point Source: *

Engineering Units: *

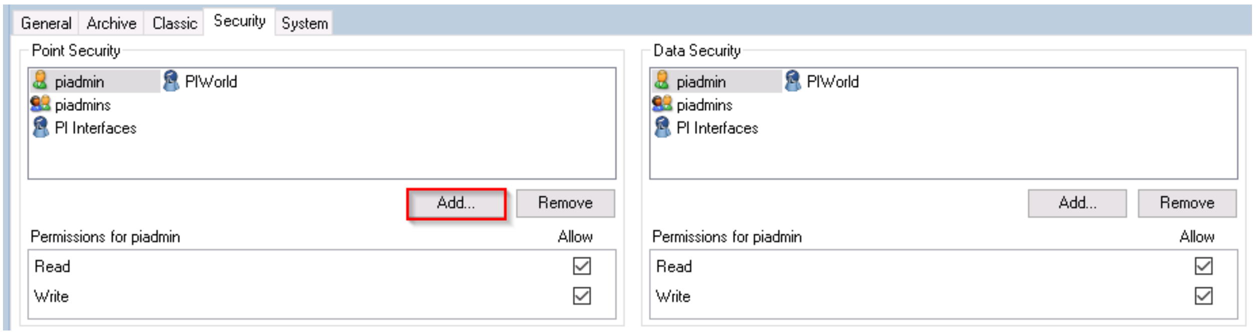
Descriptor: *

pointid: 14

Server:	Tag:	Descriptor:
PISRV01\...	ExampleTag	

Ready | List Count: 1 | Percent 100 %

Ensure ExampleTag is selected. Go to the Security tab, we can see that the PI Buffers identity is not included here. Let's add it under Point Security.



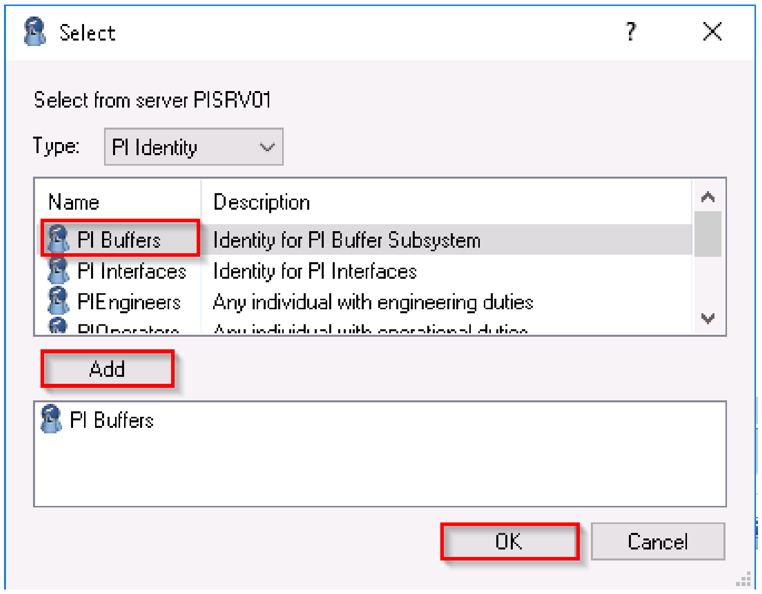
Point Security: piadmin, PIWorld, piadmins, PI Interfaces

Permissions for piadmin: Read ✓, Write ✓

Data Security: piadmin, PIWorld, piadmins, PI Interfaces

Permissions for piadmin: Read ✓, Write ✓

Click **PI Buffers> Add…> OK**


Select

Select from server PISRV01

Type: PI Identity

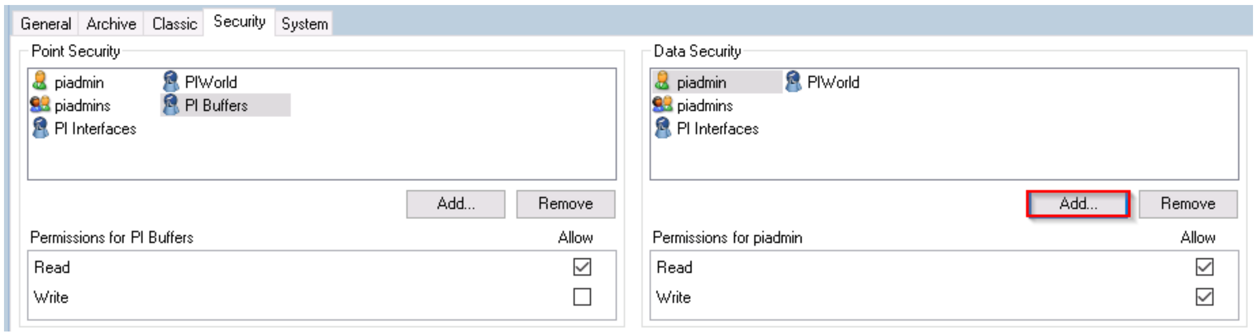
Name	Description
PI Buffers	Identity for PI Buffer Subsystem
PI Interfaces	Identity for PI Interfaces
PIEngineers	Any individual with engineering duties
PIOperators	Any individual with operational duties

Add

PI Buffers

OK Cancel

Also add PI Buffers to data security.



Point Security: piadmin, piadmins, PI Interfaces, PIWorld, PI Buffers

Add... Remove

Permissions for PI Buffers

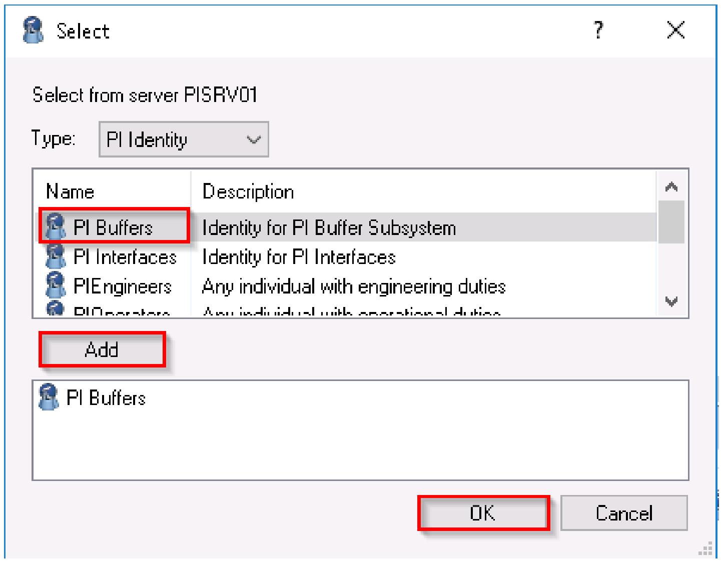
	Allow
Read	☑
Write	☐

Data Security: piadmin, piadmins, PI Interfaces, PIWorld

Add... Remove

Permissions for piadmin

	Allow
Read	☑
Write	☑


Select

Select from server PISRV01

Type: PI Identity

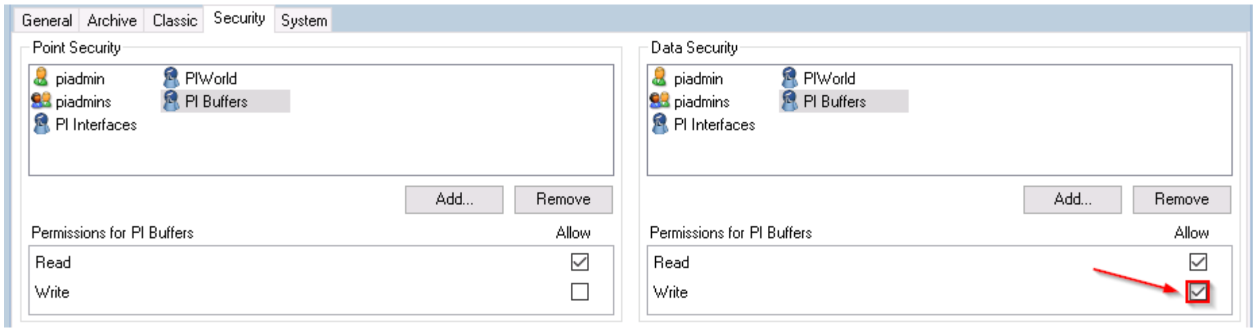
Name	Description
PI Buffers	Identity for PI Buffer Subsystem
PI Interfaces	Identity for PI Interfaces
PIEngineers	Any individual with engineering duties
PIOperators	Any individual with operational duties

Add

PI Buffers

OK Cancel

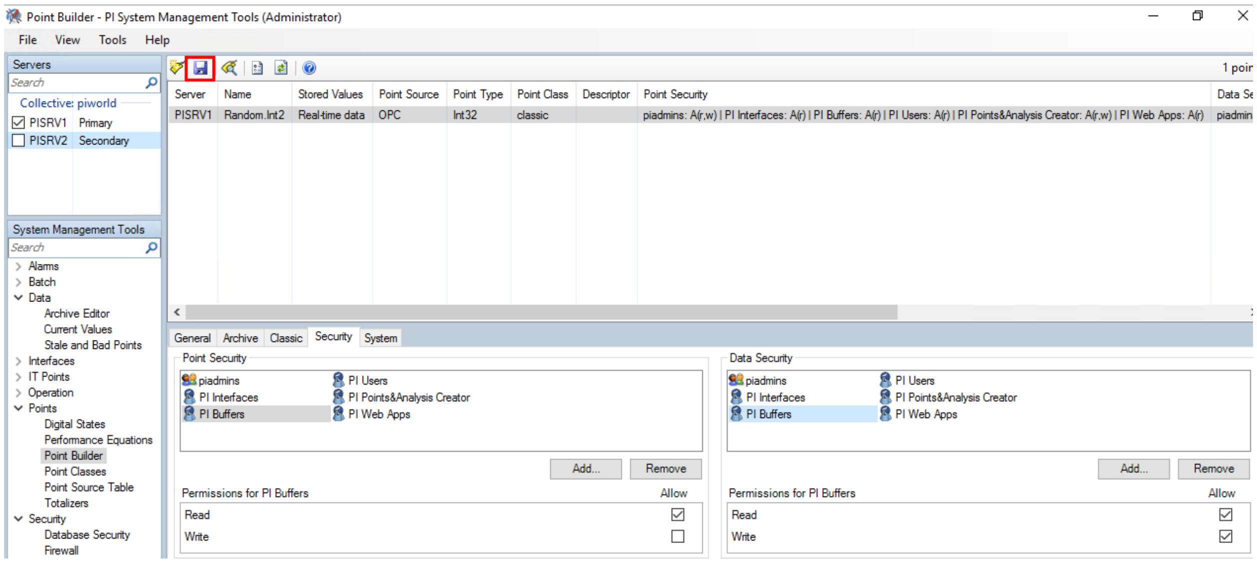
Write permission is required for Data Security.



In simple terms, **Point Security relates to the ability to search for, view, and edit the configuration of tags**. For example, Write in Point Security was required to change the instrumenttag of BrokenTag and is required here to edit tag security settings.

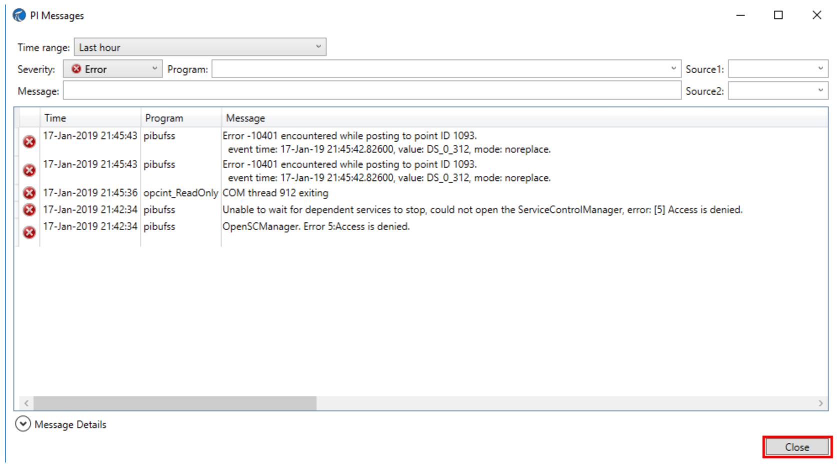
Data Security controls which PI Identities can read data from or write data to tags.

Be sure to Save!



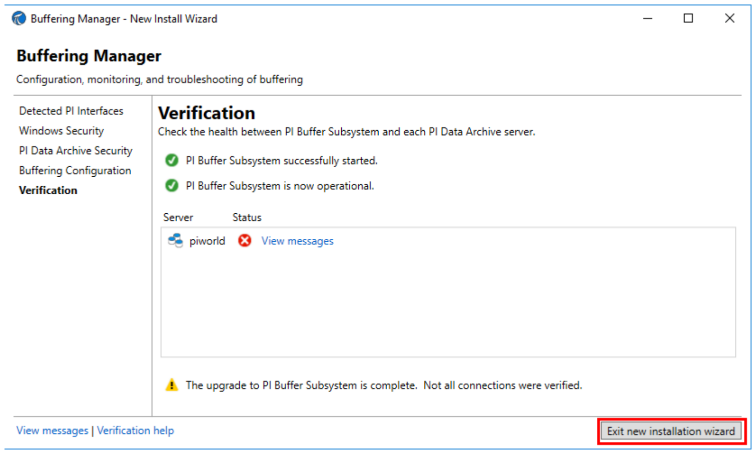
Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor	Point Security	Data Se
PISRV1	Random.Int2	Real-time data	OPC	Int32	classic		piadmins: A(r,w) | PI Interfaces: A(r) | PI Buffers: A(r) | PI Users: A(r) | PI Points&Analysis Creator: A(r,w) | PI Web Apps: A(r)	piadmin

Let’s **go back to PIINT01**. Close the message window.



Time	Program	Message
17-Jan-2019 21:45:43	pibufss	Error -10401 encountered while posting to point ID 1093. event time: 17-Jan-19 21:45:42.82600, value: DS_0_312, mode: noreplace.
17-Jan-2019 21:45:43	pibufss	Error -10401 encountered while posting to point ID 1093. event time: 17-Jan-19 21:45:42.82600, value: DS_0_312, mode: noreplace.
17-Jan-2019 21:45:36	opcint_ReadOnly	COM thread 912 exiting
17-Jan-2019 21:42:34	pibufss	Unable to wait for dependent services to stop, could not open the ServiceControlManager, error: [5] Access is denied.
17-Jan-2019 21:42:34	pibufss	OpenSCManager. Error 5:Access is denied.

Exit new installation wizard if you haven't already.


Buffering Manager - New Install Wizard
Buffering Manager
Configuration, monitoring, and troubleshooting of buffering
Detected PI Interfaces
Windows Security
PI Data Archive Security
Buffering Configuration
Verification
View messages | Verification help
Exit new installation wizard

Verification

Check the health between PI Buffer Subsystem and each PI Data Archive server.

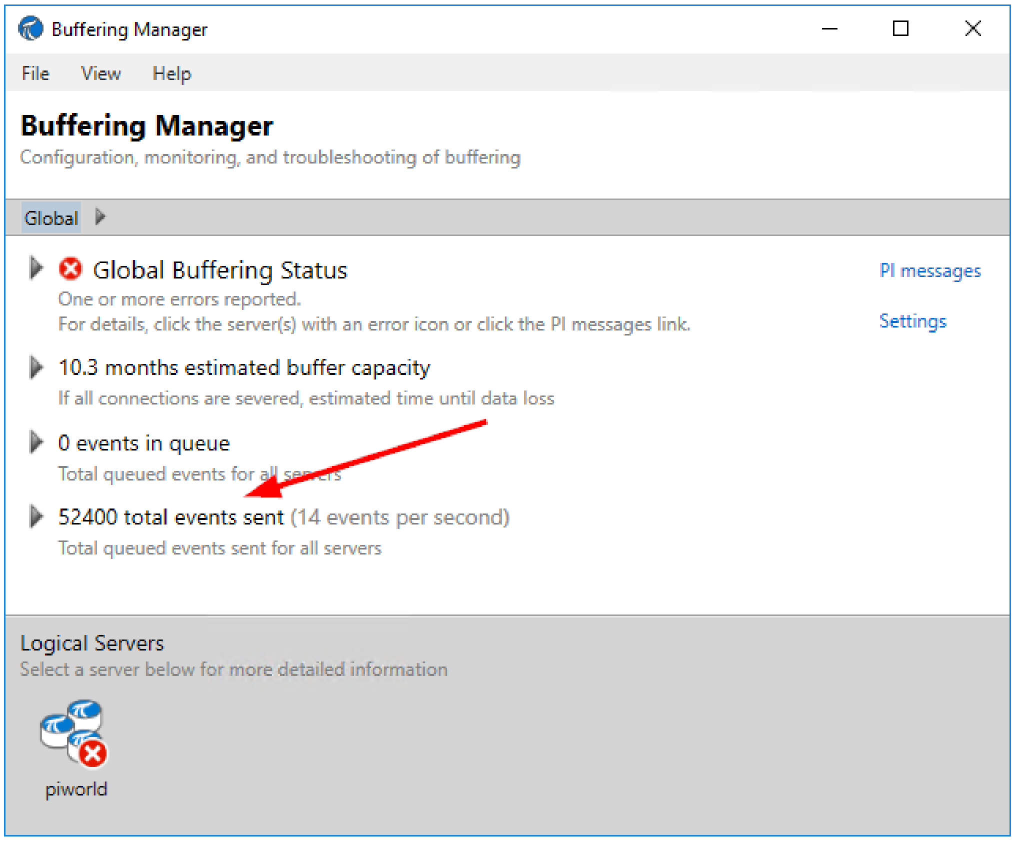
PI Buffer Subsystem successfully started.

PI Buffer Subsystem is now operational.

Server	Status
piworld	View messages

The upgrade to PI Buffer Subsystem is complete. Not all connections were verified.

Depending on timing however it may still be red. Keep reading.



Buffering Manager

File View Help

Buffering Manager

Configuration, monitoring, and troubleshooting of buffering

Global

Global Buffering Status

One or more errors reported.

For details, click the server(s) with an error icon or click the PI messages link.

PI messages

Settings

10.3 months estimated buffer capacity

If all connections are severed, estimated time until data loss

0 events in queue

Total queued events for all servers

52400 total events sent (14 events per second)

Total queued events sent for all servers

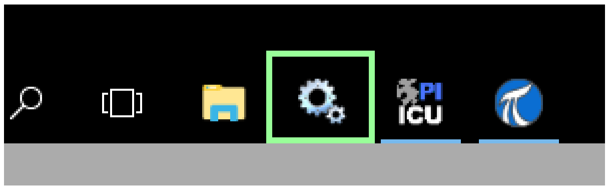
Logical Servers

Select a server below for more detailed information

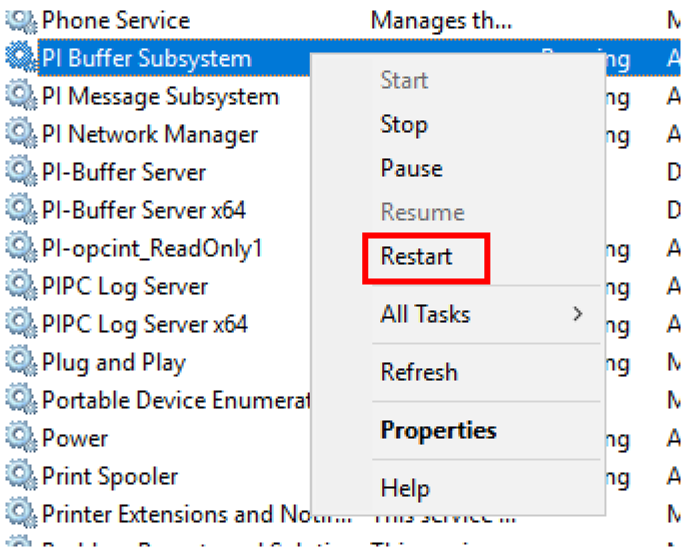
piworld

If it's still showing errors. Restart the PI Buffer Subsystem service and the PI Interface Service and see if that fixes it.

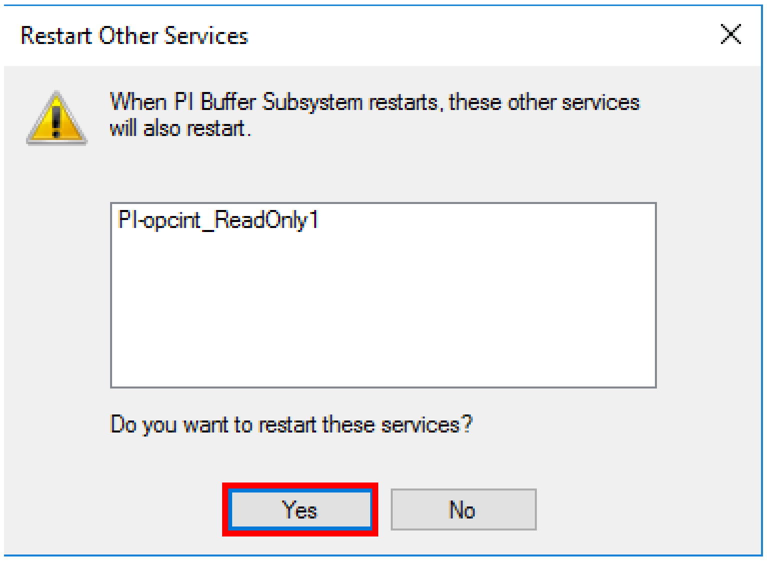
Open windows services from the taskbar.



Restart PI Buffer Subsystem



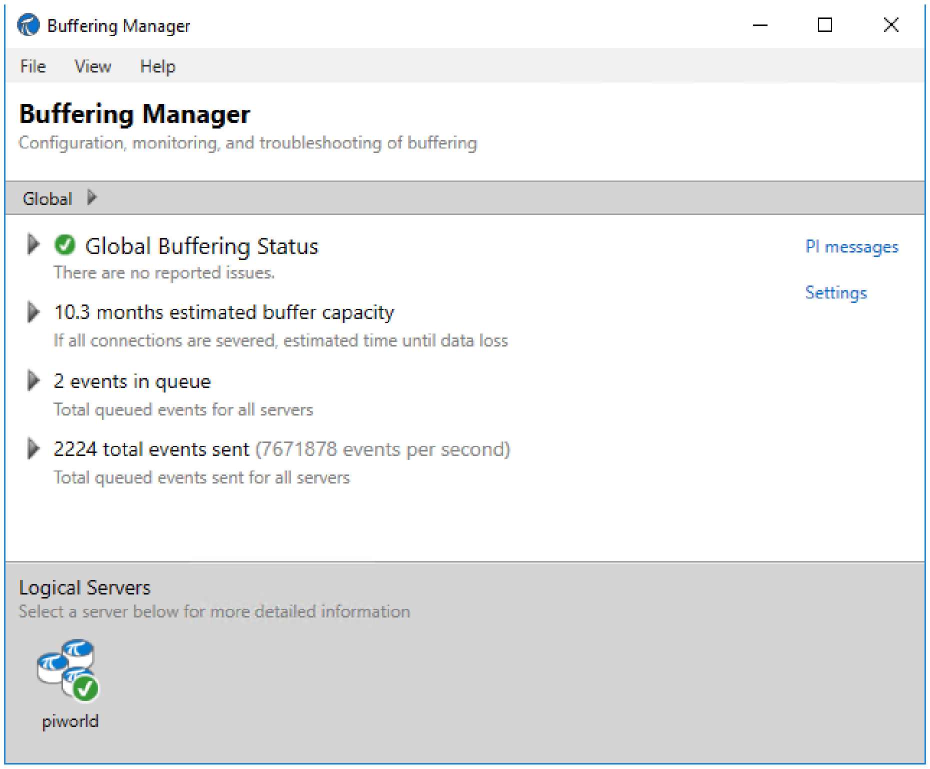
The wizard secretly added a dependency on PI Buffer Subsystem to the OPC Interface service, so it must restart too.


Restart Other Services

When PI Buffer Subsystem restarts, these other services will also restart.

PI-opcint_ReadOnly1

Do you want to restart these services?



Buffering Manager

Configuration, monitoring, and troubleshooting of buffering

Global

Global Buffering Status

There are no reported issues.

10.3 months estimated buffer capacity

If all connections are severed, estimated time until data loss

2 events in queue

Total queued events for all servers

2224 total events sent (7671878 events per second)

Total queued events sent for all servers

PI messages

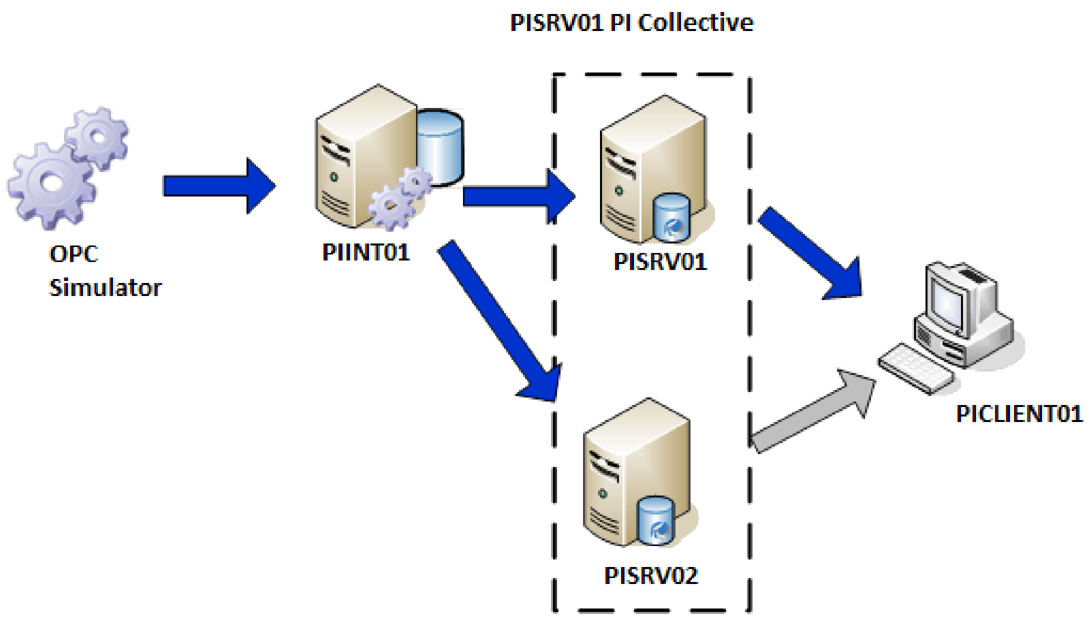
Settings

Logical Servers

Select a server below for more detailed information

piworld

Now that Buffering is configured, the OPC Interface is sending new data to PISRV1 and PISRV2.

