

PI World 2019 Lab

Beginning Administration and Management of the PI System



OSIsoft, LLC
1600 Alvarado Street
San Leandro, CA 94577

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of OSIsoft, LLC.

OSIsoft, the OSIsoft logo and logotype, Managed PI, OSIsoft Advanced Services, OSIsoft Cloud Services, OSIsoft Connected Services, OSIsoft EDS, PI ACE, PI Advanced Computing Engine, PI AF SDK, PI API, PI Asset Framework, PI Audit Viewer, PI Builder, PI Cloud Connect, PI Connectors, PI Data Archive, PI DataLink, PI DataLink Server, PI Developers Club, PI Integrator for Business Analytics, PI Interfaces, PI JDBC Driver, PI Manual Logger, PI Notifications, PI ODBC Driver, PI OLEDB Enterprise, PI OLEDB Provider, PI OPC DA Server, PI OPC HDA Server, PI ProcessBook, PI SDK, PI Server, PI Square, PI System, PI System Access, PI Vision, PI Visualization Suite, PI Web API, PI WebParts, PI Web Services, RLINK and RtReports are all trademarks of OSIsoft, LLC.

All other trademarks or trade names used herein are the property of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the US Government is subject to restrictions set forth in the OSIsoft, LLC license agreement and/or as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12-212, FAR 52.227-19, or their successors, as applicable.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording or otherwise, without the written permission of OSIsoft, LLC.

Published: March 22, 2019

Table of Contents

Contents

Table of Contents	3
1. Introduction	4
1.1 Lab Overview	4
1.2 PI System Introduction	4
1.3 Lab Environment	5
2. Directed Activity – Getting Started	6
2.1 Objective	6
2.2 Tasks	6
2.3 Step-by-Step Instructions	6
3. Directed Activity – Troubleshooting BrokenTag and RetiredTag	16
3.1 Objective	16
3.2 Tasks	16
3.3 Step-by-Step Instructions	16
4. Directed Activity – The PI Collective Misconception	27
4.1 Objective	27
4.2 Tasks	27
4.3 Step-by-Step Instructions	27
5. Directed Activity – Stale and Bad Data Alerts	58
5.1 Objective	58
5.2 Tasks	58
5.3 Step-by-Step Instructions	58
Save the Date!	78

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:

- Learn tips for setting up your workstation and servers to efficiently administer your PI System.
- 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

Before PI Asset Framework (PI AF), the main back-end server was the PI Data Archive. At the time it was referred to as the PI Server, and many people still use the two terms interchangeably.

The PI Data Archive is where the PI Tags are stored and today is still the backbone of the PI System. Every PI System has a PI Data Archive. It is an absolute requirement for nearly every facet of operation in a modern PI System, even those where PI AF is the preferred point of access.

PI Tags are also commonly referred to as simply “tags”, but are also called PI Points in some literature to differentiate them from control system and instrumentation tags, such as those on an OPC Server. PI Tags contain all current and historical data that has been written to the PI Data Archive over time.

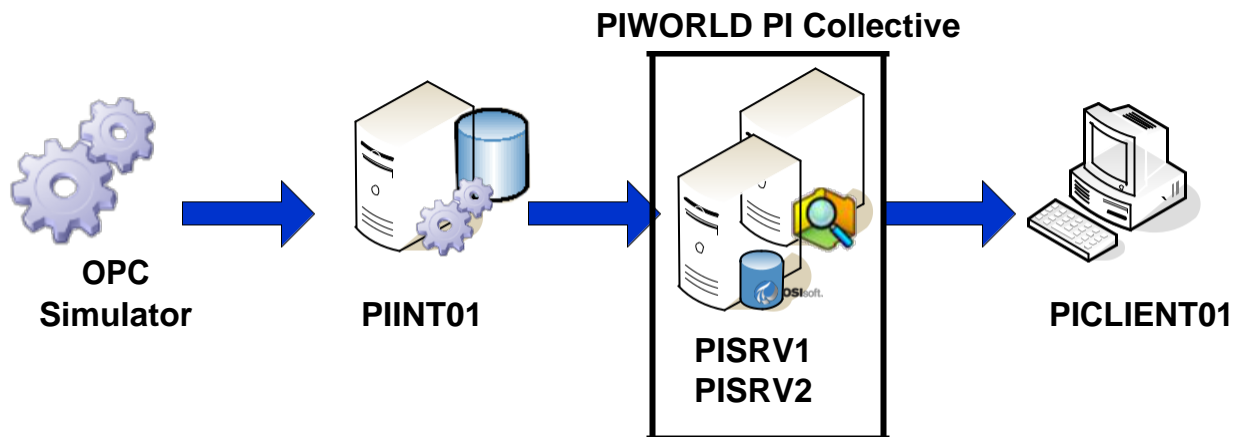
Data is written to PI Tags in many different ways, but the majority is written by products called

PI Interfaces, which come in many different flavors depending on the source of the data. PI Interfaces generally poll or receive data from a control system or other data source, and write the result to a PI Tag with a timestamp at regular intervals, for example every few seconds. The result is a current and historical record of the readings from a given data source for the entire time the PI System has been configured. For some customers 30 years of data or more has been archived.

This data collection platform forms the back-end of various applications that rely heavily on this real-time and historical process data. Various out of the box tools, data access methods, and programming libraries allow the administrators and users of the PI System to configure an array of reports and dashboards.

1.3 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 PIWORLD PI Collective. Data is stored in PI Tags which are consumed by various client applications:



1.4 Connecting to the Virtual Machines

If you haven't already, now is a good time to connect to your Virtual Learning Environment. Locate the slip of paper on your desk that contains your connection information.

Step-by-Step:

1. Open Remote Desktop on your PC
2. Enter the connection string you were provided for your Client machine and click OK
3. Log into the machine using the provided credentials

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 are now displaying a value of “Configure”.

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

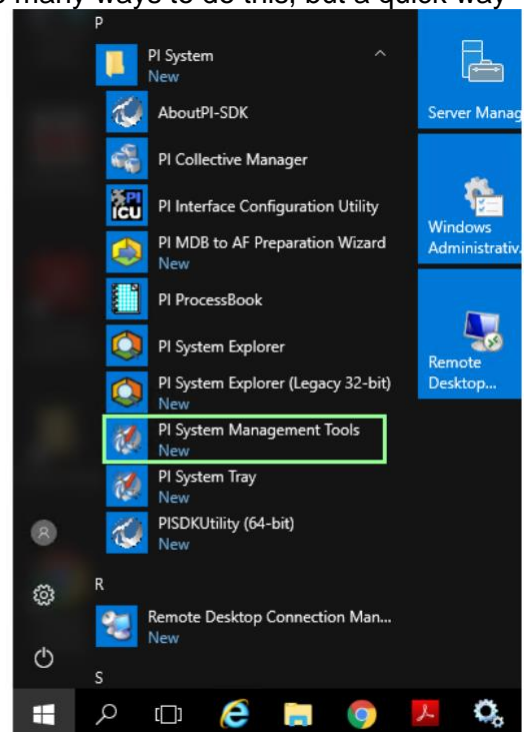
- Use the PI System Management Tools (PI SMT) Current Value plugin
- Configure Remote Desktop Connection Manager

2.3 Step-by-Step Instructions