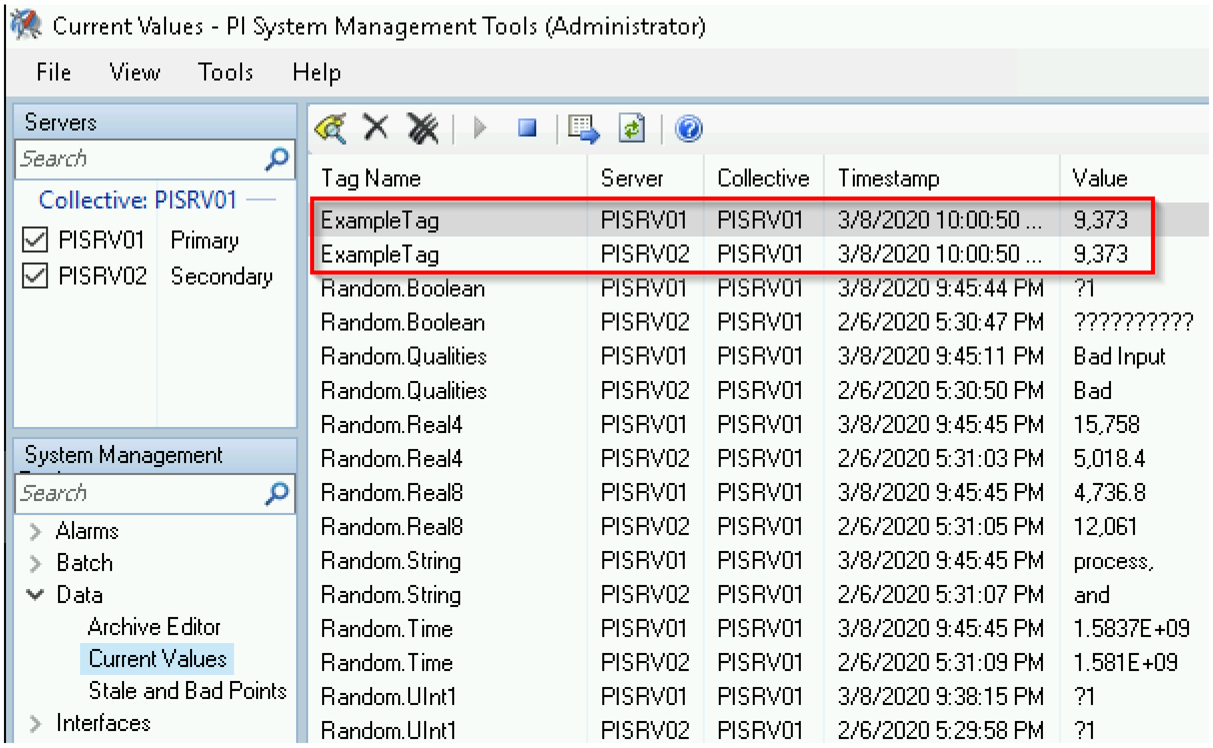


e. Verify Correct Operation

On **PICLIENT01**, Go to PI SMT, connect to **PISRV1 and PISRV2**, and add **all** OPC tags to the **Current Values plugin**. We should see identical timestamps and values on both servers now.

Note: Point security gets replicated across the collective (changes can only be made on the primary server).

Everyone's values will be different and will not match the workbook because this is randomly generated data!

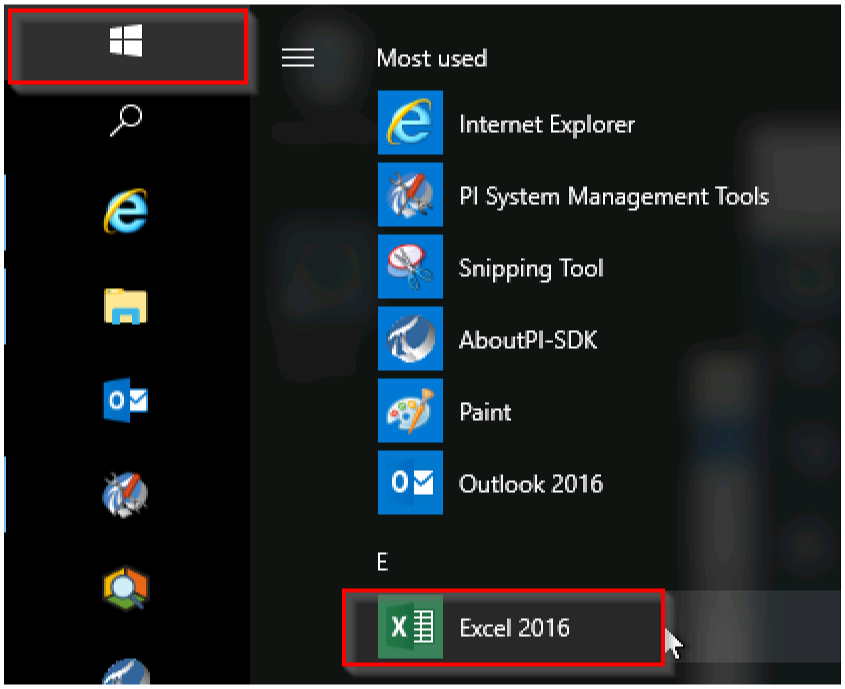


PISRV01	Primary
PISRV02	Secondary

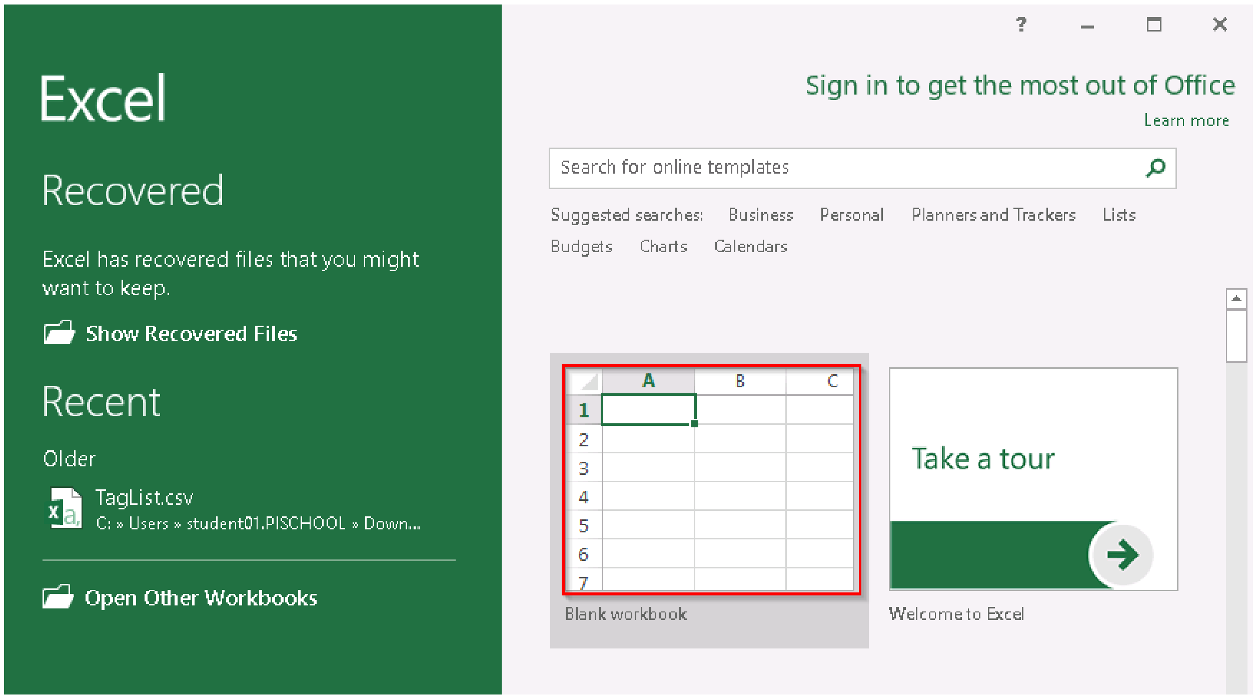
Tag Name	Server	Collective	Timestamp	Value
ExampleTag	PISRV01	PISRV01	3/8/2020 10:00:50 ...	9,373
ExampleTag	PISRV02	PISRV01	3/8/2020 10:00:50 ...	9,373
Random.Boolean	PISRV01	PISRV01	3/8/2020 9:45:44 PM	?1
Random.Boolean	PISRV02	PISRV01	2/6/2020 5:30:47 PM	??????????
Random.Qualities	PISRV01	PISRV01	3/8/2020 9:45:11 PM	Bad Input
Random.Qualities	PISRV02	PISRV01	2/6/2020 5:30:50 PM	Bad
Random.Real4	PISRV01	PISRV01	3/8/2020 9:45:45 PM	15,758
Random.Real4	PISRV02	PISRV01	2/6/2020 5:31:03 PM	5,018.4
Random.Real8	PISRV01	PISRV01	3/8/2020 9:45:45 PM	4,736.8
Random.Real8	PISRV02	PISRV01	2/6/2020 5:31:05 PM	12,061
Random.String	PISRV01	PISRV01	3/8/2020 9:45:45 PM	process,
Random.String	PISRV02	PISRV01	2/6/2020 5:31:07 PM	and
Random.Time	PISRV01	PISRV01	3/8/2020 9:45:45 PM	1.5837E+09
Random.Time	PISRV02	PISRV01	2/6/2020 5:31:09 PM	1.581E+09
Random.UInt1	PISRV01	PISRV01	3/8/2020 9:38:15 PM	?1
Random.UInt1	PISRV02	PISRV01	2/6/2020 5:29:58 PM	?1

ExampleTag is the only tag with current values. The remaining tags will not update until we add the PI Buffer identity to their point and data security settings. This can easily be done using OSIsoft's PI Builder tool which is an excel add-in used to make bulk changes to the PI Server.

From **PICLIENT01**, open Microsoft excel:

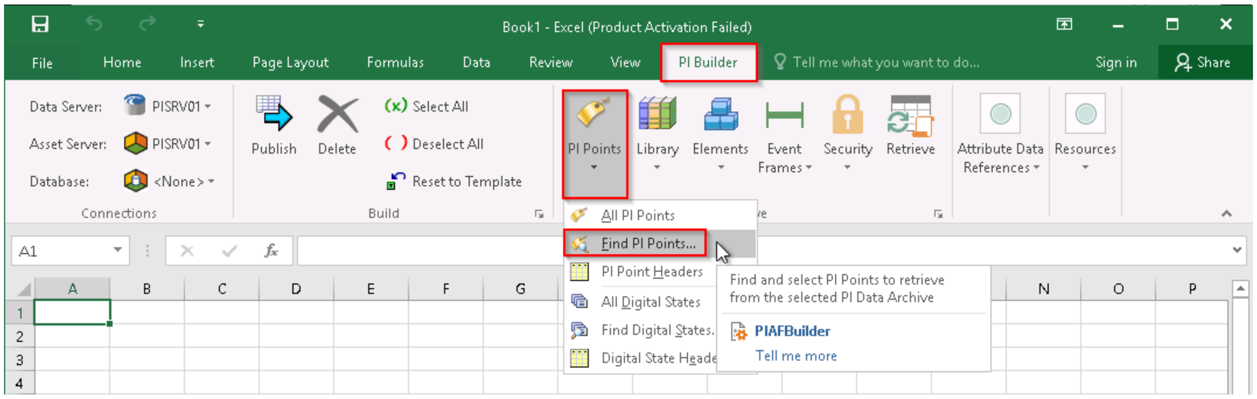


Select a blank workbook:

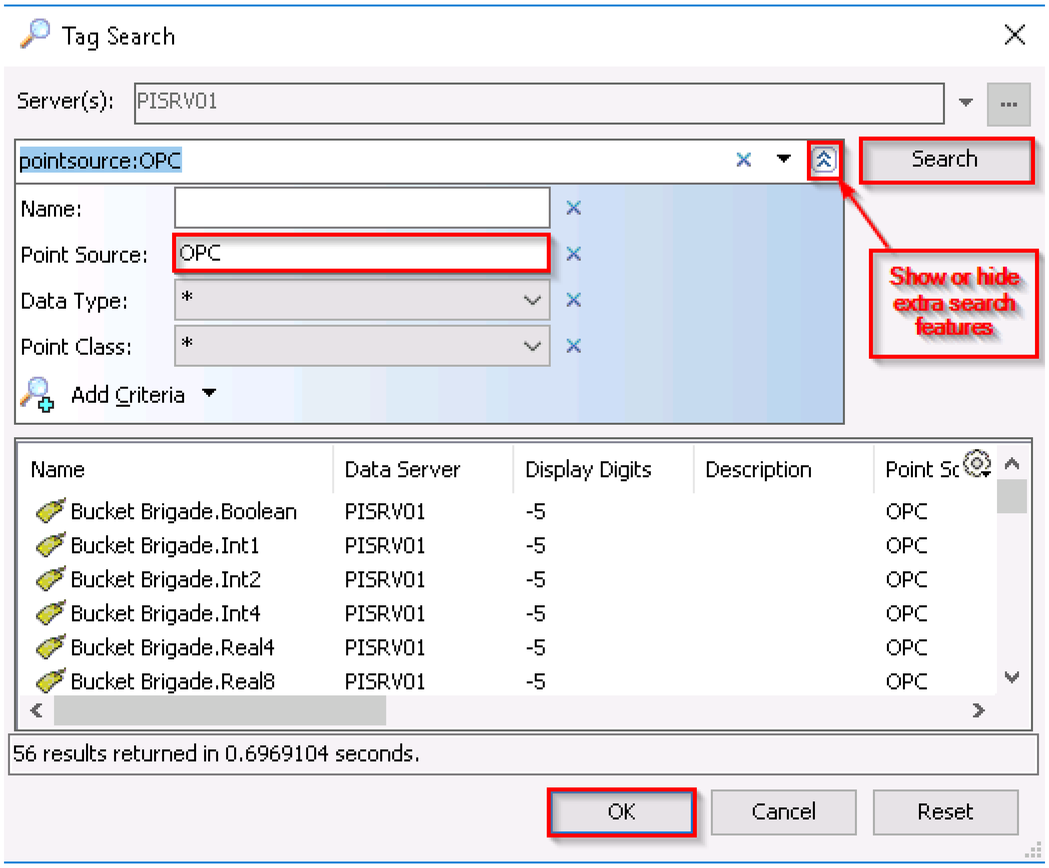


Excel
Recovered
Excel has recovered files that you might want to keep.
Show Recovered Files
Recent
Older
TagList.csv
C: » Users » student01.PISCHOOL » Down...
Open Other Workbooks
Sign in to get the most out of Office
Learn more
Search for online templates
Suggested searches: Business Personal Planners and Trackers Lists
Budgets Charts Calendars
Blank workbook
Take a tour
Welcome to Excel

Go to the **PI Builder Ribbon > PI Points > Find PI Points…**



A Tag Search window will pop up. Expand the “show or hide extra search features” double arrow to filter the search criteria. Under Point Source, enter “OPC” > Search > OK:



Server(s): PISRV01

pointsource:OPC

Name:

Point Source: OPC

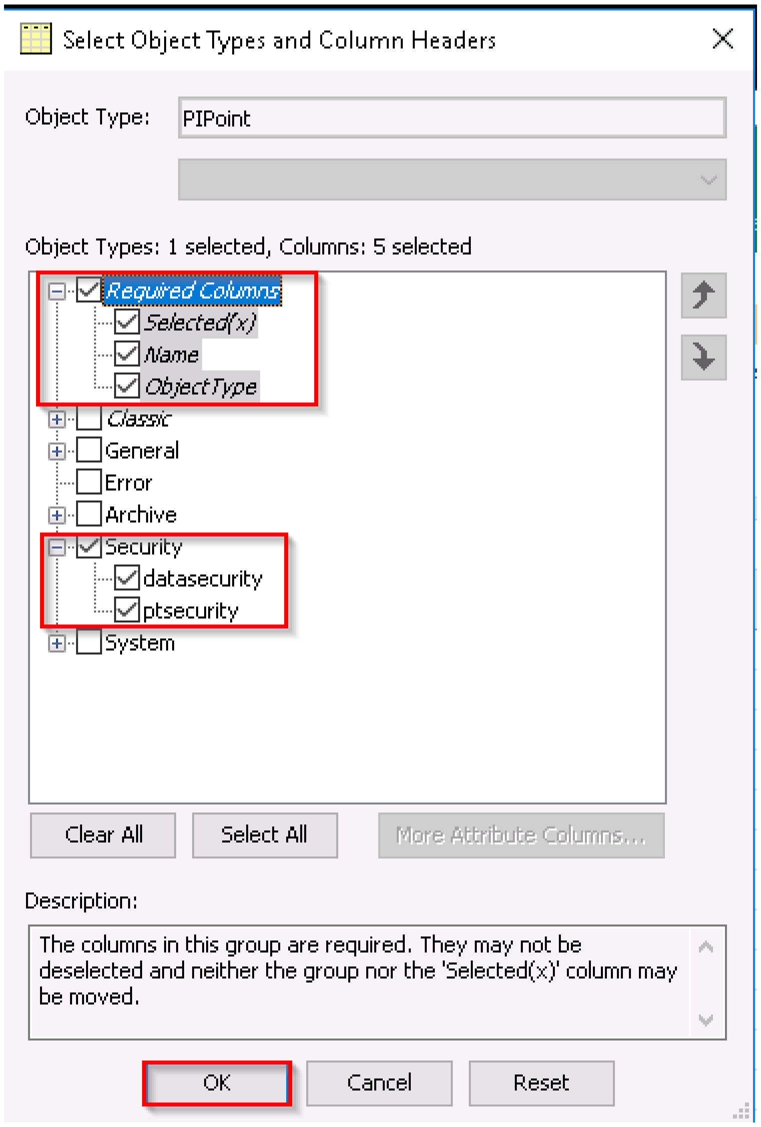
Data Type: *

Point Class: *

Name	Data Server	Display Digits	Description	Point Sc
Bucket Brigade.Boolean	PISRV01	-5		OPC
Bucket Brigade.Int1	PISRV01	-5		OPC
Bucket Brigade.Int2	PISRV01	-5		OPC
Bucket Brigade.Int4	PISRV01	-5		OPC
Bucket Brigade.Real4	PISRV01	-5		OPC
Bucket Brigade.Real8	PISRV01	-5		OPC

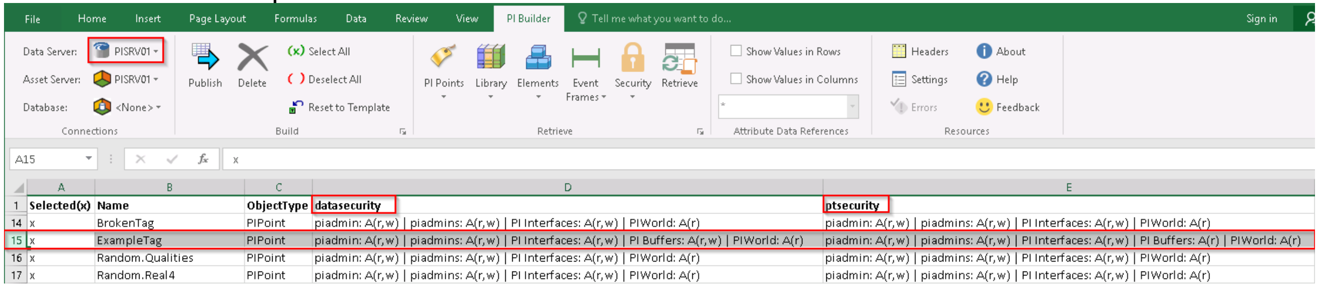
56 results returned in 0.6969104 seconds.

You will get another window to select Object Types and Column Headers. Only check the following then click OK:



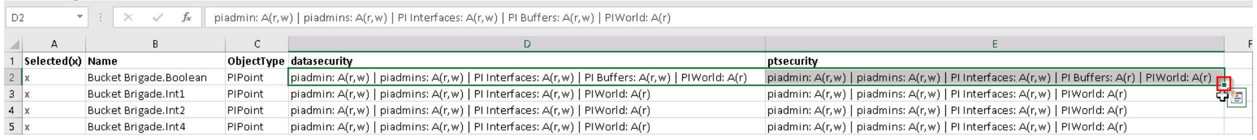
The columns in this group are required. They may not be deselected and neither the group nor the 'Selected(x)' column may be moved.

Notice the difference between the security strings for ExampleTag compared to the rest of the PI Points with point source OPC.



	A	B	C	D	E
1	Selected(x)	Name	ObjectType	datasecurity	ptsecurity
14	x	BrokenTag	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)
15	x	ExampleTag	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r) | PIWorld: A(r)
16	x	Random.Qualities	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)
17	x	Random.Real4	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)

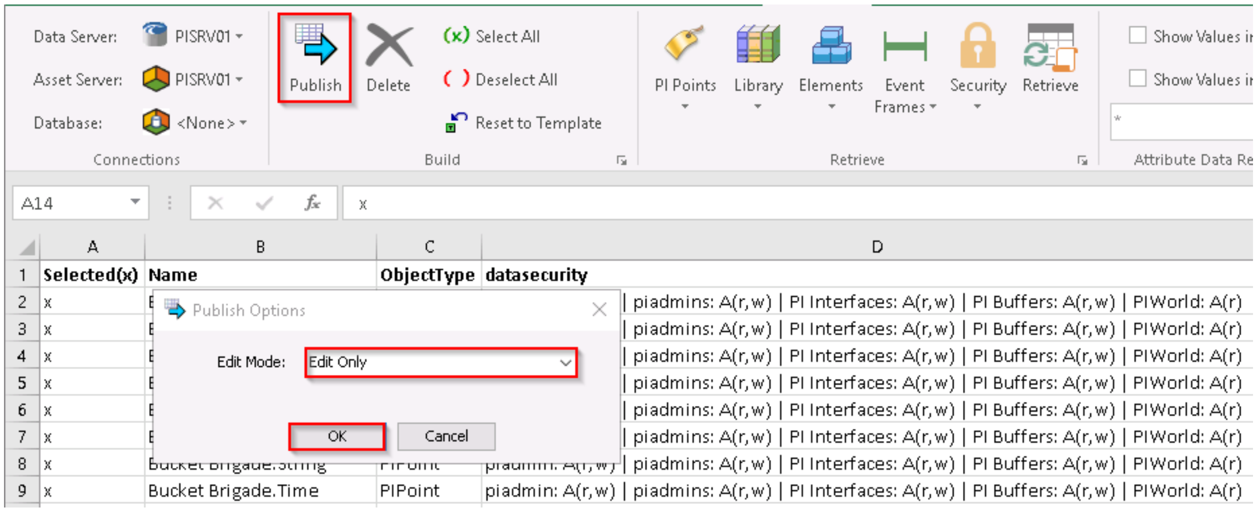
ExampleTag has “PI Buffer: A(r,w)” under datasecurity and “PI Buffer: A(r)” under ptsecurity. This is because we had added the PI Buffer identity to the PI Point security and gave the identity read permissions on point security as well as read/write permissions on data security. Copy this security string to the rest of the tags. The easiest way to do this would be to copy the datasecurity and ptsecurity columns of ExampleTag to the PI Point in row 2, then select both datasecurity and ptsecurity columns of row 2 and double click on the square at the base of the highlighted columns to propagate the change to the rest of the columns below it:



D2 — piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)

	A	B	C	D	E
1	Selected(x)	Name	ObjectType	datasecurity	ptsecurity
2	x	Bucket Brigade.Boolean	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r) | PIWorld: A(r)
3	x	Bucket Brigade.Int1	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)
4	x	Bucket Brigade.Int2	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)
5	x	Bucket Brigade.Int4	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PIWorld: A(r)

Once all the rows have the same datasecurity and ptsecurity, go to the top pane and select Publish > Publish options, select Edit Only > OK:



A14 — x

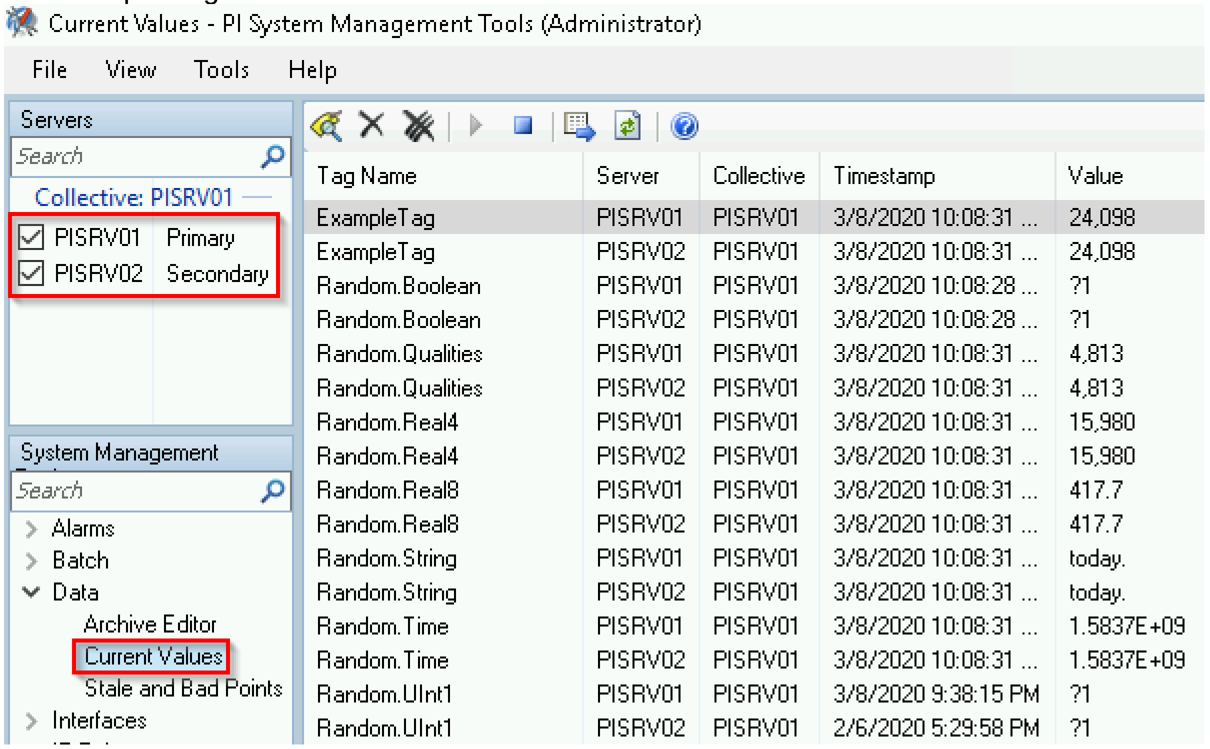
Publish Options

Edit Mode: Edit Only

OK Cancel

	A	B	C	D
1	Selected(x)	Name	ObjectType	datasecurity
2	x			| piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
3	x			| piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
4	x			| piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
5	x			| piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
6	x			| piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
7	x			| piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
8	x	Bucket Brigade.String	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)
9	x	Bucket Brigade.Time	PIPoint	piadmin: A(r,w) | piadmins: A(r,w) | PI Interfaces: A(r,w) | PI Buffers: A(r,w) | PIWorld: A(r)

Go back to PI SMT's Current Values plugin. All PI points with point source "OPC" will now be updating:



Tag Name	Server	Collective	Timestamp	Value
ExampleTag	PISRV01	PISRV01	3/8/2020 10:08:31 ...	24,098
ExampleTag	PISRV02	PISRV01	3/8/2020 10:08:31 ...	24,098
Random.Boolean	PISRV01	PISRV01	3/8/2020 10:08:28 ...	?1
Random.Boolean	PISRV02	PISRV01	3/8/2020 10:08:28 ...	?1
Random.Qualities	PISRV01	PISRV01	3/8/2020 10:08:31 ...	4,813
Random.Qualities	PISRV02	PISRV01	3/8/2020 10:08:31 ...	4,813
Random.Real4	PISRV01	PISRV01	3/8/2020 10:08:31 ...	15,980
Random.Real4	PISRV02	PISRV01	3/8/2020 10:08:31 ...	15,980
Random.Real8	PISRV01	PISRV01	3/8/2020 10:08:31 ...	417.7
Random.Real8	PISRV02	PISRV01	3/8/2020 10:08:31 ...	417.7
Random.String	PISRV01	PISRV01	3/8/2020 10:08:31 ...	today.
Random.String	PISRV02	PISRV01	3/8/2020 10:08:31 ...	today.
Random.Time	PISRV01	PISRV01	3/8/2020 10:08:31 ...	1.5837E+09
Random.Time	PISRV02	PISRV01	3/8/2020 10:08:31 ...	1.5837E+09
Random.UInt1	PISRV01	PISRV01	3/8/2020 9:38:15 PM	?1
Random.UInt1	PISRV02	PISRV01	2/6/2020 5:29:58 PM	?1

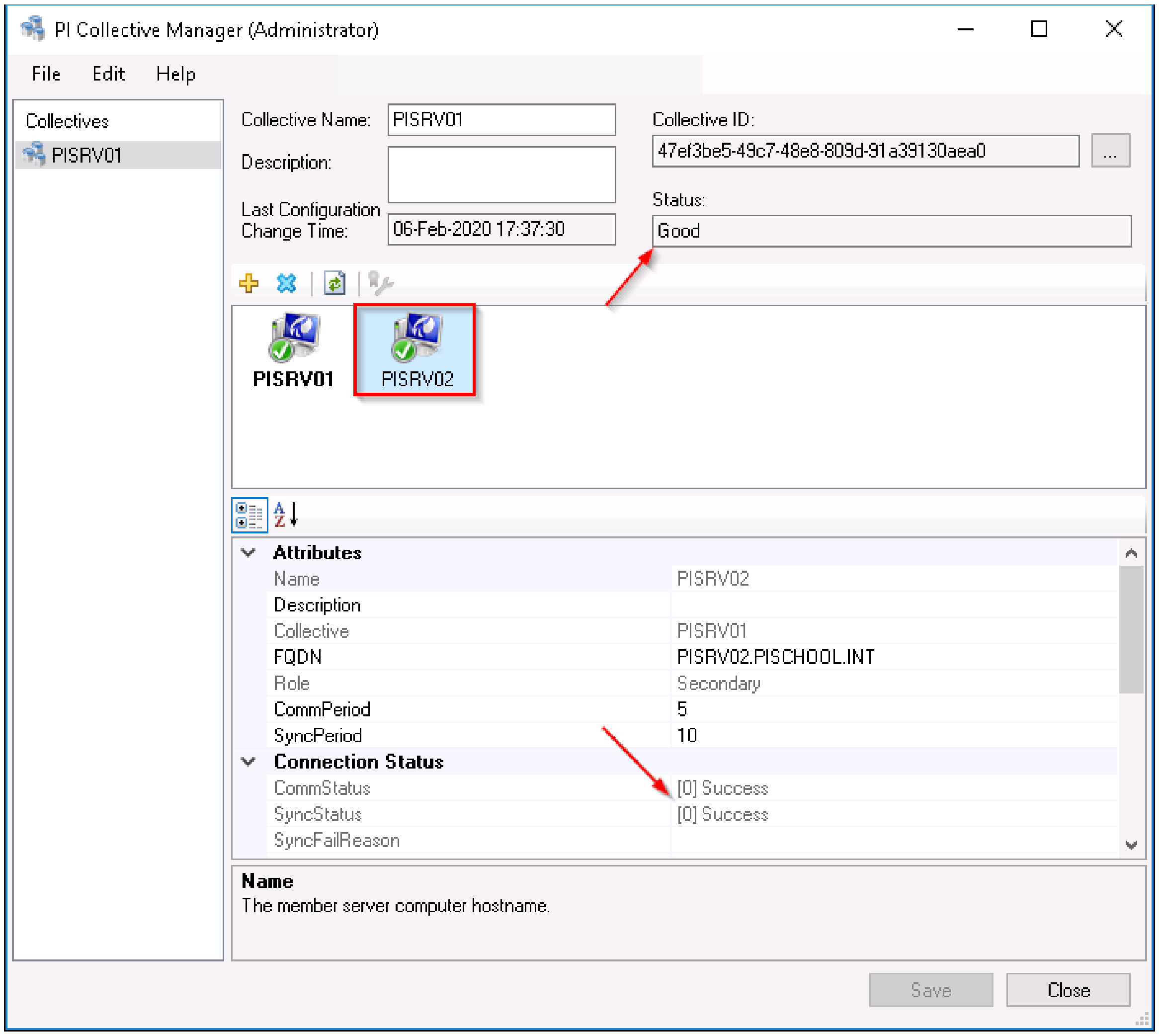
f. Copy Archive data to PISRV2 by Reinitializing the PI Collective

We're not quite done. Data on PISRV2 is missing up to the point where we configured buffering. Luckily, the data exists on PISRV1. We can Reinitialize PISRV2, which will copy all the archive files from PISRV1 and synchronize the data up to that point. PI Buffer Subsystem will handle anything after that.

Let's **go to the PISRV1** and Reinitialize the PI Collective.

Launch PI Collective Manager from the taskbar.

Note that the Status is Good and the SyncStatus is Success despite the fact that PISRV2 is missing data. The SyncStatus has nothing to do with data; it only cares about synchronizing configuration, such as tag creation/edits and security settings.

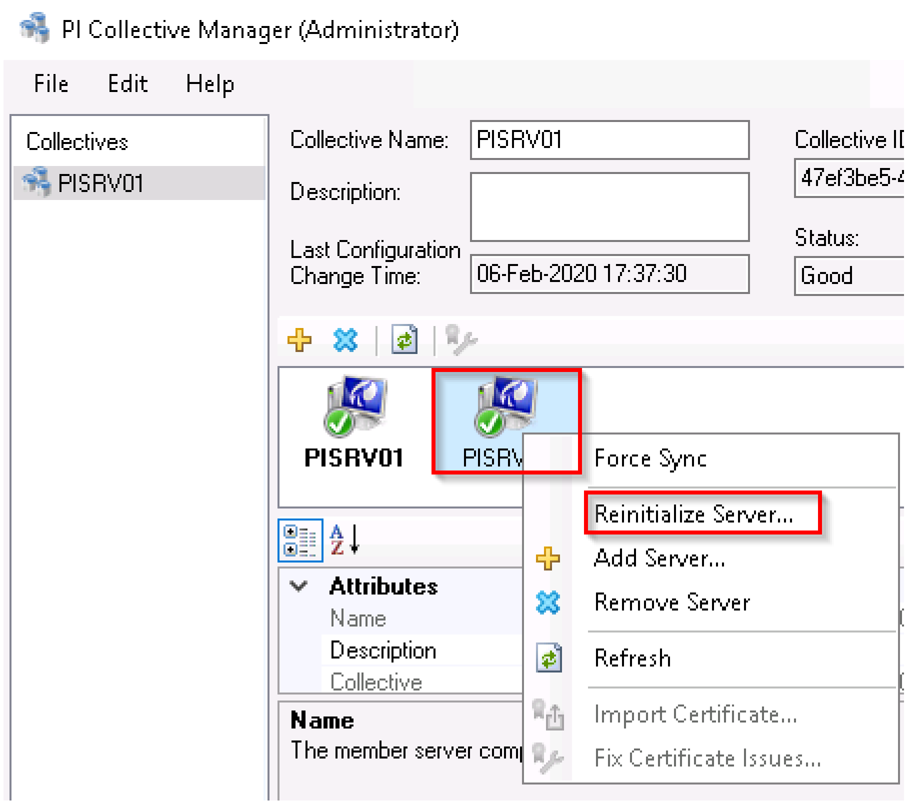


Collective Name: PISRV01
Description:
Last Configuration Change Time: 06-Feb-2020 17:37:30
Collective ID: 47ef3be5-49c7-48e8-809d-91a39130aea0
Status: Good

Attributes	
Name	PISRV02
Description	
Collective	PISRV01
FQDN	PISRV02.PISCHOOL.INT
Role	Secondary
CommPeriod	5
SyncPeriod	10
Connection Status	
CommStatus	[0] Success
SyncStatus	[0] Success
SyncFailReason	

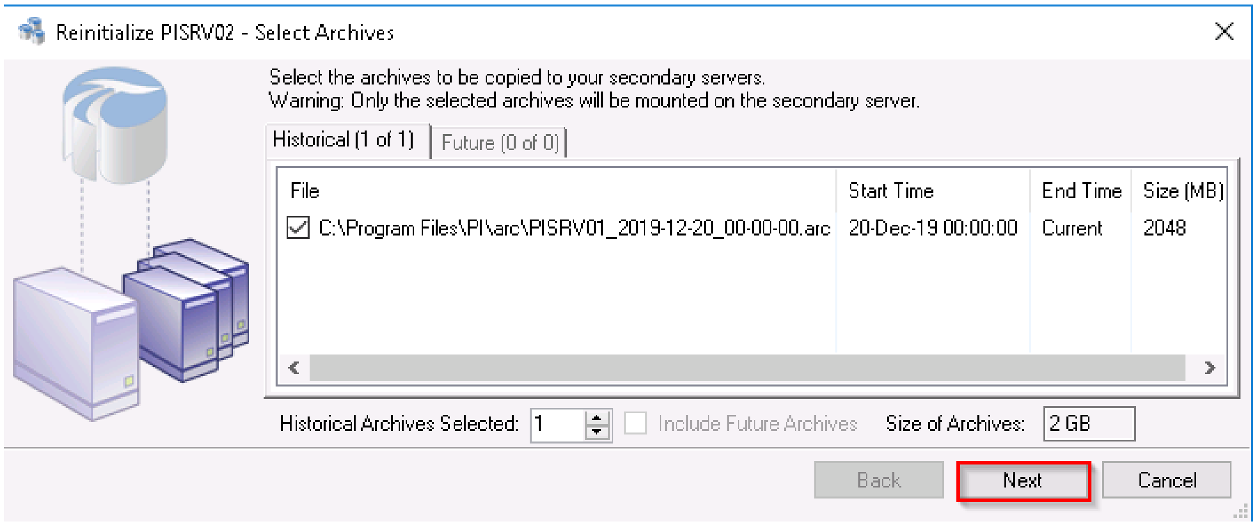
Name
The member server computer hostname.

Provided the administrator has the necessary permissions, Reinitialization is straightforward. Right-click **PISRV2 > Reinitialize Server…**



At this point you have the option to choose which Archive files are copied. Typically, you only have to go as far back as the missing data. In a production environment, you may choose to copy only the first archive using the software and manually copy and register the remaining archives, since there could be 100s of GB of archives and things get messy if the file transfer is interrupted.

In our case, just leave the defaults (copy everything). Next.



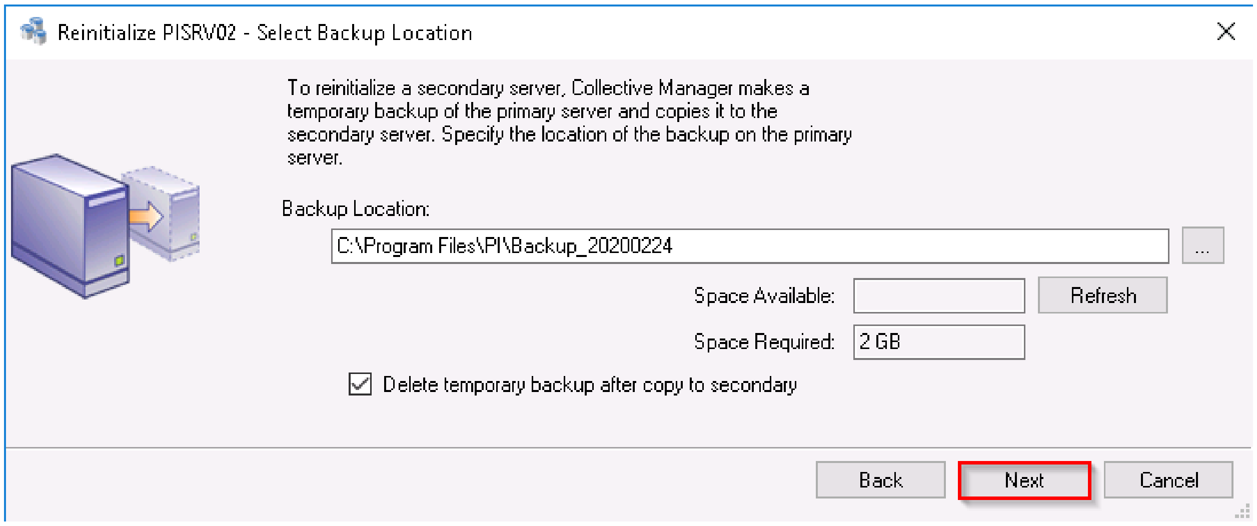
Select the archives to be copied to your secondary servers.
Warning: Only the selected archives will be mounted on the secondary server.

File	Start Time	End Time	Size (MB)
☑ C:\Program Files\PI\arc\PISRV01_2019-12-20_00-00-00.arc	20-Dec-19 00:00:00	Current	2048

Historical Archives Selected: 1 ☐ Include Future Archives Size of Archives: 2 GB

Reinitialization simply takes a backup of the Primary and restores it to the Secondary.

This next screen lets you decide what to do with the temporary backup. Again, defaults are fine. Click **Next**.



Reinitialize PISRV02 - Select Backup Location

To reinitialize a secondary server, Collective Manager makes a temporary backup of the primary server and copies it to the secondary server. Specify the location of the backup on the primary server.

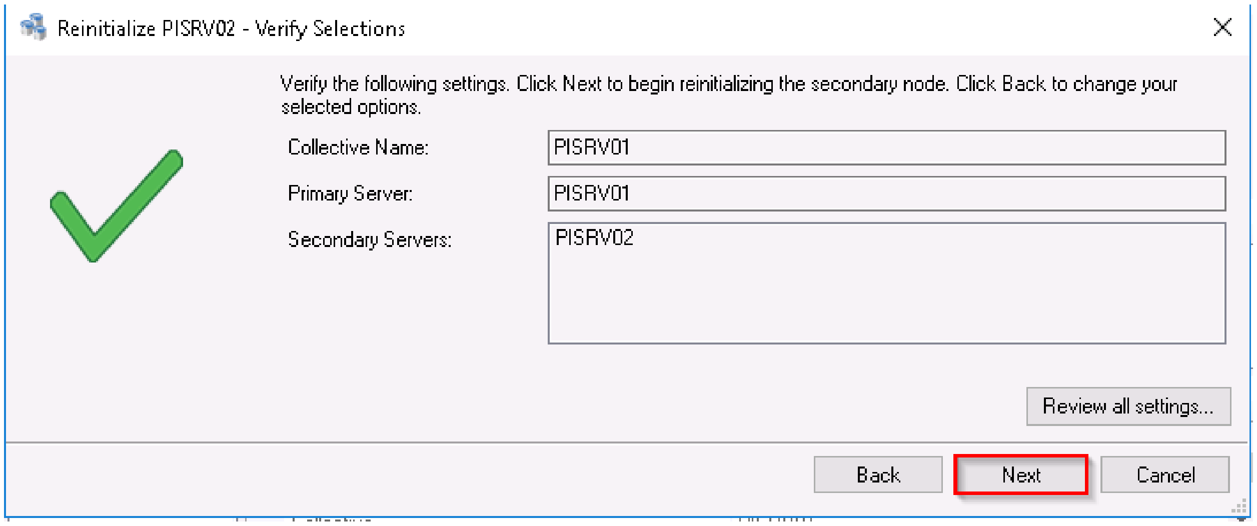
Backup Location: C:\Program Files\PI\Backup_20200224

Space Available:

Space Required: 2 GB

☑ Delete temporary backup after copy to secondary

Typically there’s no need to review the settings. Leave the defaults and click Next.



Reinitialize PISRV02 - Verify Selections

Verify the following settings. Click Next to begin reinitializing the secondary node. Click Back to change your selected options.

Collective Name: PISRV01

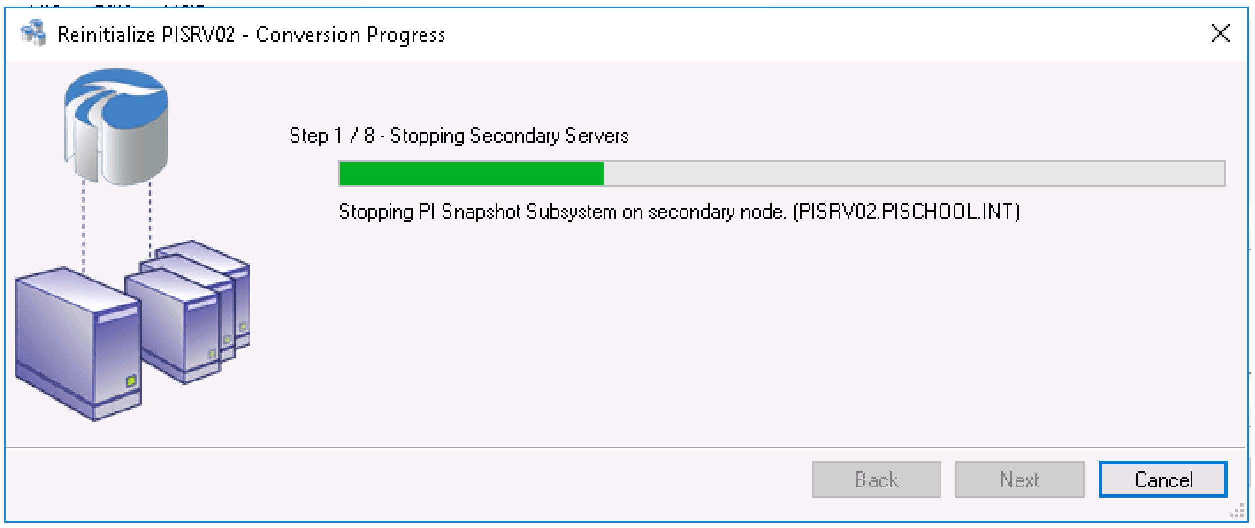
Primary Server: PISRV01

Secondary Servers: PISRV02

The Reinitialization process will:

- Stop PI Data Archive windows services on the Secondary,
- Execute a PI Data Archive backup
- Copy the backup to the Secondary and restore the backup
- Start windows services on the secondary

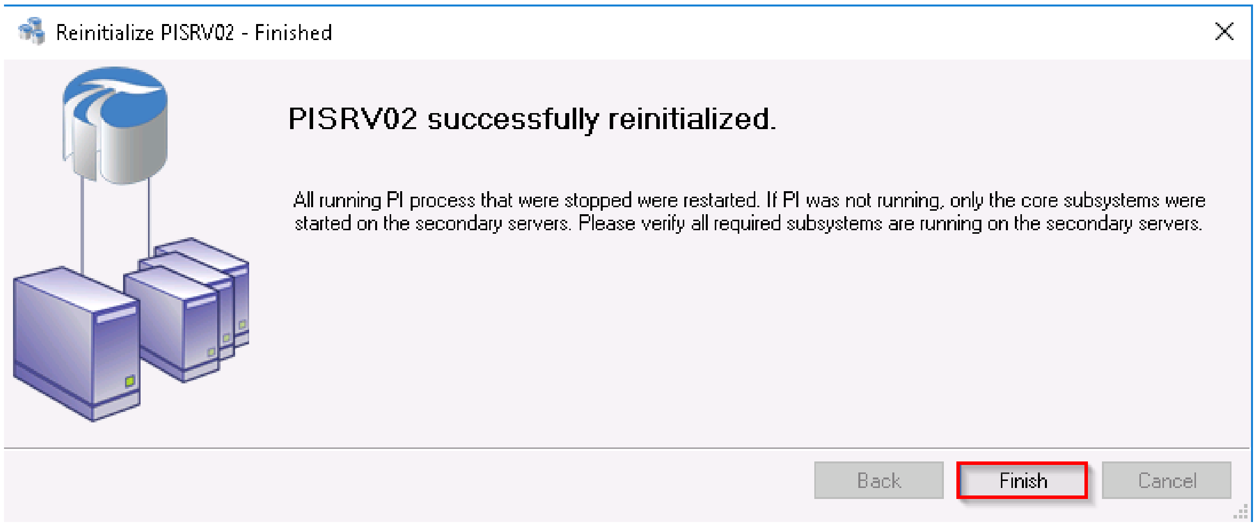
This will take 5 minutes or so.



Step 1 / 8 - Stopping Secondary Servers

Stopping PI Snapshot Subsystem on secondary node. (PISRV02.PISCHOOL.INT)

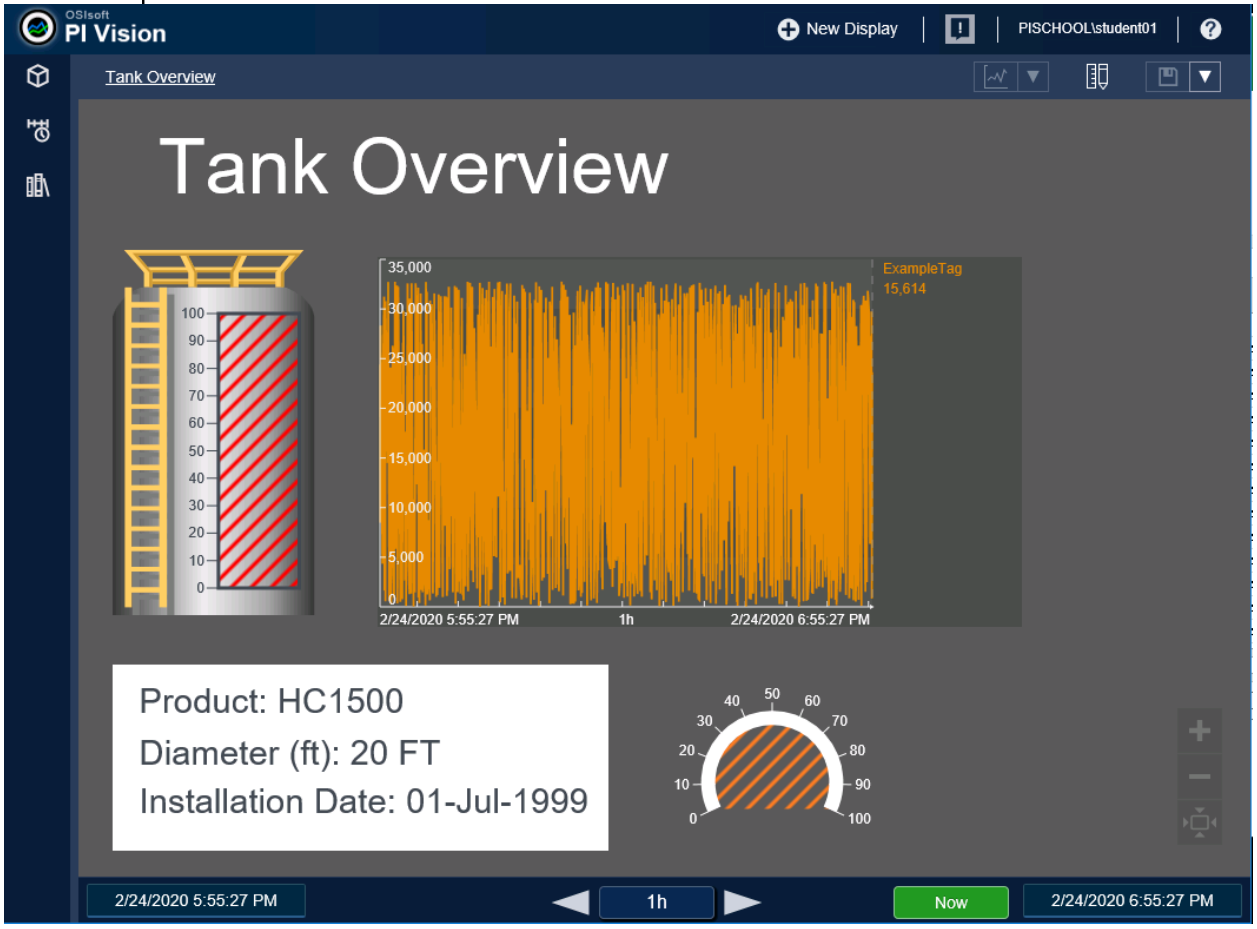
Eventually it should complete successfully.



PISRV02 successfully reinitialized.

All running PI process that were stopped were restarted. If PI was not running, only the core subsystems were started on the secondary servers. Please verify all required subsystems are running on the secondary servers.

Finally, we can verify that the data exists on PISRV2 by changing the collective priority (change the priority of PISRV01 server to -1) on the PISRV01 machine where PI Vision is installed and doing an iisreset. From PICLIENT01, open the PI Vision display link on the desktop.



Product: HC1500
Diameter (ft): 20 FT
Installation Date: 01-Jul-1999

5. Directed Activity – Stale and Bad Data Alerts

5.1 Objective

Interruptions to data flow are just a fact of life for the PI System Administrator. Sometimes there are issues with source data systems. Sometimes there are network interruptions. You do your best to be proactive and prevent these issues, but often they are simply beyond your control. In this exercise we will set up basic stale and bad data alerts. This will help you become aware of problems before they impact your users, or at least make it look like you're on top of things.

5.2 Tasks

- Create a basic PI AF Element
- Create a basic Event Frame Template
- Create a basic PI Analysis to detect stale or bad data
- Create a PI Notification to send an email when stale or bad data is detected

5.3 Step-by-Step Instructions

a. Discussion

What are some of the worst things that can happen when data is bad or stale?

- Bad data is fed into a report, and it becomes a hassle to clean up and/or recall the report.
- People make decisions based on incorrect data.
- Users lose faith in the data and turn to alternatives or more primitive methods

b. Setting Expectations

There are a lot of steps involved in setting up a simple PI Notification and we could easily spend several hours or a day learning all the details of each individual step. These topics are covered much more thoroughly in a 4-day course called Building PI System Assets and Analytics with PI AF, which is highly recommended for PI System Administrators.

c. Getting Started

Ultimately, we want to get to the point of setting up a Notification. However, the following prerequisites must be complete first:

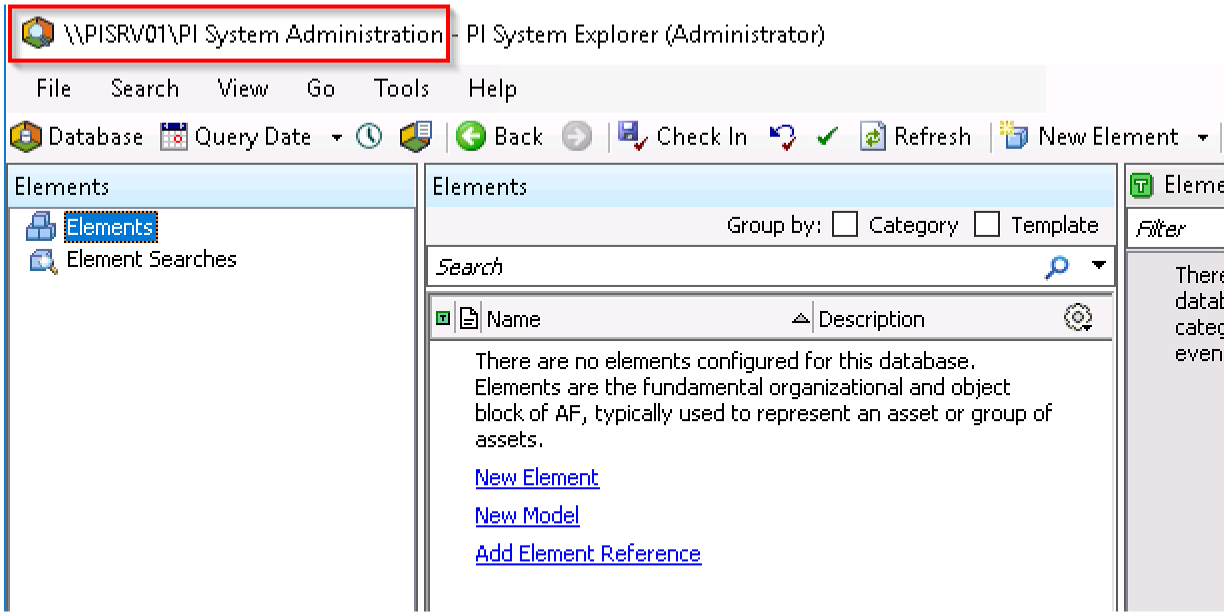
- Input tags must be mapped to PI AF Attributes
- An Event Frame Template must be created
- Logic to generate Event Frames must be configured

Let's start by choosing a tag to monitor. Random.Real4 is as good as any.

First, we'll build a simple PI AF hierarchy and map Random.Real4.

On PICLIENT01, Open PI System Explorer (PSE) which is pinned to taskbar.

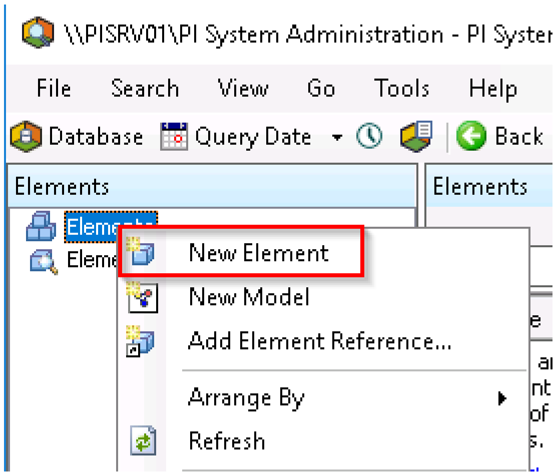
You will be connected to PISRV01 PI AF Server and the PI System Administration PI AF default database.


\\PISRV01\PI System Administration - PI System Explorer (Administrator)
File Search View Go Tools Help
Database Query Date Back Check In Refresh New Element
Elements
Elements
Element Searches
Elements
Group by: Category Template
Search
Name Description
New Element
New Model
Add Element Reference
Filter

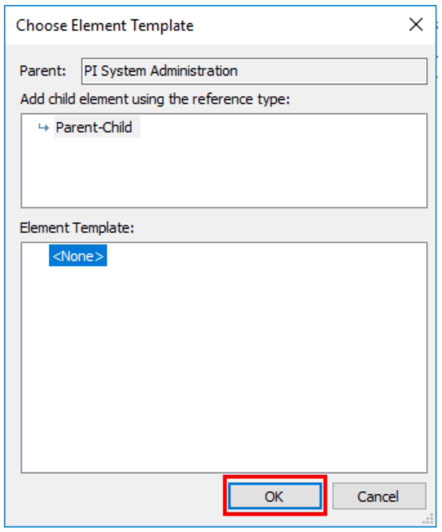
There are no elements configured for this database. Elements are the fundamental organizational and object block of AF, typically used to represent an asset or group of assets.

Next, we'll create a New top-level Element to contain Stale and Bad Data alerts.

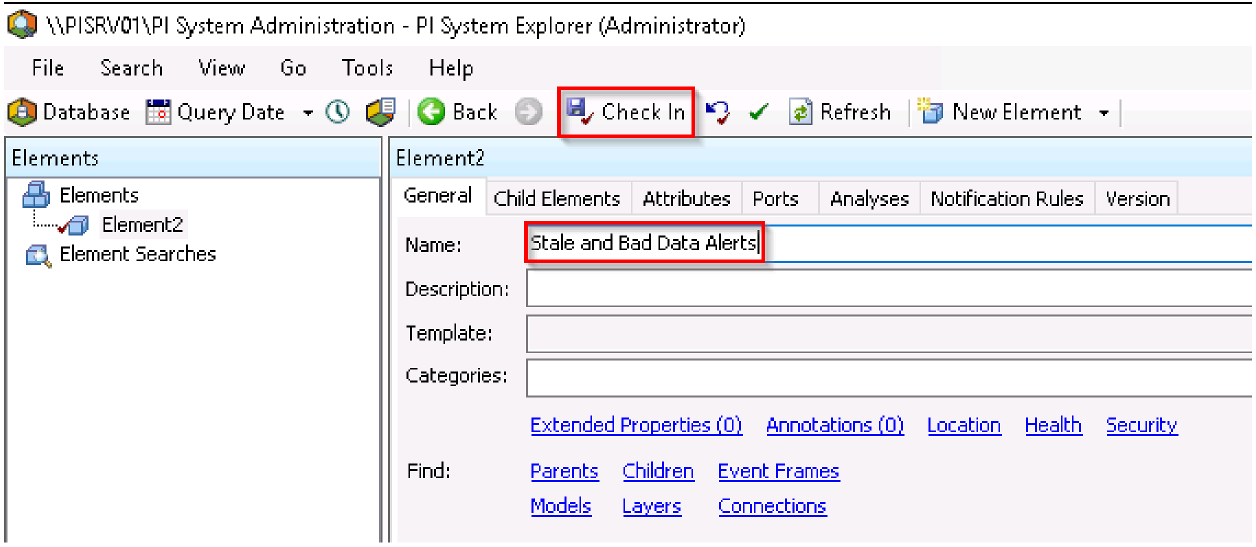
Right-click Elements and select New Element.



Leave the defaults and click OK.

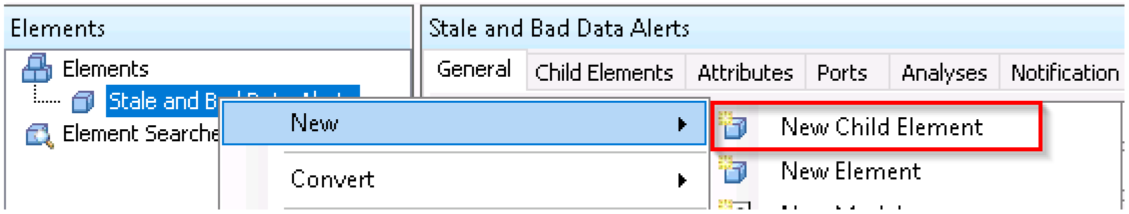


Enter Stale and Bad Data Alerts as the name and click Check In.

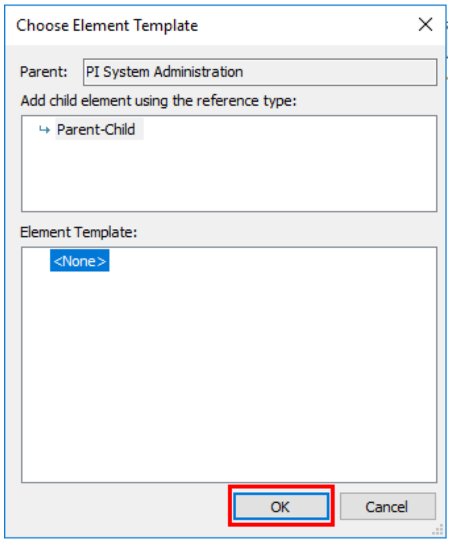


Now we'll create an element for Random.Real4.

Right-click the Stale and Bad Data Alerts Element and create a new child Element.



Again, leave the defaults and click OK.


Choose Element Template

Parent: PI System Administration

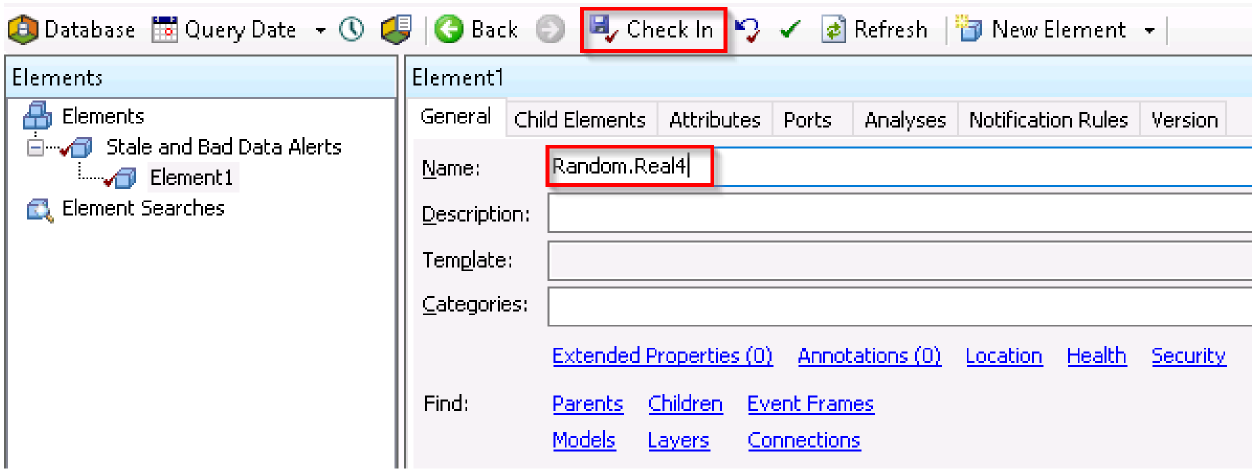
Add child element using the reference type:

Parent-Child

Element Template:

<None>

Name the Element Random.Real4 and check in.



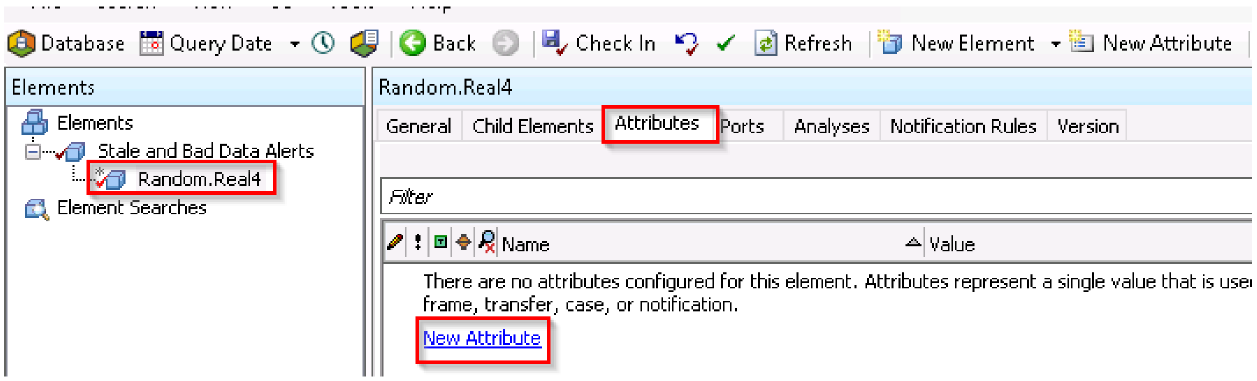
Name: Random.Real4

Description:

Template:

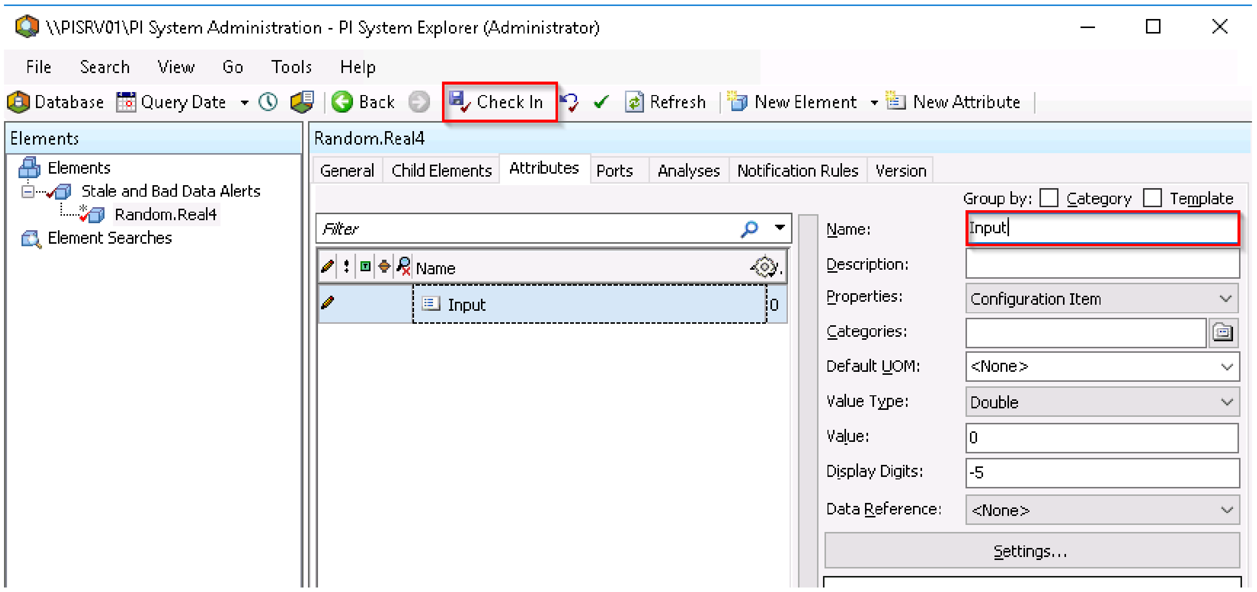
Categories:

With the Random.Real4 Element selected, select the Attributes tab, and create a new attribute.



There are no attributes configured for this element. Attributes represent a single value that is use frame, transfer, case, or notification.

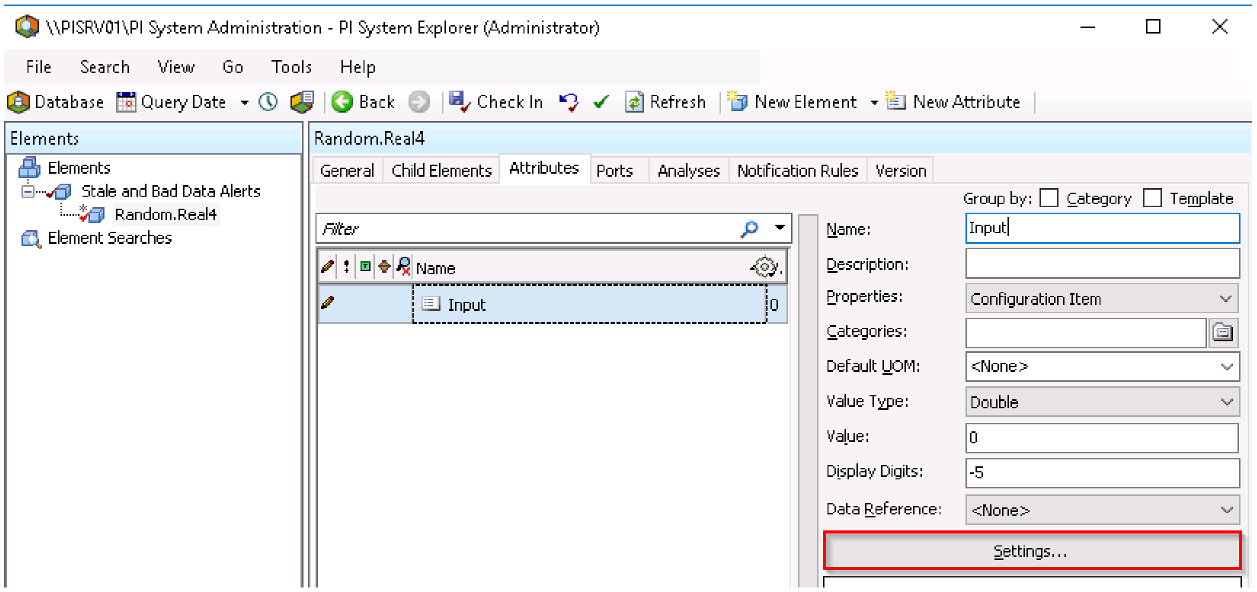
Name the attribute Input and check in.



Group by: Category Template

Name: Input
Description:
Properties: Configuration Item
Categories:
Default UOM: <None>
Value Type: Double
Value: 0
Display Digits: -5
Data Reference: <None>

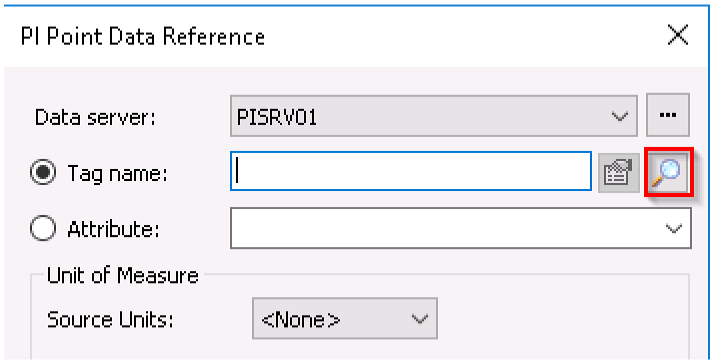
Click Settings.



Group by: Category Template

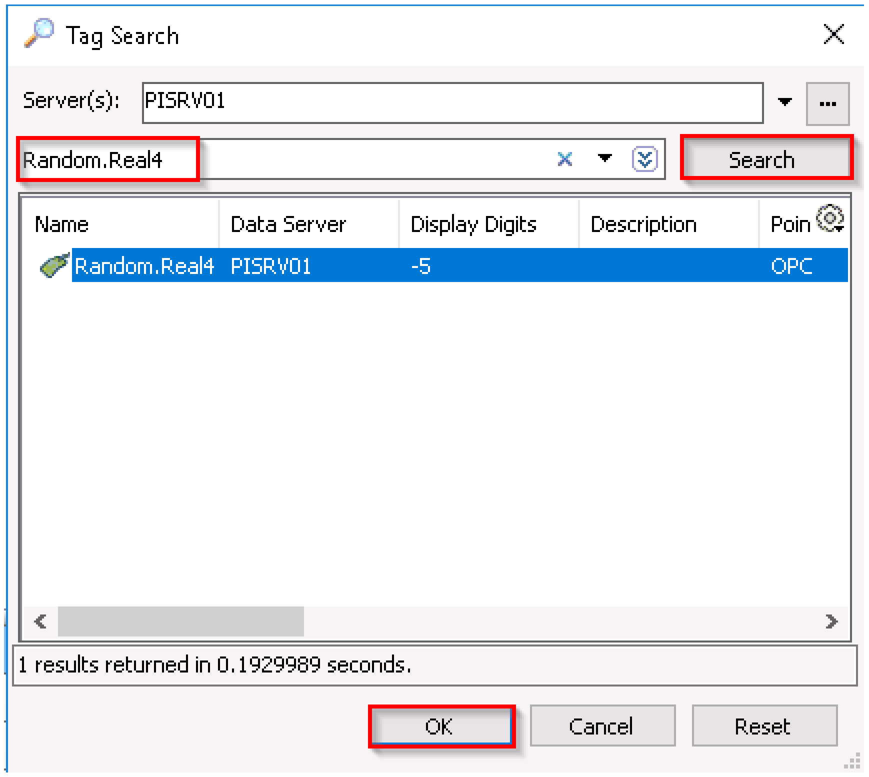
Name: Input
Description:
Properties: Configuration Item
Categories:
Default UOM: <None>
Value Type: Double
Value: 0
Display Digits: -5
Data Reference: <None>

Click the search icon next to the tag name field.



Data server: PISRV01
Tag name:
Attribute:
Source Units: <None>

Search for Random.Real4, select it, and click OK.


Tag Search

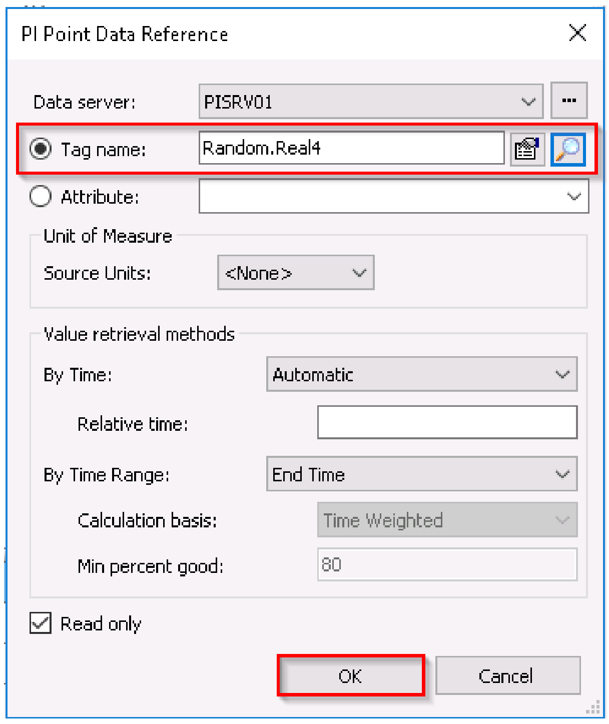
Server(s): PISRV01

Random.Real4

Name	Data Server	Display Digits	Description	Poin
Random.Real4	PISRV01	-5		OPC

1 results returned in 0.1929989 seconds.

Random.Real4 should now be in the Tag name field. Of course, we could have just typed “Random. Real4” here directly without searching. Leave the defaults and click OK.


PI Point Data Reference

Data server: PISRV01

(●) Tag name: Random.Real4

( ) Attribute:

Unit of Measure

Source Units: <None>

Value retrieval methods

By Time: Automatic

Relative time:

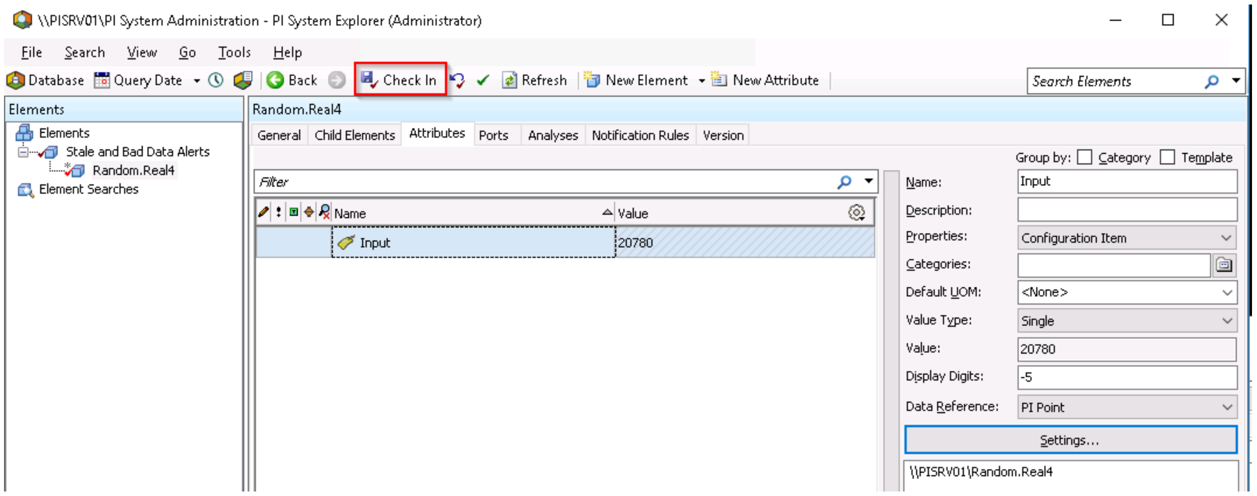
By Time Range: End Time

Calculation basis: Time Weighted

Min percent good: 80

☑ Read only

You should see a number for the value of Input. **Check In.** Since this is random data everyone's number will be different and won't match the workbook.

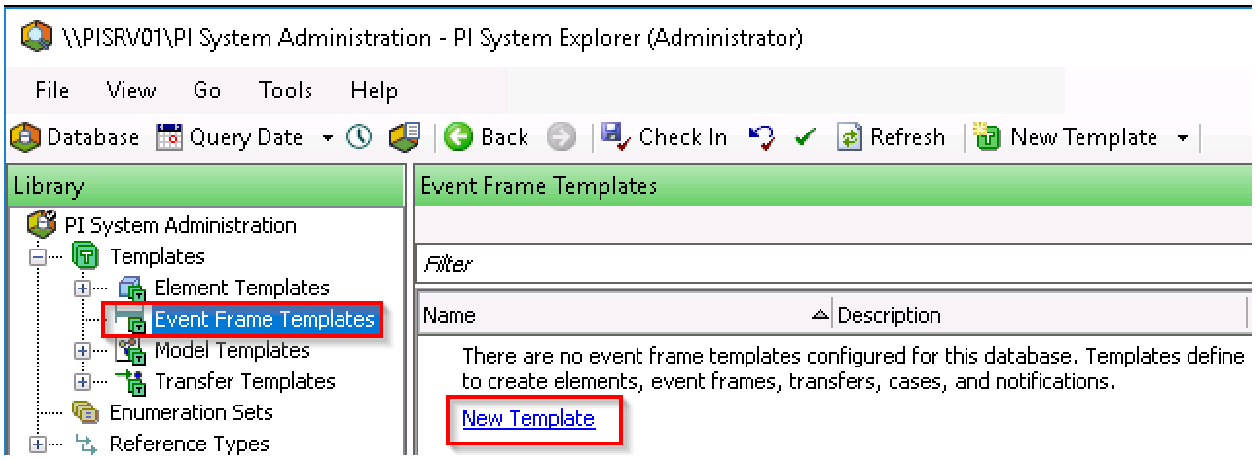

\\PISRV01\PI System Administration - PI System Explorer (Administrator)
File Search View Go Tools Help
Database Query Date Back Check In Refresh New Element New Attribute
Search Elements
Elements
Elements
Stale and Bad Data Alerts
Random.Real4
Element Searches
Random.Real4
General Child Elements Attributes Ports Analyses Notification Rules Version
Filter
Name Value
Group by: Category Template
Settings...

Name	Value
Input	20780

Name: Input
Description:
Properties: Configuration Item
Categories:
Default UOM: <None>
Value Type: Single
Value: 20780
Display Digits: -5
Data Reference: PI Point
\\PISRV01\Random.Real4

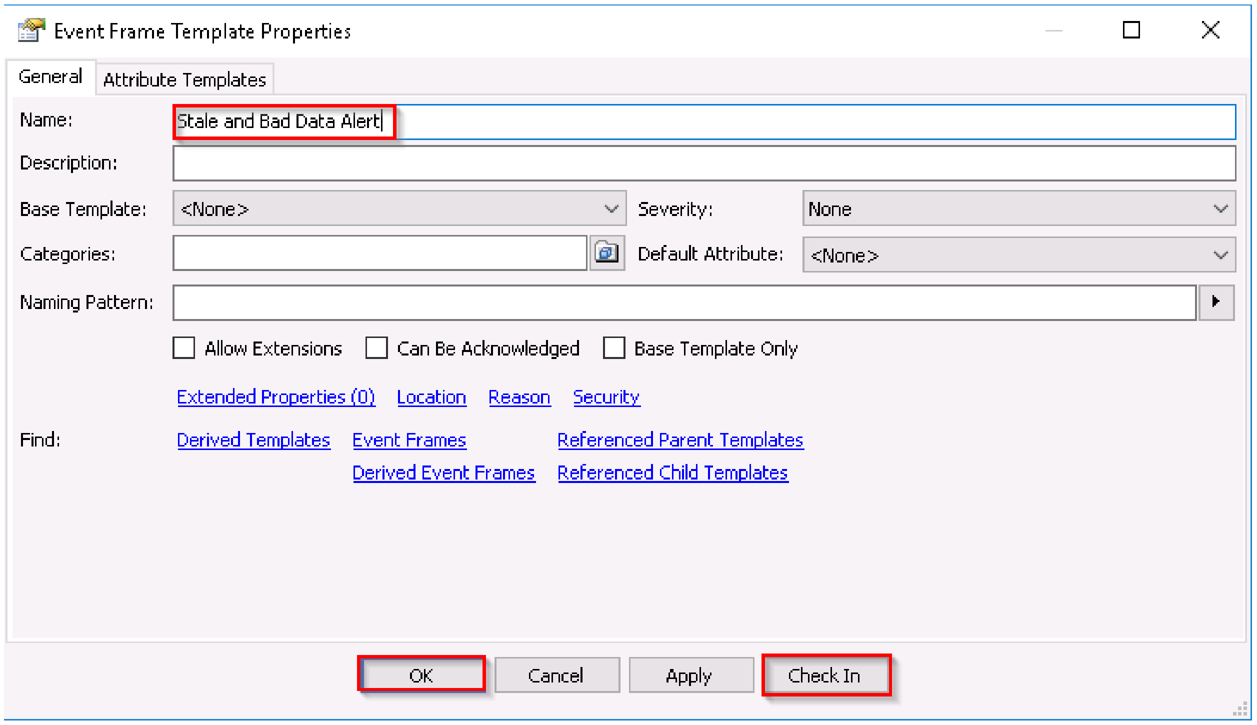
Next, we'll create an Event Frame template. If this is your first experience with Event Frames, you'll just have to accept the mystery for now. For our purposes it's a prerequisite for configuring PI Notifications that can't be avoided. We recommend taking the Building PI System Assets and Analytics with AF class to become more familiar with PI AF objects.

Go to the Library, Right-click Event Frame Templates, Select New Template.



There are no event frame templates configured for this database. Templates define to create elements, event frames, transfers, cases, and notifications.

Name it Stale and Bad Data Alert, Check in > then OK


Event Frame Template Properties

General | Attribute Templates

Name: Stale and Bad Data Alert

Description:

Base Template: <None>

Severity: None

Categories:

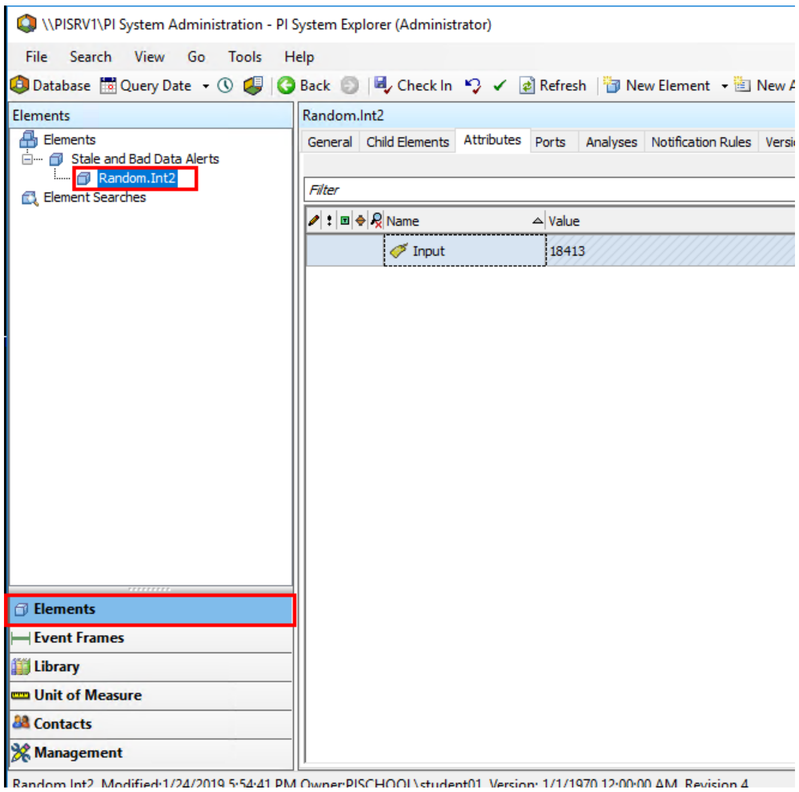
Default Attribute: <None>

Naming Pattern:

☐ Allow Extensions ☐ Can Be Acknowledged ☐ Base Template Only

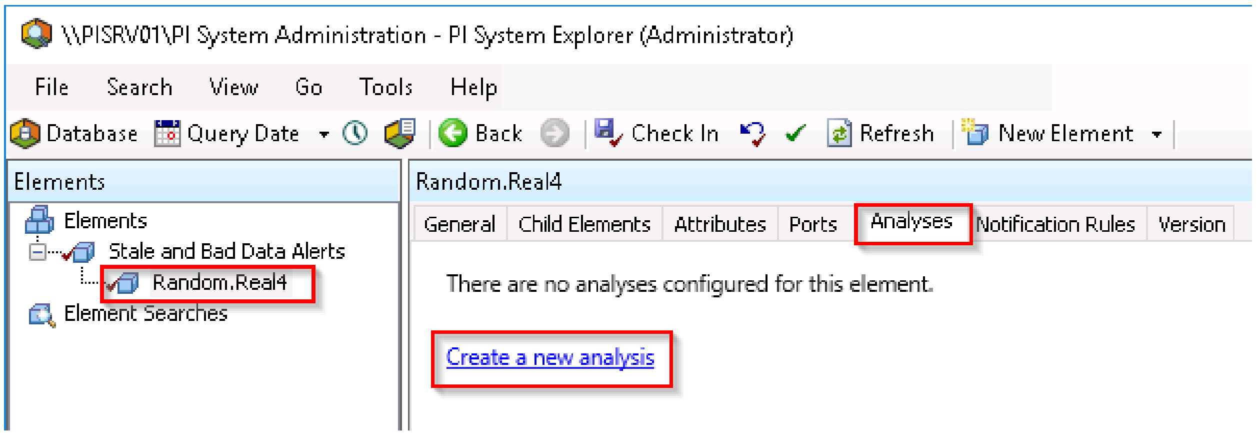
The Library warrants some explanation. This is where all the Templates and other building blocks used in PI AF are configured.

Next, we'll configure the logic that defines Stale and Bad data. Go back to Elements and select the Random.Real4 Element.


\\PISRV1\PI System Administration - PI System Explorer (Administrator)
File Search View Go Tools Help
Database Query Date Back Check In Refresh New Element
Elements
Elements
Stale and Bad Data Alerts
Random.Int2
Element Searches
Elements
Event Frames
Library
Unit of Measure
Contacts
Management
Random.Int2
General Child Elements Attributes Ports Analyses Notification Rules
Filter
Name Value
Input 18413

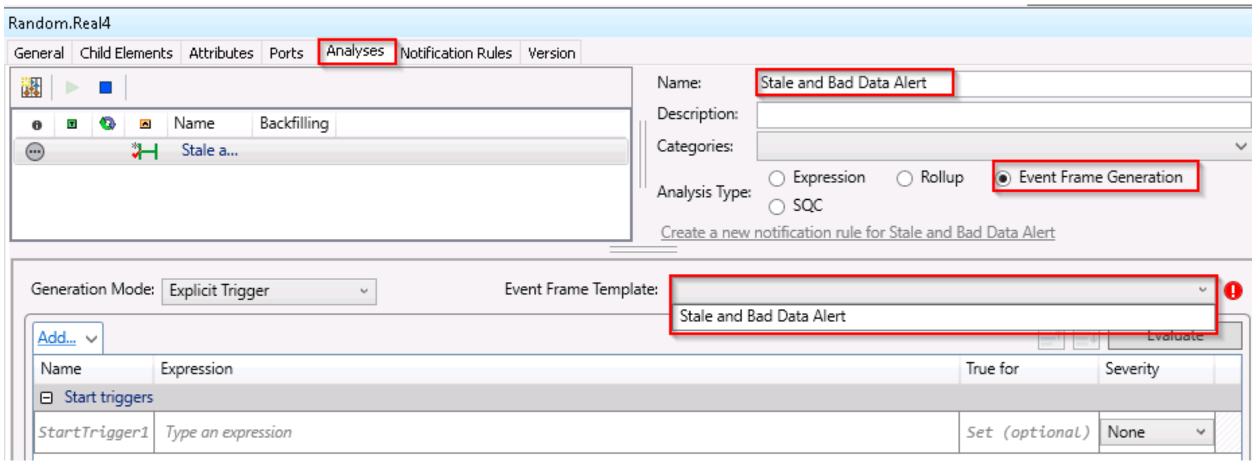
Random.Int2 Modified:1/24/2019 5:54:41 PM Owner:PISCHOOL\student01 Version: 1/1/1970 12:00:00 AM Revision 4

Click the Analyses tab and Create a new analysis.

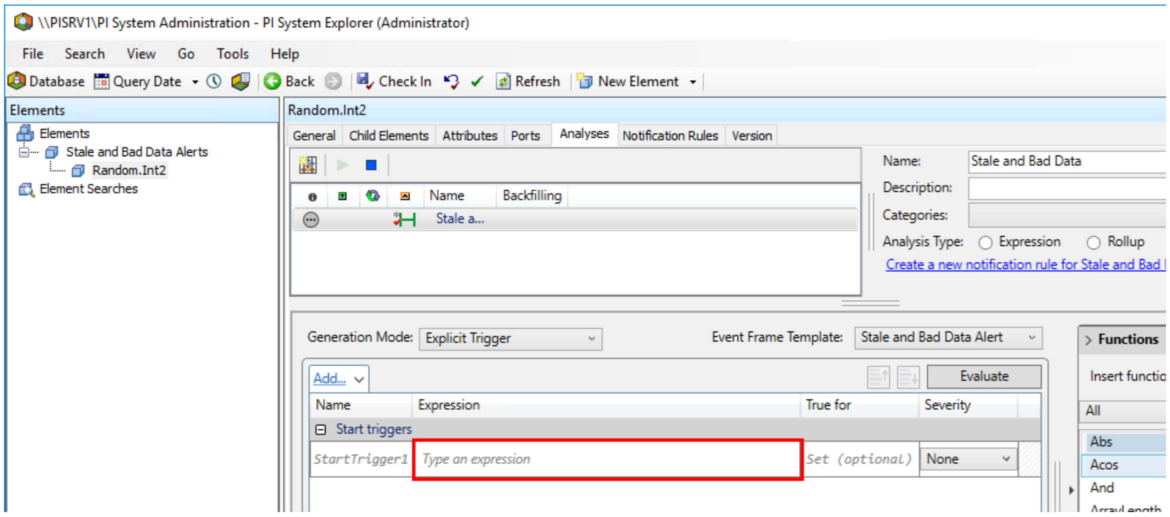

\\PISRV01\PI System Administration - PI System Explorer (Administrator)
File Search View Go Tools Help
Database Query Date Back Check In Refresh New Element
Elements
Elements
Stale and Bad Data Alerts
Random.Real4
Element Searches
Random.Real4
General Child Elements Attributes Ports Analyses Notification Rules Version
Create a new analysis

There are no analyses configured for this element.

Name it Stale and Bad Data, select Event Frame Generation, and then choose Stale and Bad Data Alert as the Event Frame Template.

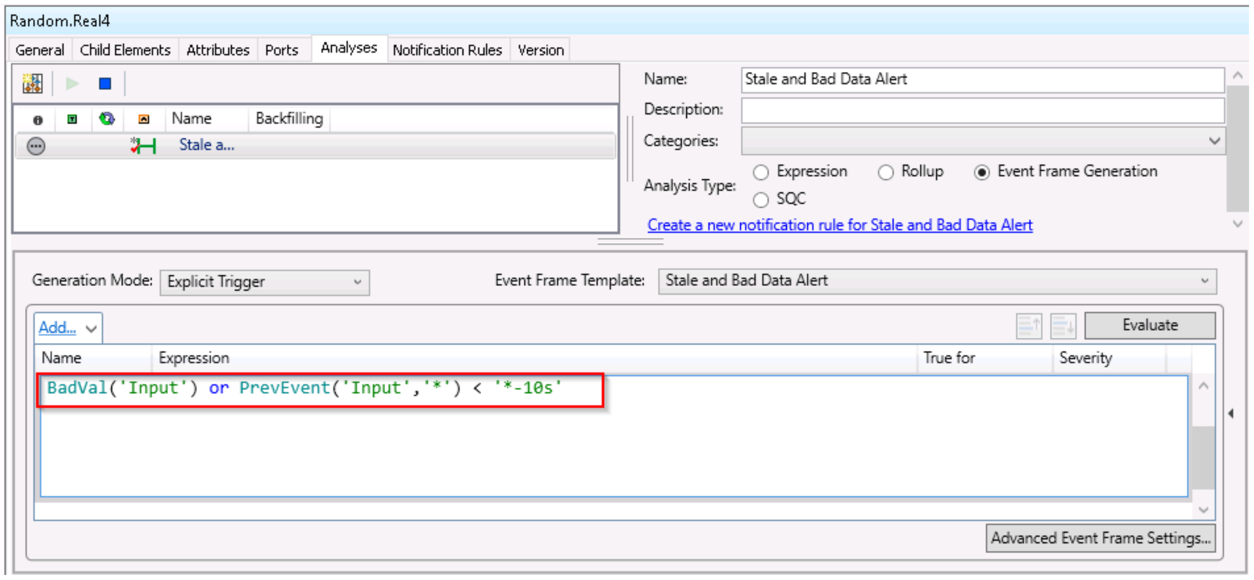


Click on the field that says Type an expression.



Enter the following expression. **You can copy and paste the text from the workbook PDF in the Class folder on PICLIENT01.**

```
BadVal('Input') or PrevEvent('Input','*') < '*-10s'
```



Name: Stale and Bad Data Alert
Description:
Categories:
Analysis Type: Expression Rollup [✓ Event Frame Generation] SQC
Generation Mode: Explicit Trigger
Event Frame Template: Stale and Bad Data Alert

```
BadVal('Input') or PrevEvent('Input','*') < '*-10s'
```

The BadVal() function checks whether the tag value is bad. In simple terms, bad generally means that there's an error status string where there should be a number. "Configure" would be considered bad. Other common bad values you'll probably encounter are "Calc failed" and "I/O Timeout".

The PrevEvent() function returns the last timestamp prior to '*', and '*' is shorthand for the time right now. The result is the timestamp of the last value sent to PI.

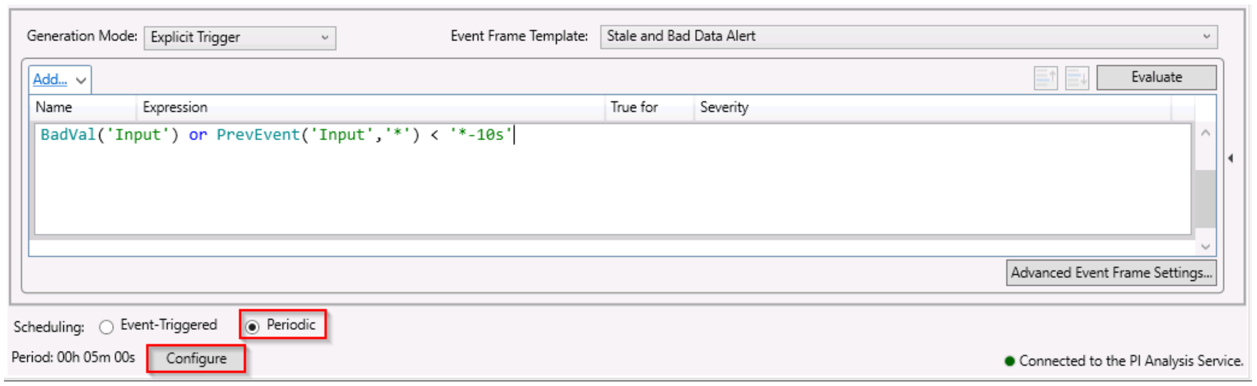
The time abbreviation '*-10s' is shorthand for 10 seconds ago.

Hence, the triggering condition is true if the value of Input (mapped to Random. Real4) is Bad or more than 10 seconds stale.

Of course, being 10 seconds stale is normal for many tags. It really depends on the tag and the application in which the tag is being used. You might only care if the value is stale for an hour, or even a day. We are using 10 seconds here for testing purposes so that we only have to wait 10 seconds to see if it works.

The default Event-Triggered scheduling won't work for Stale data. This would mean the trigger logic would only be checked whenever a new value is sent to PI and would never detect staleness. We need to use Periodic.

Click Periodic, then Configure

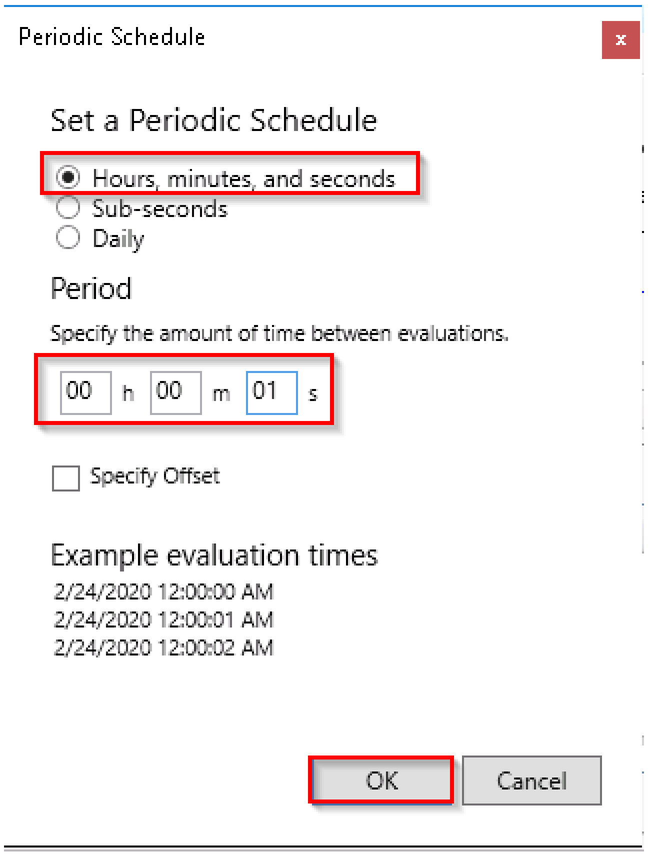


Generation Mode: Explicit Trigger
Event Frame Template: Stale and Bad Data Alert

Name	Expression	True for	Severity
	`BadVal('Input') or PrevEvent('Input','*') < '*-10s'`		

Scheduling: ○ Event-Triggered ◉ Periodic
Period: 00h 05m 00s
Connected to the PI Analysis Service.

Change the period to every 1 second for now and click OK. We will change it to something more realistic later.



Periodic Schedule

Set a Periodic Schedule

◉ Hours, minutes, and seconds
○ Sub-seconds
○ Daily

Period

Specify the amount of time between evaluations.

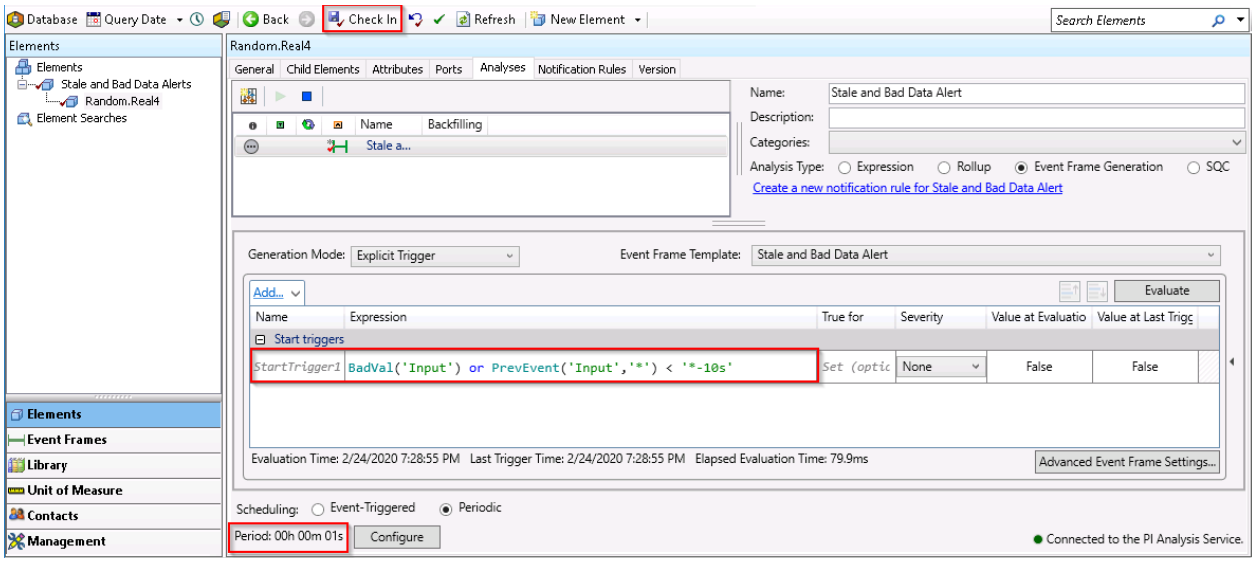
00 h 00 m 01 s

☐ Specify Offset

Example evaluation times

2/24/2020 12:00:00 AM
2/24/2020 12:00:01 AM
2/24/2020 12:00:02 AM

Verify the analysis configuration and Check In; The Analysis should start after a few seconds.



Name: Stale and Bad Data Alert
Description:
Categories:
Analysis Type: Expression Rollup [✓ Event Frame Generation] SQC
Create a new notification rule for Stale and Bad Data Alert

Generation Mode: Explicit Trigger
Event Frame Template: Stale and Bad Data Alert

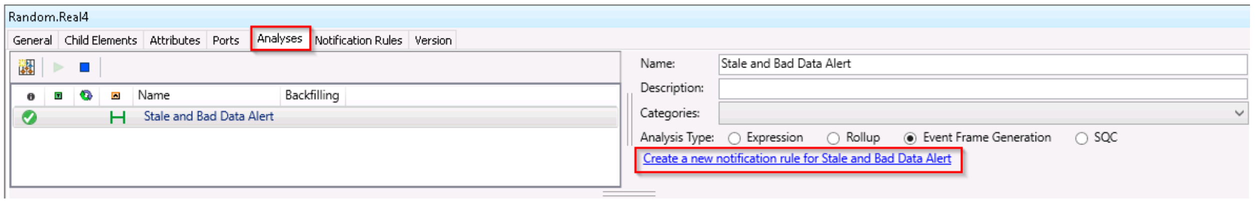
Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
Start triggers					
StartTrigger1	`BadVal('Input') or PrevEvent('Input','*') < '*-10s'`	Set (optic	None	False	False

Evaluation Time: 2/24/2020 7:28:55 PM   Last Trigger Time: 2/24/2020 7:28:55 PM   Elapsed Evaluation Time: 79.9ms

Scheduling: Event-Triggered [✓ Periodic]
Period: 00h 00m 01s
Connected to the PI Analysis Service.

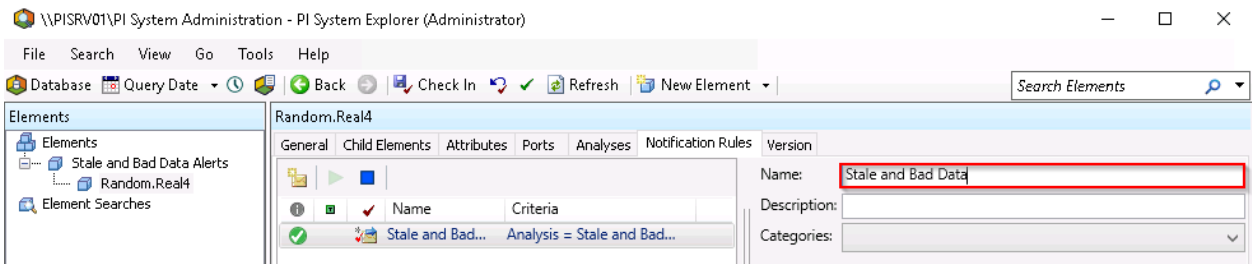
Now to attach a Notification to the Event Frame.

Click ‘Create a new notification rule for Stale and Bad Data’.



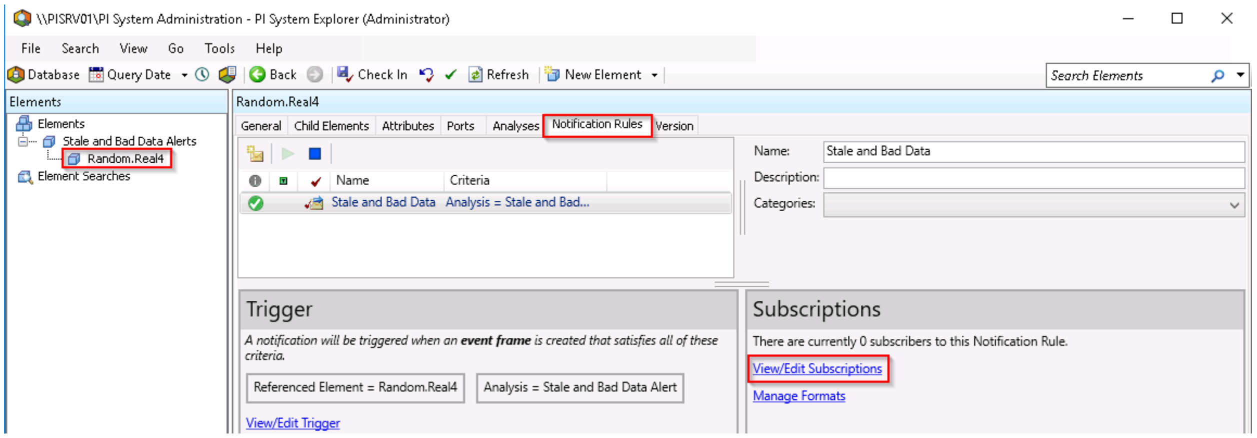
Name: Stale and Bad Data Alert
Description:
Categories:
Analysis Type: Expression Rollup [✓ Event Frame Generation] SQC
Create a new notification rule for Stale and Bad Data Alert

Name it Stale and Bad Data.



Name: Stale and Bad Data
Description:
Categories:

Click View/Edit Subscriptions to add a subscriber.


\\PISRV01\PI System Administration - PI System Explorer (Administrator)
File Search View Go Tools Help
Database Query Date Back Check In Refresh New Element
Search Elements
Elements
Stale and Bad Data Alerts
Random.Real4
Element Searches
General Child Elements Attributes Ports Analyses Notification Rules Version
View/Edit Trigger
View/Edit Subscriptions
Manage Formats

Name	Criteria
Stale and Bad Data	Analysis = Stale and Bad...

Name: Stale and Bad Data
Description:
Categories:

Trigger

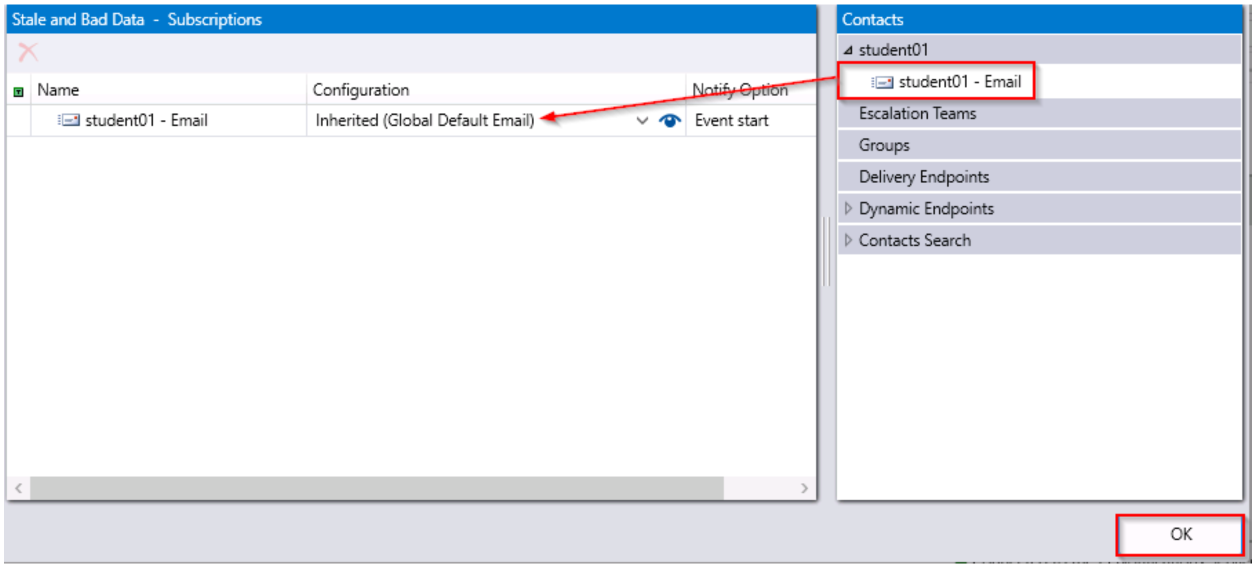
*A notification will be triggered when an **event frame** is created that satisfies all of these criteria.*

Referenced Element = Random.Real4 | Analysis = Stale and Bad Data Alert

Subscriptions

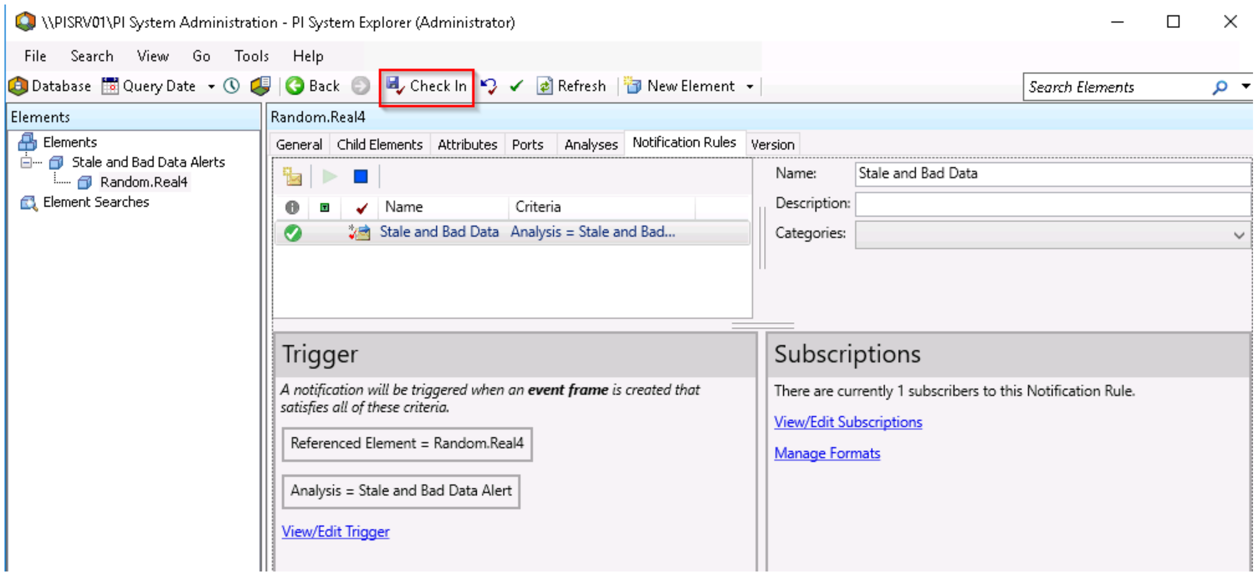
There are currently 0 subscribers to this Notification Rule.

Expand student01, drag and drop student01 – Email to the subscribers pane, click OK.



Name	Configuration	Notify Option
student01 - Email	Inherited (Global Default Email)	Event start

Check in, the Notification Rule should start after a few seconds.


\\PISRV01\PI System Administration - PI System Explorer (Administrator)
File Search View Go Tools Help
Database Query Date Back Check In Refresh New Element
Search Elements
Elements
Elements
Stale and Bad Data Alerts
Random.Real4
Element Searches
Random.Real4
General Child Elements Attributes Ports Analyses Notification Rules Version

Name	Criteria
Stale and Bad Data	Analysis = Stale and Bad...

Name: Stale and Bad Data
Description:
Categories:

Trigger

A notification will be triggered when an ***event frame*** *is created that satisfies all of these criteria.*

Referenced Element = Random.Real4

Analysis = Stale and Bad Data Alert

View/Edit Trigger

Subscriptions

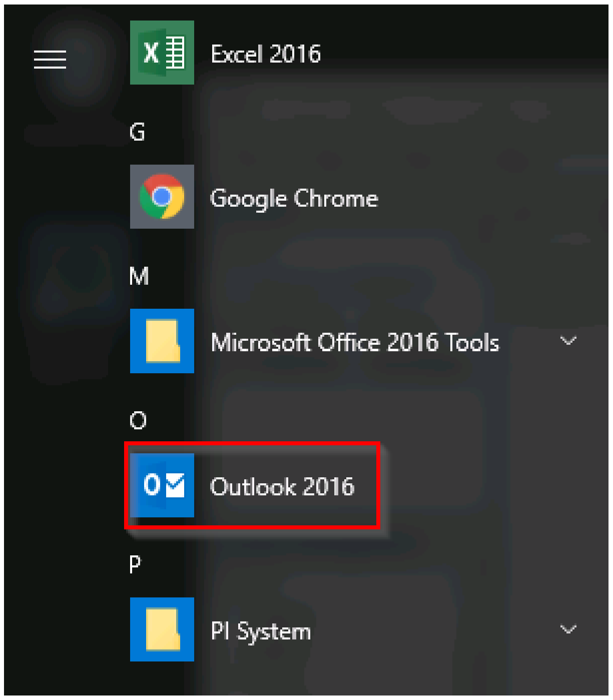
There are currently 1 subscribers to this Notification Rule.

View/Edit Subscriptions

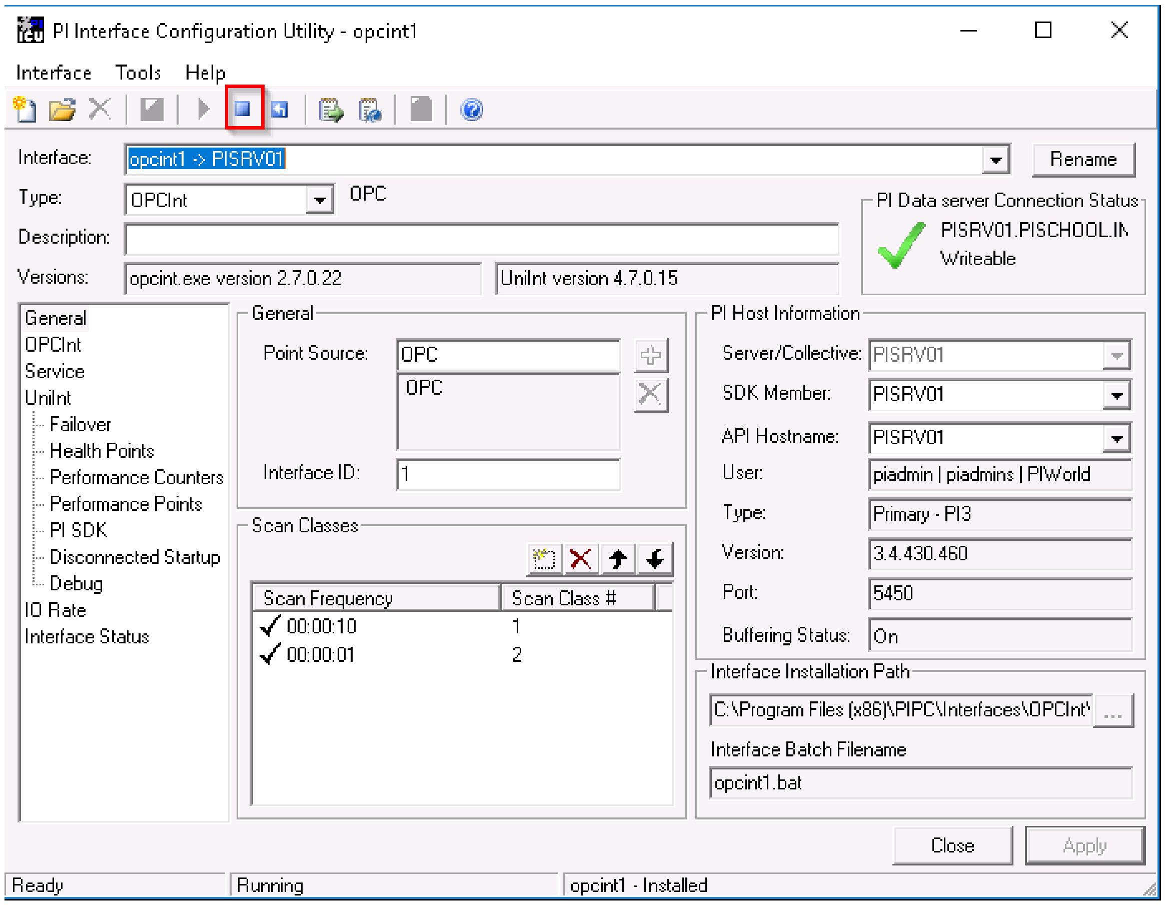
Manage Formats

Now to test it. In a production environment you’d use a less intrusive method, but we’re just going to stop the OPC Interface on PIINT01, then wait a little more than 10 seconds and see if we get an email.

First open Outlook **on PICLIENT01.**



Connect to PIINT01, open PI ICU, select opcint1, and stop the interface service.


PI Interface Configuration Utility - opcint1

Interface: opcint1 -> PISRV01

Type: OPCInt OPC

Description:

Versions: opcint.exe version 2.7.0.22 | UniInt version 4.7.0.15

PI Data server Connection Status: PISRV01.PISCHOOL.IN Writeable

General

Point Source: OPC

Interface ID: 1

Scan Classes

Scan Frequency	Scan Class #
✓ 00:00:10	1
✓ 00:00:01	2

PI Host Information

Server/Collective: PISRV01

SDK Member: PISRV01

API Hostname: PISRV01

User: piadmin | piadmins | PIWorld

Type: Primary - PI3

Version: 3.4.430.460

Port: 5450

Buffering Status: On

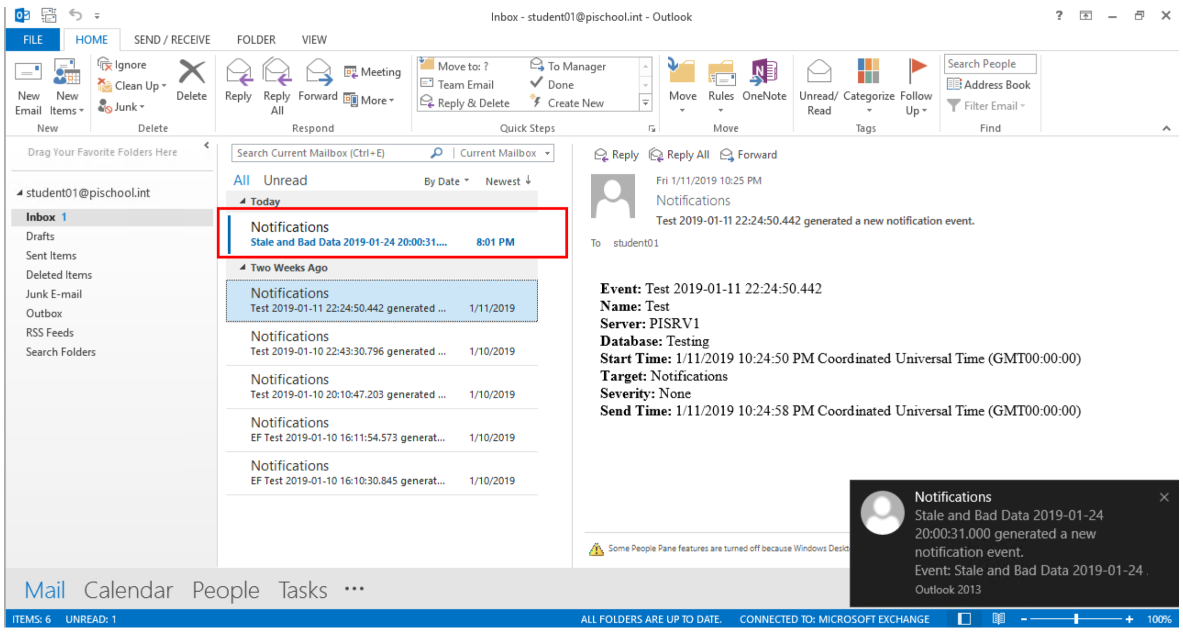
Interface Installation Path: C:\Program Files (x86)\PIPC\Interfaces\OPCInt\

Interface Batch Filename: opcint1.bat

Ready | Running | opcint1 - Installed

The Notification should trigger after 10 seconds, but it will take a few more seconds to propagate through the Exchange server.

You should see a new email with the default email formatting. The Sender, Subject, and Body are all fully customizable, we just didn't change the default settings at all.


Inbox - student01@pischool.int - Outlook
FILE HOME SEND / RECEIVE FOLDER VIEW
New Email New Items Ignore Clean Up Junk Delete Reply Reply All Forward Meeting More Move to: ? Team Email Reply & Delete To Manager Done Create New Move Rules OneNote Unread/ Read Categorize Follow Up Search People Address Book Filter Email
Mail Calendar People Tasks

student01@pischool.int
Inbox 1
Drafts
Sent Items
Deleted Items
Junk E-mail
Outbox
RSS Feeds
Search Folders

Today
Notifications
Stale and Bad Data 2019-01-24 20:00:31.... 8:01 PM

Two Weeks Ago
Notifications
Test 2019-01-11 22:24:50.442 generated ... 1/11/2019
Notifications
Test 2019-01-10 22:43:30.796 generated ... 1/10/2019
Notifications
Test 2019-01-10 20:10:47.203 generated ... 1/10/2019
Notifications
EF Test 2019-01-10 16:11:54.573 generat... 1/10/2019
Notifications
EF Test 2019-01-10 16:10:30.845 generat... 1/10/2019

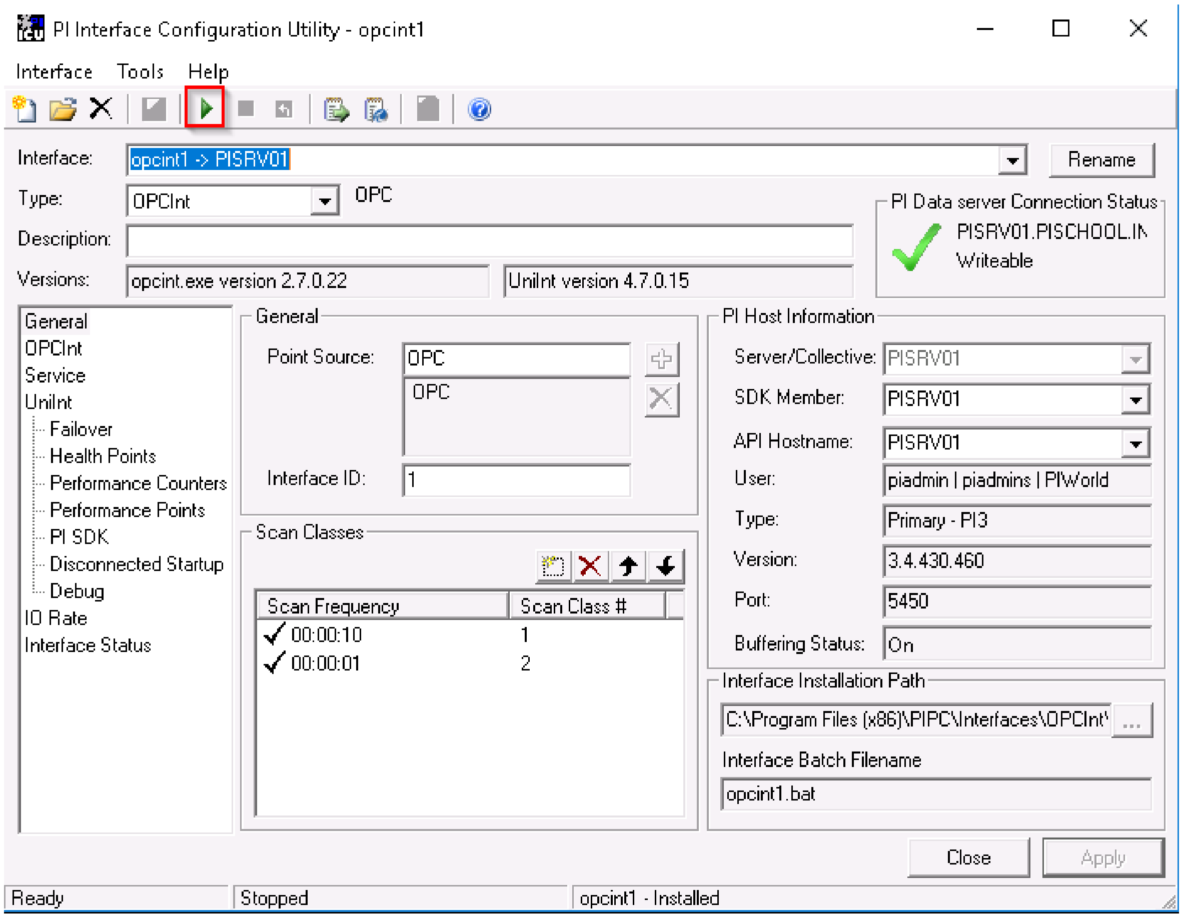
Fri 1/11/2019 10:25 PM
Notifications
Test 2019-01-11 22:24:50.442 generated a new notification event.
To student01

Event: Test 2019-01-11 22:24:50.442
Name: Test
Server: PISRV1
Database: Testing
Start Time: 1/11/2019 10:24:50 PM Coordinated Universal Time (GMT00:00:00)
Target: Notifications
Severity: None
Send Time: 1/11/2019 10:24:58 PM Coordinated Universal Time (GMT00:00:00)

Notifications
Stale and Bad Data 2019-01-24 20:00:31.000 generated a new notification event.
Event: Stale and Bad Data 2019-01-24 .
Outlook 2013

ITEMS: 6 UNREAD: 1 ALL FOLDERS ARE UP TO DATE. CONNECTED TO: MICROSOFT EXCHANGE 100%

Now let's start the OPC Interface from PIINT01 to get things back to where they were.



Interface: opcint1 -> PISRV01
Type: OPCInt OPC
Description:
Versions: opcint.exe version 2.7.0.22 UniInt version 4.7.0.15

PI Data server Connection Status
PISRV01.PISCHOOL.IN
Writeable

General
OPCInt
Service
UniInt
Failover
Health Points
Performance Counters
Performance Points
PI SDK
Disconnected Startup
Debug
IO Rate
Interface Status

General
Point Source: OPC
OPC
Interface ID: 1

Scan Classes

Scan Frequency	Scan Class #
✓ 00:00:10	1
✓ 00:00:01	2

PI Host Information
Server/Collective: PISRV01
SDK Member: PISRV01
API Hostname: PISRV01
User: piadmin | piadmins | PIWorld
Type: Primary - PI3
Version: 3.4.430.460
Port: 5450
Buffering Status: On

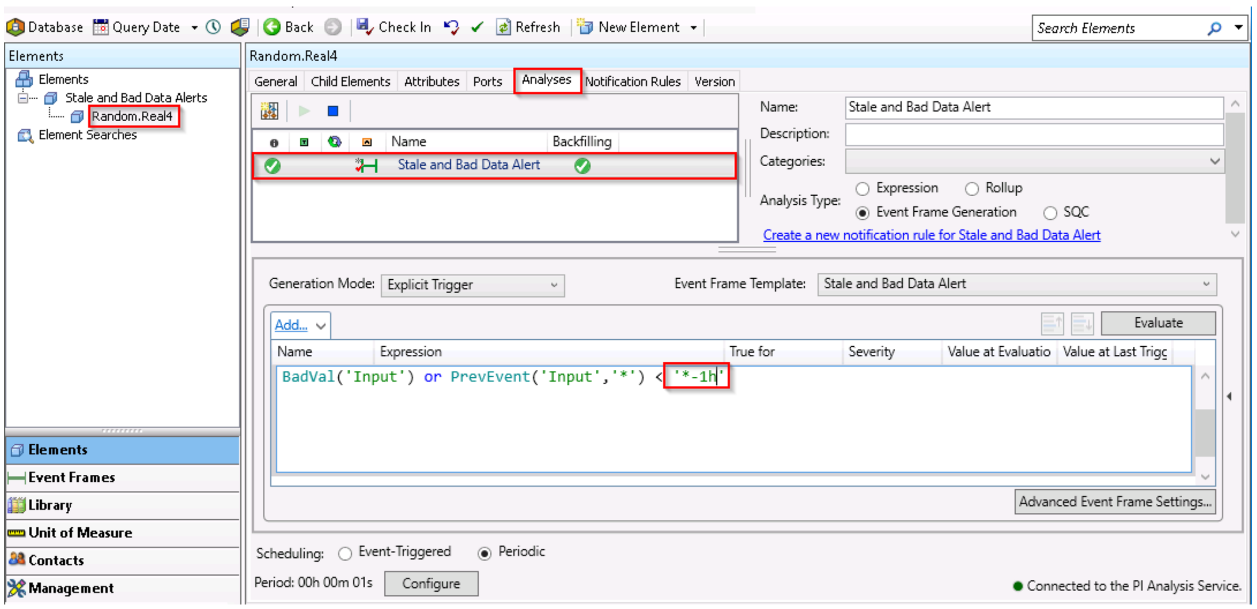
Interface Installation Path
C:\Program Files (x86)\PIPC\Interfaces\OPCInt\
Interface Batch Filename
opcint1.bat

Ready Stopped opcint1 - Installed

Let's change the triggering logic so that a Notification is only sent when data is stale for an hour.

Go back to PI System Explorer on PICLIENT01. With the Random.Real4 Element selected go back to the Analyses tab.

Change 10s to 1h and Check in.



Name: Stale and Bad Data Alert
Description:
Categories:
Analysis Type: Expression Rollup [✓ Event Frame Generation] SQC
Create a new notification rule for Stale and Bad Data Alert

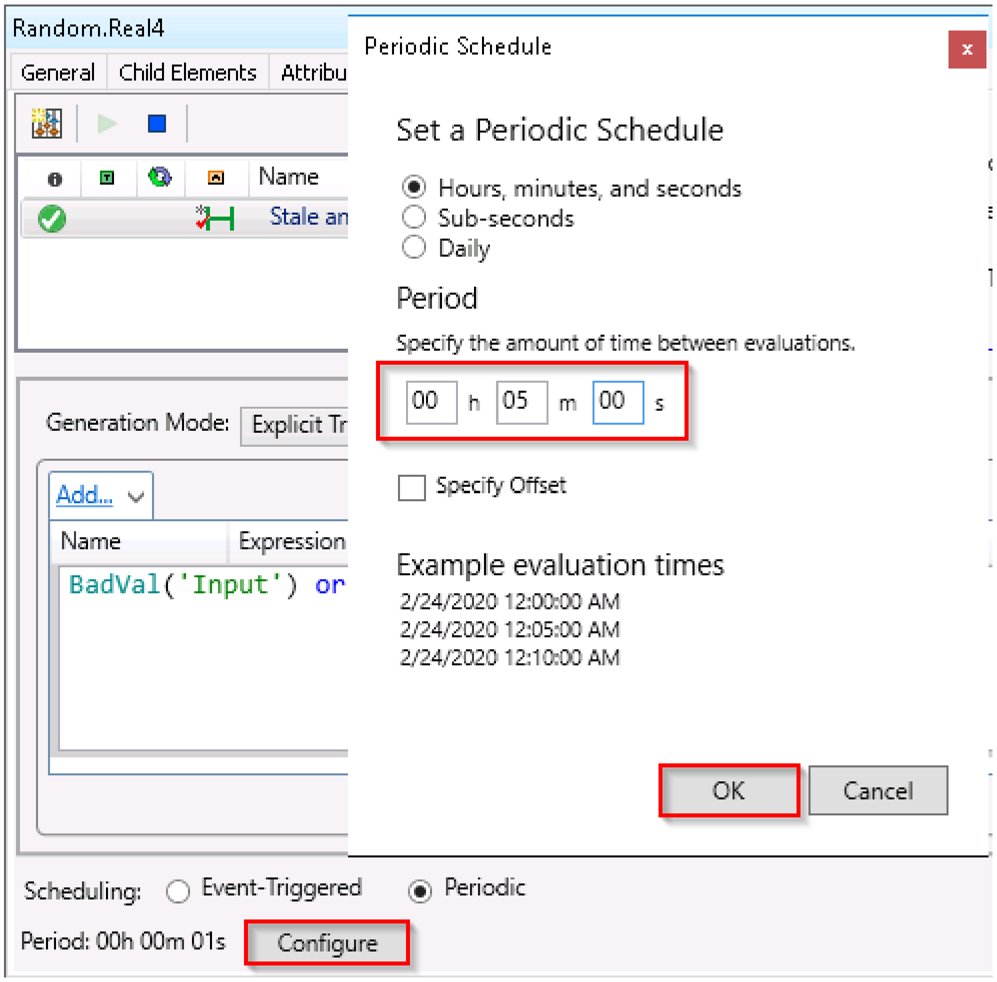
Name	Backfilling
Stale and Bad Data Alert	✓

Generation Mode: Explicit Trigger
Event Frame Template: Stale and Bad Data Alert

Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
	BadVal('Input') or PrevEvent('Input','*') < '*-1h'				

Scheduling: Event-Triggered [✓ Periodic]
Period: 00h 00m 01s
Connected to the PI Analysis Service.

Finally change the scheduling to evaluate the trigger every 5 minutes and click OK. It’s ok to have 1 Analysis that checks every second, but generally you want the longest period you can tolerate for performance reasons. Thousands of Analyses evaluating every second could overwhelm the calculation engine and cause calculations to be skipped.



Random.Real4

General | Child Elements | Attribu

Name

Stale an

Generation Mode: Explicit Tr

Add...

Name	Expression
BadVal('Input') or	

Scheduling: ( ) Event-Triggered [✓ Periodic]

Period: 00h 00m 01s Configure

Periodic Schedule

Set a Periodic Schedule

[✓ Hours, minutes, and seconds]
( ) Sub-seconds
( ) Daily

Period

Specify the amount of time between evaluations.

00 h 05 m 00 s

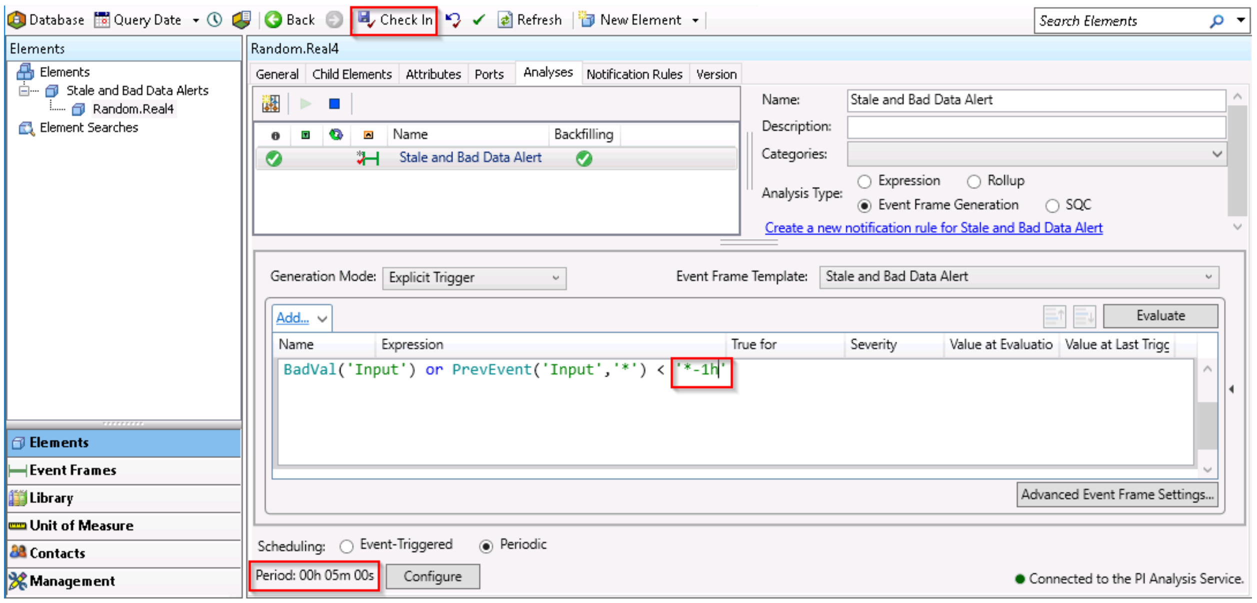
☐ Specify Offset

Example evaluation times

2/24/2020 12:00:00 AM
2/24/2020 12:05:00 AM
2/24/2020 12:10:00 AM

OK Cancel

Check in changes after confirming the edits are reflected.



Name: Stale and Bad Data Alert
Description:
Categories:
Analysis Type: Expression, Rollup, [✓ Event Frame Generation], SQC

Create a new notification rule for Stale and Bad Data Alert

Generation Mode: Explicit Trigger
Event Frame Template: Stale and Bad Data Alert

Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
	`BadVal('Input') or PrevEvent('Input','*') < '*-1h'`				

Scheduling: Event-Triggered [✓ Periodic]
Period: 00h 05m 00s
Connected to the PI Analysis Service.

Note: In the example above, you became familiar with building assets in PI AF. The naming conventions used are not recommended and are just for the purposes of this lab. For more details on naming standards for PI AF, refer to section 6.6 of the appendix.

Congratulations! That's all for today.

6. Appendix

6.1 Software Versions in this Document

Software	Version
Data Archive	2018 SP3
AF Server	2018 SP3
PI OPC interface	2.6.18.2
System Management Tools	2018 SP3
PI System Explorer	2018 SP3

6.2 Message logs

The first step in troubleshooting is always the same: check the message logs! All PI System software write messages to log files. It is therefore important to learn which log files to check, and how to read them.

PI Message Logs

Also known as the "SDK Logs", these are the logs of all applications based on the PI SDK. There is one PI Message log per computer where an SDK application is installed. The logs are managed by the PI Message Subsystem.

Applications that write to this log:

- Data Archive subsystems
- PI Interfaces (UniInt version 4.5.0.x and later)
- PI Client applications

How to access these logs:

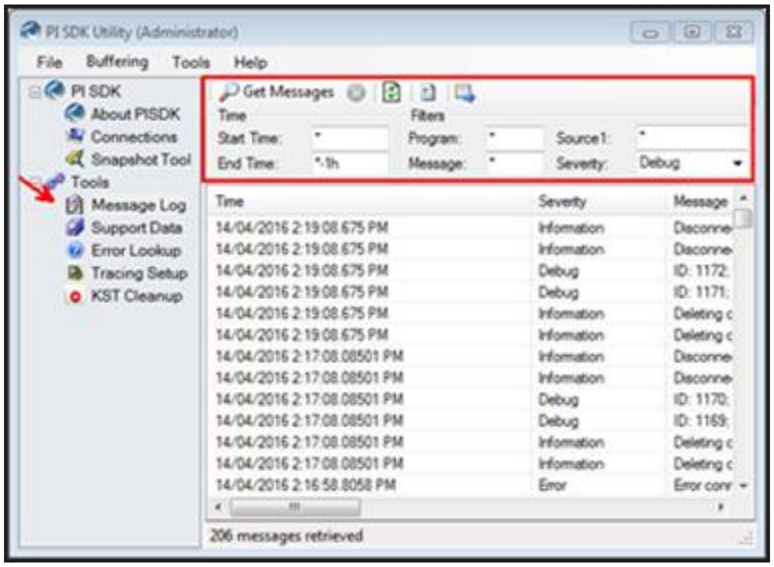
- On the Data Archive: SMT > Operations > Message Logs
- On all computers with PI-SDK 1.4.0 and greater:

Run the program "PISDKUtility"

In the left pane, select Tools > Message Logs

Set the filters to obtain messages (start time, severity, etc.)

Click on "Get Messages"



Time
Start Time: *
End Time: *-1h

Filters
Program: *
Message: *
Source1: *
Severity: Debug

Time	Severity	Message
14/04/2016 2:19:08.675 PM	Information	Disconne
14/04/2016 2:19:08.675 PM	Information	Disconne
14/04/2016 2:19:08.675 PM	Debug	ID: 1172;
14/04/2016 2:19:08.675 PM	Debug	ID: 1171;
14/04/2016 2:19:08.675 PM	Information	Deleting c
14/04/2016 2:19:08.675 PM	Information	Deleting c
14/04/2016 2:17:08.08501 PM	Information	Disconne
14/04/2016 2:17:08.08501 PM	Information	Disconne
14/04/2016 2:17:08.08501 PM	Debug	ID: 1170;
14/04/2016 2:17:08.08501 PM	Debug	ID: 1169;
14/04/2016 2:17:08.08501 PM	Information	Deleting c
14/04/2016 2:17:08.08501 PM	Information	Deleting c
14/04/2016 2:16:58.8058 PM	Error	Error conr

206 messages retrieved

- On a PI Interface node: PI ICU > Press the "View Current PI Message Log continuously" button

- On all computers, you can use the command line utility pigetmsg.exe

  Open a command prompt window

  Change the directory to pi\adm or pipc\adm

  Type **pigetmsg –f** to view the logs continuously

  For more filtering options, type **pigetmsg –?**

Event Logs

Event Logs are the centralized logs on a Windows machine. There are two different types of logs:

- **Windows Logs**: These logs include all important events on the operating system, split up into the following categories: Application, Security, Setup, System and Forwarded Events.
- **Applications and Services Logs**: These logs are specifically for applications, with each application writing to its own log.

PI System applications write to the Windows Application Log, and sometimes to a dedicated log under Applications and Services.

It's also a good idea to look at the other Windows logs (Security, System) if you suspect an issue might have occurred at the operating system level.

Applications that write to these logs:

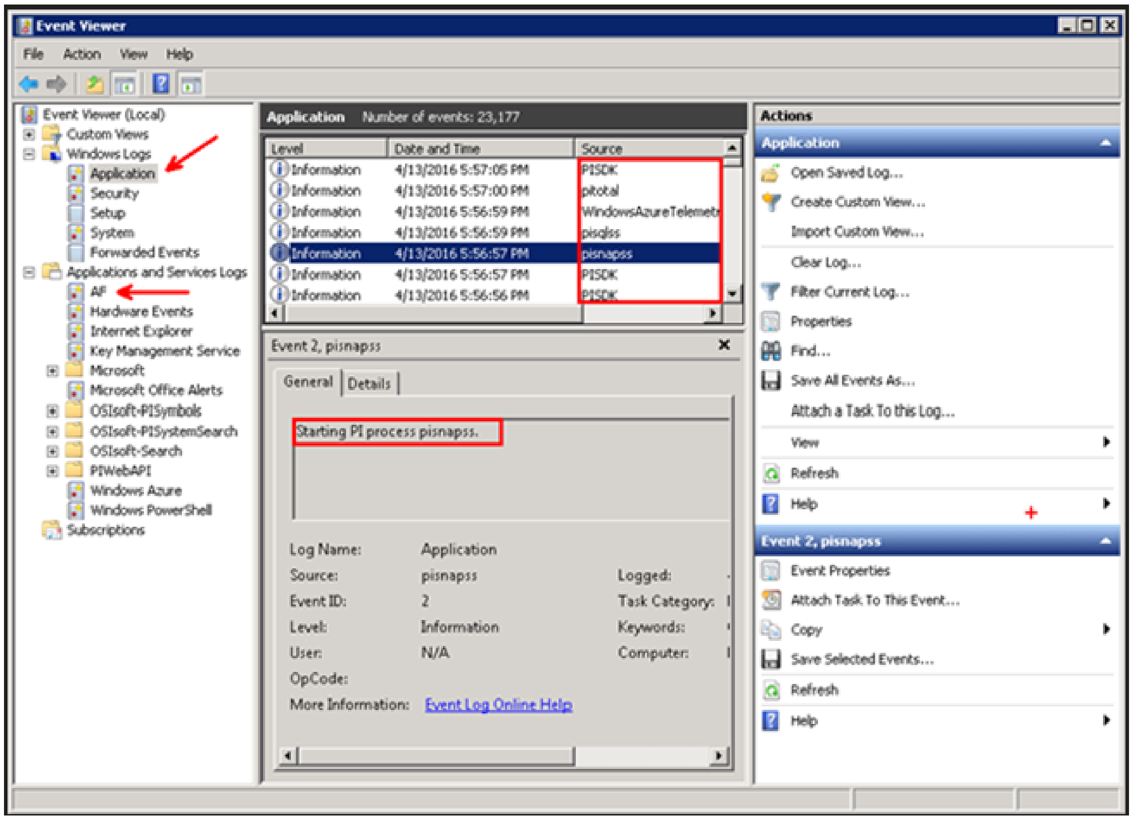
- AF Application Service
- PI Analysis Service
- Data Archive Subsystems (occasionally)

How to access these logs:

Run the application “Event Viewer”

To access the Windows Application log, browse to Windows Log > Application. Look at the “Source” column to find messages written by PI System applications. You can also use the filter functionality of Event Viewer.

To access a specific application’s log, browse to Application and Services Logs, then find the name of your application (e.g. AF)



Application Number of events: 23,177

Level	Date and Time	Source
Information	4/13/2016 5:57:05 PM	PISDK
Information	4/13/2016 5:57:00 PM	pitotal
Information	4/13/2016 5:56:59 PM	WindowsAzureTelemetr
Information	4/13/2016 5:56:59 PM	pisqlss
Information	4/13/2016 5:56:57 PM	pisnapss
Information	4/13/2016 5:56:57 PM	PISDK
Information	4/13/2016 5:56:56 PM	PISDK

Event 2, pisnapss

Starting PI process pisnapss.

Log Name: Application
Source: pisnapss
Event ID: 2
Level: Information
User: N/A
OpCode:
More Information: Event Log Online Help
Logged:
Task Category:
Keywords:
Computer:

PIPC Logs

These logs are only used by older, PI API based applications. You should only need to access these logs if you are running older software.

Applications that write to these logs:

- PI Interfaces with a UniInt version earlier than 2.5.0.x
- PI API based applications

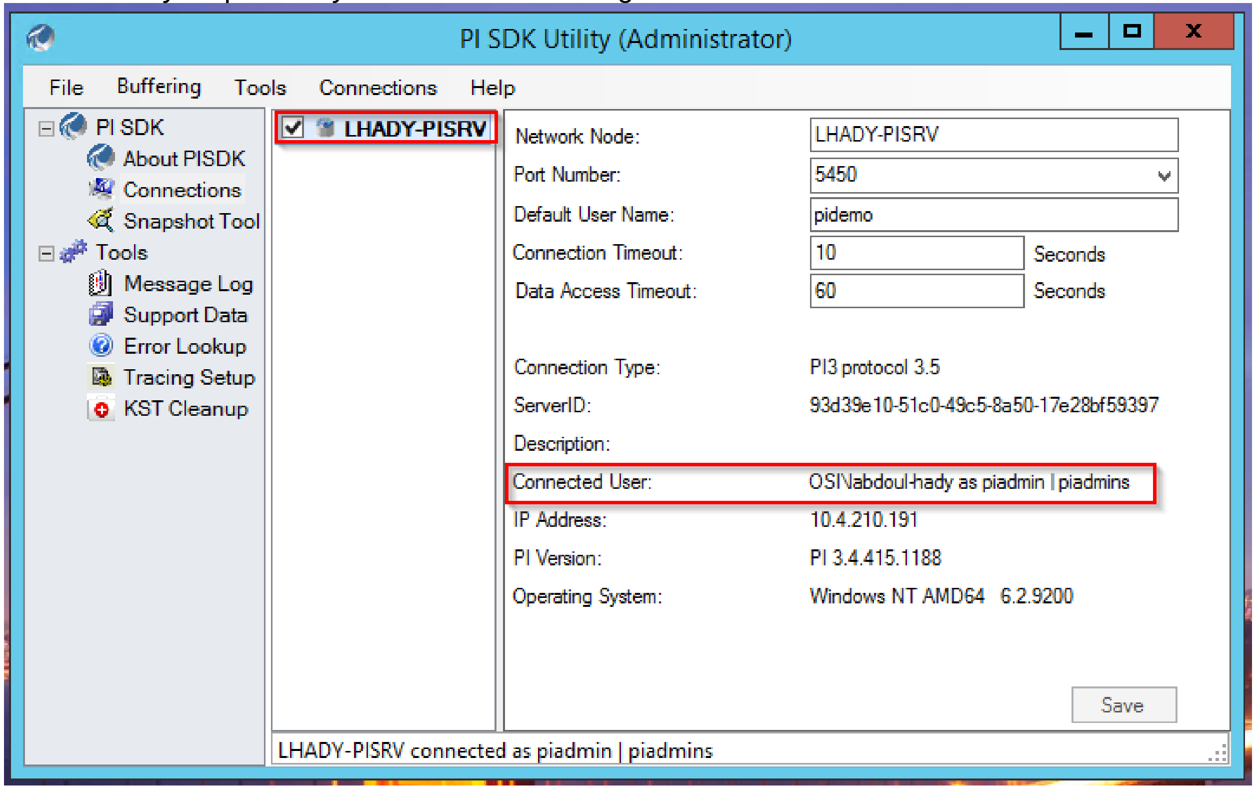
How to access these logs:

- On a PI Interface node: PI ICU > Press the “View current pipc.log continuously” button

- Open the file PIPC\dat\pipc.log

6.3 PI SDK Utility

This is a very helpful utility when troubleshooting PI SDK connections from client nodes.



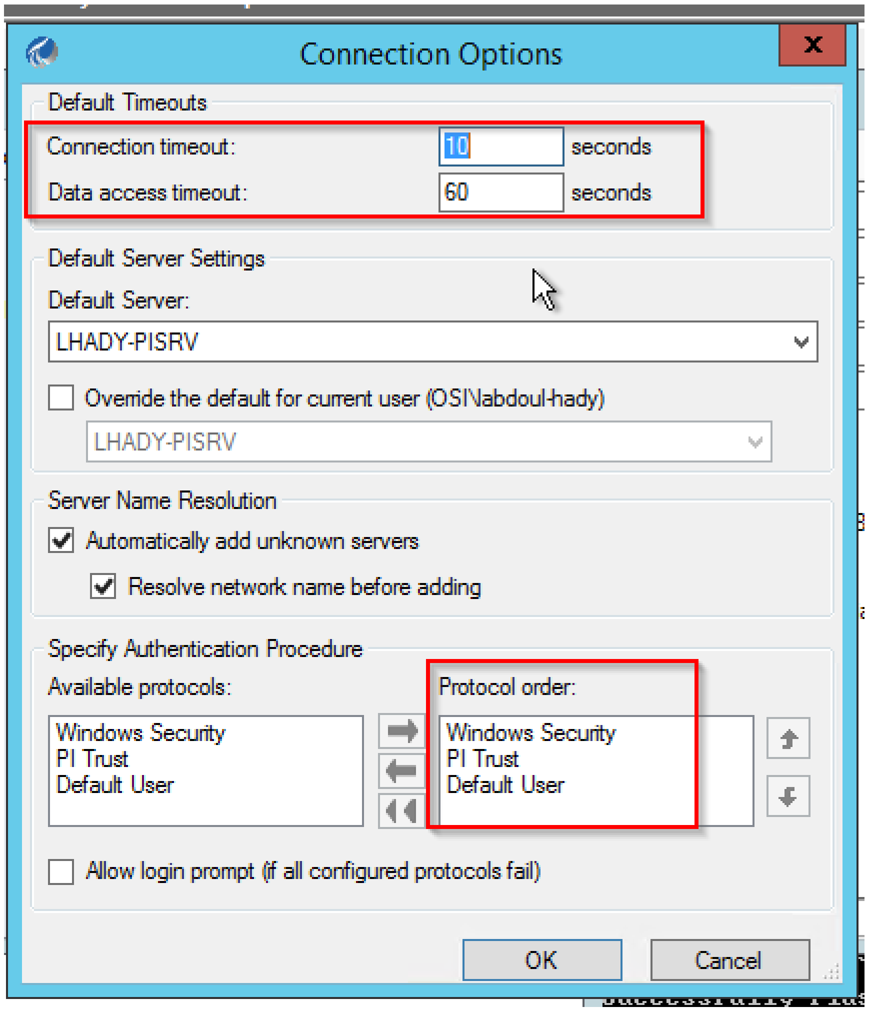
Network Node:	LHADY-PISRV	
Port Number:	5450	
Default User Name:	pidemo	
Connection Timeout:	10	Seconds
Data Access Timeout:	60	Seconds
Connection Type:	PI3 protocol 3.5	
ServerID:	93d39e10-51c0-49c5-8a50-17e28bf59397	
Description:		
Connected User:	OSI\abdoul-hady as piadmin | piadmins	
IP Address:	10.4.210.191	
PI Version:	PI 3.4.415.1188	
Operating System:	Windows NT AMD64 6.2.9200	

LHADY-PISRV connected as piadmin | piadmins

When you check the box in the middle pane next to a certain PI Data Archive name, the right pane displays the connection information. You can see the OS user you are connecting as, as well as the identity or identities that are associated to your user on the PI Data Archive.

Here are some features available from PI SDK Utility:

- Enable PI SDK Buffering (PI SDK Utility > Buffering > PI SDK Buffering Configuration….)
- Open Buffering Manager (PI SDK Utility > Buffering > Buffering Manager…)
- Customize connection options. The most common ones include:
    - Increase connection and data access timeouts
    - Change the protocol order for PI Server connections. This dictates which protocols local applications will use to connect to the PI Data Archive. If you only want Windows Integrated Security (WIS) to be used on a certain client, then you can remove PI Trust and Default User from the Protocol order. The way it is set up now, PI Mapping will be attempted first and if that fails, PI Trust will be used, and then Explicit Login. See the section below for more details on authentication protocols.



Connection timeout: 10 seconds
Data access timeout: 60 seconds

Default Server: LHADY-PISRV

☐ Override the default for current user (OSI\abdoul-hady)
LHADY-PISRV

☑ Automatically add unknown servers
☑ Resolve network name before adding

Available protocols:
Windows Security
PI Trust
Default User

Protocol order:
Windows Security
PI Trust
Default User

☐ Allow login prompt (if all configured protocols fail)

Authentication

There are three different methods of authentication on the Data Archive:

PI Mappings

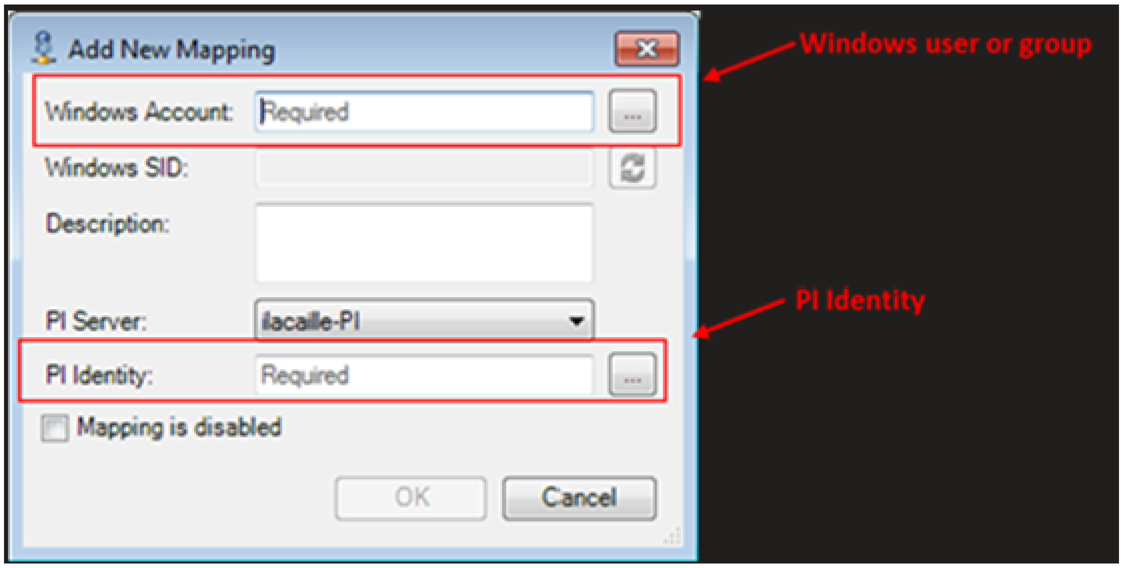
PI Mappings use *Windows Integrated Security* to authenticate users on the Data Archive. With this method, users and services connect directly to the Data Archive using their Windows account. A PI Mapping grants a Windows user or group specific rights on the Data Archive by assigning a PI Identity.

This method of authentication has several advantages:

- It is the most secure
- It enables transport security (encryption in transit) of communications with the Data Archive
- It represents the least amount of maintenance for PI System administrators
- It allows users to connect directly with their Windows accounts

The recommended strategy for using PI Mappings is to create a Windows Group for each level of authentication needed on the Data Archive (e.g. one group for Read-Only users, one group for PI System Administrators, etc.), then assign a unique PI Identity to each one of these groups.

PI Mappings are created from System Management Tools, from Security > Mappings & Trusts > Mappings Tab, by pressing the New button. This will open the Add New Mapping Window



The following conditions must be true in order to use PI Mappings:

- The application must connect with **PI AFSDK (any version), PI SDK version 1.3.6 or later or the PI API for Windows Integrated Security (version 2.0.1.35 and later, released in 2016)**
- The connecting application is running on a Windows operating system

In the event that these conditions cannot be met, a PI Trust should be used for authentication.

PI Trusts

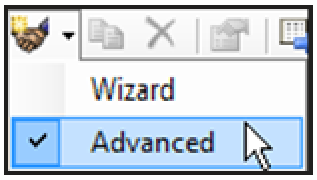
PI Trusts should NOT be used unless it is not possible to authenticate using Windows Integrated Security. The most common scenario is:

- PI Interfaces and other applications running on non-Windows Operating Systems

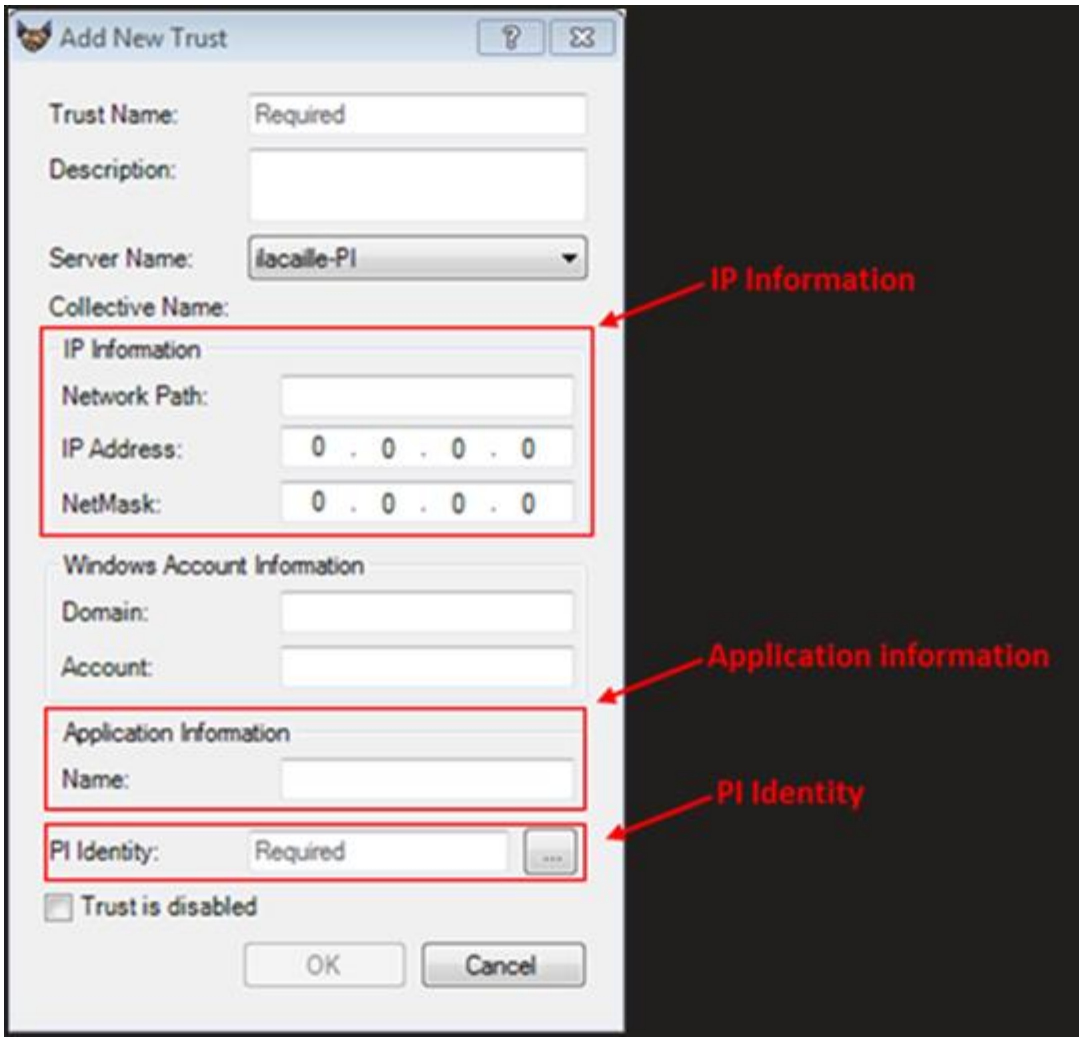
Note: Prior to 2016 release of the PI API for Windows Integrated Security, any applications using the PI API, such as PI Interfaces, could not use PI Mappings. Now, almost all PI Interface nodes can be upgraded to the new security model, regardless domain or workgroup configuration. For more information, see KB00354 - Supported Windows Security Configurations in Domains and Workgroups for the PI Data Archive

The PI Trust authentication method work by comparing the connection credentials of the connecting application to the credentials saved in PI Trusts. If the credentials match, the connection is allowed. No login is required by the application.

PI Trusts are created from System Management Tools, from Security > Mappings & Trusts > Trusts Tab, by pressing the arrow next to the New… button and selecting the advanced option:



This will open the Add New Trust Window.



Trust Name: Required

Description:

Server Name: ilacaille-PI

Collective Name:

IP Information

Network Path:

IP Address: 0 . 0 . 0 . 0

NetMask: 0 . 0 . 0 . 0

Windows Account Information

Domain:

Account:

Application Information

Name:

PI Identity: Required

Trust is disabled

It is not necessary to fill in all of the information in this Window. OSIsoft recommends that you fill out PI Trusts using the 2+ Trust convention. This means you need to enter the following:

- **The IP Information:**

  The Network Path (Host name or fully qualified domain name of the computer)

  <u>OR</u>

  The IP Address and a NetMask of 255.255.255.255.

- **The Application Information**

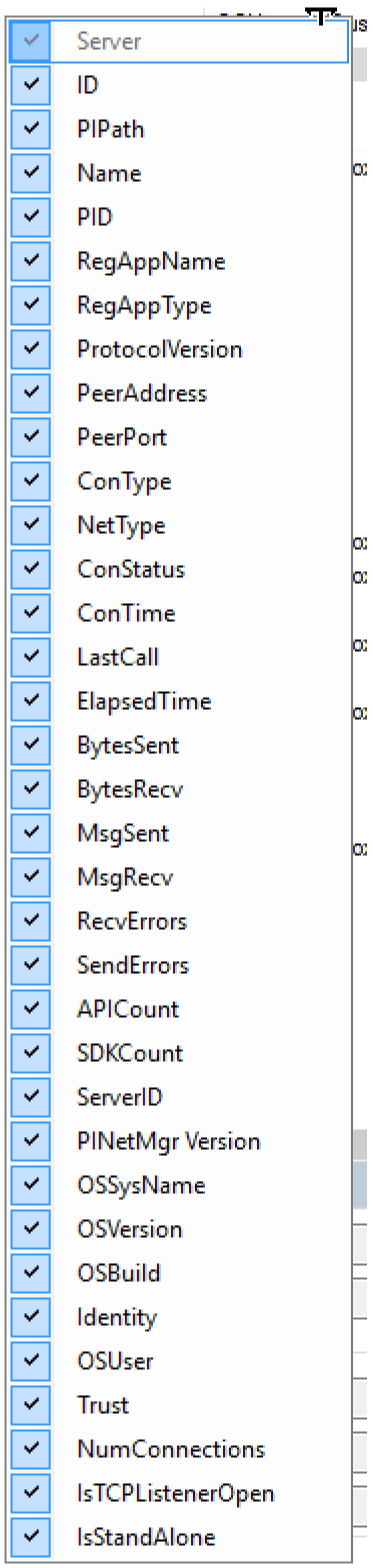
  Applications that connect through the PI API send an identifier called an application process name, or procname. This is a four-character string with an E appended. For example, the procname for the PI Perfmon interface is PIPeE.

  Explicit Login

  The final authentication method, Explicit Login, is not recommended in any scenario. It only exists for backwards compatibility purposes. Using this method, users login to the Data Archive directly using a PI User and a password.

6.4 Checking Network Manager Statistics

PI SMT's Network Manager Statistics tool (**PI SMT > Operation > Network Manager Statistics**) is a very valuable tool for PI system administrators when troubleshooting connection or permissions issues to the PI Data Archive. The tool can be used to view and export statistical information about applications and services on each connected PI Data Archive server. It gives you valuable insight regarding the location of the application, the user context that it is operating under and the user load that it is generating on the PI Data Archive. Below is a screenshot of the columns returned by the tool. All the columns are selected by default and can be removed by clicking on the blue checkmark.



- ✓ Server
- ✓ ID
- ✓ PIPath
- ✓ Name
- ✓ PID
- ✓ RegAppName
- ✓ RegAppType
- ✓ ProtocolVersion
- ✓ PeerAddress
- ✓ PeerPort
- ✓ ConType
- ✓ NetType
- ✓ ConStatus
- ✓ ConTime
- ✓ LastCall
- ✓ ElapsedTime
- ✓ BytesSent
- ✓ BytesRecv
- ✓ MsgSent
- ✓ MsgRecv
- ✓ RecvErrors
- ✓ SendErrors
- ✓ APICount
- ✓ SDKCount
- ✓ ServerID
- ✓ PINetMgr Version
- ✓ OSSysName
- ✓ OSVersion
- ✓ OSBuild
- ✓ Identity
- ✓ OSUser
- ✓ Trust
- ✓ NumConnections
- ✓ IsTCPListenerOpen
- ✓ IsStandAlone

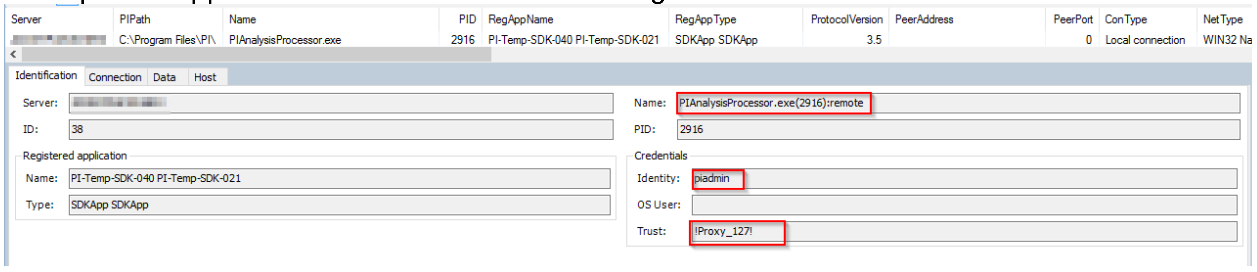
Important fields to note are:

Field	Indicates
ID	PINetmgr's identifier for the connection
ProtocolVersion	The PInet protocol version being used
NetType	Whether the connection uses TCP/IP or the named pipe
User	The identities associated with the connection
LastCall	The time of the last exchange between server and client
BytesSent	Total data sent back to the connected application
BytesRecv	Total data received from the connected application

To export the statistics file,

1. Right-click on the list and select **Export**.
2. Select the location to save the **Network Manager Statistics** list.
3. Click **Save**.

Examples of applications shown in Network Manager Statistics



Server	PIPath	Name	PID	RegAppName	RegAppType	ProtocolVersion	PeerAddress	PeerPort	ConType	NetType
[redacted]	C:\Program Files\PI\	PIAnalysisProcessor.exe	2916	PI-Temp-SDK-040 PI-Temp-SDK-021	SDKApp SDKApp	3.5		0	Local connection	WIN32 Na

Identification | Connection | Data | Host

Server: [redacted]
ID: 38
Registered application
Name: PI-Temp-SDK-040 PI-Temp-SDK-021
Type: SDKApp SDKApp

Name: PIAnalysisProcessor.exe(2916):remote
PID: 2916
Credentials
Identity: piadmin
OS User:
Trust: !Proxy_127!

From the screenshot above, we can conclude the following:

- The application connecting to the PI Data Archive is the PI Analysis Service (specifically the PIAnalysisProcessor process of the analysis server which is one of three processes that gets spun up when the PI Analysis service is started)
- The PI Analysis Service is not running under a domain service account. This is because the OS User field is null.
- The authentication protocol used when the PI Analysis Service is connecting to the PI Data Archive is PI Trusts (See section 6.3 above for the discussion on authentication protocols for PI Data Archive). The trust name is "!Proxy_127!"and is associated with the piadmin identity (identity with the highest privileges on the PI Data Archive).

Examples of when this tool is useful:

- You want to know which identity and trust your PI API connections are coming in as
- You want to check the connection ID a certain application is using to connect to the PI Data Archive
- You want to check user credentials used when a certain PI Client connection is made (PI ProcessBook, PI System Explorer, etc)

6.5 PI System Ports

Which firewall ports should be opened for the PI Server or PI Data Archive to properly function?

Functionality	Remote Application	Protocol	Port	Direction*	Local Application	Service
Client connections to PI Data Archive	All client applications	TCP	5450	Inbound	PI Data Archive	PI Network Manager
SPN registration (PI Mappings)	Domain Controller	TCP/UDP	135	Outbound	PI Data Archive	PI Network Manager
Kerberos (PI Mappings)	Key Distribution Center	TCP/UDP	88	Outbound	PI Client	PI Network Manager
NTLM (PI Mappings)	Domain Controller	TCP/UDP	Dynamic	Outbound	PI Data Archive	PI Network Manager
IP/Host lookup (PI Trust)	DNS	UDP	53	Outbound	PI Data Archive	PI Network Manager
Domain/OSUser lookup (PI Trust)	Domain Controller	TCP/UDP	Dynamic	Outbound	PI Data Archive	PI Network Manager

*The direction is in relation to the machine on which the PI Data Archive is running. For example, Outbound means that the PI Data Archive node firewall rule needs to allow traffic to leave the PI Data Archive in order to direct it towards the remote node, whereas Inbound means that the PI Data Archive node firewall rule needs to accept incoming connections from the remote node.

Please search article "Which firewall ports should be opened for the PI Server / PI Data Archive?" on the customer portal for more background on this.

Which firewall ports should be open for the PI AF Server to properly function?

For full feature functionality of the PI AF Server, the following ports are required:

Infrastructure

Technology	Functionality	Remote Application	Protocol	Port	Direction	Local Application	Service
SQL Server	Hosting the PIFD database	PI AF Server	TCP	1433*	Inbound	SQL Server	SQL Server
SQL Server Browser	Remotely identify SQL instances	PI AF Server	UDP	1434	Inbound	SQL Server	SQL Server
SPN registration	Authentication	Domain Controller	TCP/UDP	135	Outbound	PI AF Server	PI AF Application Service
Kerberos	Authentication	Key Distribution Center	TCP/UDP	88	Outbound	PI Client	PI Network Manager
NTLM	Authentication	Domain Controller	TCP/UDP	Dynamic	Outbound	PI AF Server	PI AF Application Service
SMB	Search for local accounts to manage mappings remotely through AF Client	Domain Controller	TCP	445**	Outbound	PI Client	PI Network Manager
LDAP	Cross-domain authentication	Domain Controller	TCP/UDP	389***	Outbound	PI Client	PI Network Manager

* Can be configured to use a dynamic port.

** You could also use port 139 instead if NetBIOS is enabled. SMB is used when the Windows user group/object picker is opened. The SMB connection directs to the node requested by the user; as such this is not a hard requirement to browse users on the AF Server (except when setting up permission for AF Link).

*** Allowing outbound connections from the client through Port 389 is necessary if the PI AF Client is on one domain and the PI AF Server on a different domain. This is necessary for the client to contact the domain controller on the remote domain through the LDAP protocol

Base Functionality

Functionality	Remote Application	Protocol	Port	Direction	Local Application	Service
PI AF SDK Client connections to PI AF Server	PI AF SDK Clients (eg PI System Explorer, PI Coresight, etc)	TCP	5457	Inbound	PI AF Server	PI AF Application Service
PI SQL for AF Client connections to PI AF Server	PI SQL for AF Clients (eg PI OLEDB Enterprise, PI Coresight 1.x, etc)	TCP	5459	Inbound	PI AF Server	PI AF Application Service

Additional Functionality

Functionality	Remote Application	Protocol	Port	Direction	Local Application	Service
Client connection to PI Analysis Service	PI System Explorer	TCP	5463	Inbound	PI Analysis Service Server	PI Analysis Service
PI AF Collective Creation	PI AF Server (Primary)	TCP	5457	Inbound	PI AF Server (Secondary)	PI AF Application Service
PI AF Collective Creation and Operation	SQL Agent	TCP	1433	Inbound	SQL Server Subscriber	SQL Server Agent
PI Notifications Services and Scheduler	See What ports need to remain open in a Firewall for PI Notifications	TCP	See What ports need to remain open in a Firewall for PI Notifications	See What ports need to remain open in a Firewall for PI Notifications	See What ports need to remain open in a Firewall for PI Notifications	

Please search for the “Firewall Port Requirements” article for links to articles that list port requirements for the PI System to be functional.

6.6 Standards for Naming Assets in PI AF

PI AF allows you to enhance your PI System data to present the PI Tags in a more user-friendly manner. Some business impacts include but are not limited to:

1. Asset health: Enable condition-based maintenance (rather than calendar or run-to-failure maintenance), root cause analysis, predictive insight, and smart expense allocations.
2. Energy Efficiency: Reduce energy, water, and raw material costs. Establish baselines for asset/ process/ site-specific consumption, and identify underperforming assets
3. Process productivity: Avoid unexpected downtime, process defects, monitor asset performance, standardize best practices and maximize efficiency. Automatically report standard KPIs to measure performance.
4. Quality tracking: Reveal process variability or defects affecting product quality to improve consistency and quality across batches, shifts and sites. Traceability for reporting and audits.
5. Regulatory reporting: Automatically and accurately roll data up to create reports without manual data entry or analysis. Environmental, safety, business, and compliance reporting.
6. Safety and Security: Identify operational risks, before they happen. Alert operators, plant and business managers of conditions that could harm employees. Cyber security of the control network.

It is very important to know the business impact that is important to you when designing a PI AF database. This allows you to picture the hierarchy that you want to design for your users. Based on this hierarchy, you can come up with tag naming conventions that will make it easier to import PI Points into PI AF and later when sizing up.

The naming of PI AF elements and attributes is an important consideration when building a PI AF implementation. It is a best practice to model the contents of PI AF on industry wide standards when such a standard exists and is possible to implement. Even when an industry standard is not available, a companywide internal standard for naming conventions of PI AF elements and attributes should be adopted. Some of the common industry standards that include or reference naming conventions that are relevant to PI AF include CIM (Common Interface Model), the MultiSpeak Specification, and ANSI/ISA S95. This helps in interoperability, exchange of data between applications, and ease of use across sites for users. If the PI tags are also named according to an industry standard or internal standard, naming the elements and attributes according to the standard can make it easier to automatically configure those elements and attributes. Query tools such as PI OLEDB Enterprise can take advantage of naming conventions by building generic SQL queries at the PI AF template level rather than individual elements and attributes. Following a naming standard can be particularly advantageous for customers who already have the PI System deployed across multiple sites, especially if those sites acquired the PI System individually or were acquired from other organizations. In situations like this, even if the PI tags are named differently across sites, PI AF can be used to present the data to the users in a consistent fashion across all sites by following a naming standard within PI AF. This will abstract the actual PI tag names away from users, and get them thinking about how to use the data rather than how to find it.

For ideas on naming conventions, you can explore the Asset Based PI Example Kits on PI Square.

6.7 Where to look for answers

So, you found an error message, now what? There are a few resources you can use to translate that message and find your solution:

Search for a Solution on the customer portal (https://my.osisoft.com)

This solution search crawls all of our online resources, including product documentation, Knowledge Base (KB) articles, PI Square forum discussions, Known Issues, and more.

Search the PI Live Library (https://livelibrary.osisoft.com)

This is an online repository of OSIsoft documentation. It contains all of the up to date administration and user guides for our products.

Ask the community on PISquare (https://pisquare.osisoft.com)

Contact OSIsoft Tech Support! (https://my.osisoft.com)

When contacting Technical Support, always make sure to have the following information on hand:

A clear description of the issue

Product and version information

A copy of the relevant message logs

Relevant screenshots, and if possible, steps to reproduce the issue

Urgency and Impact of this case

Your PI Server Serial Number (SMT > Operation > Licensing > InstallatonID)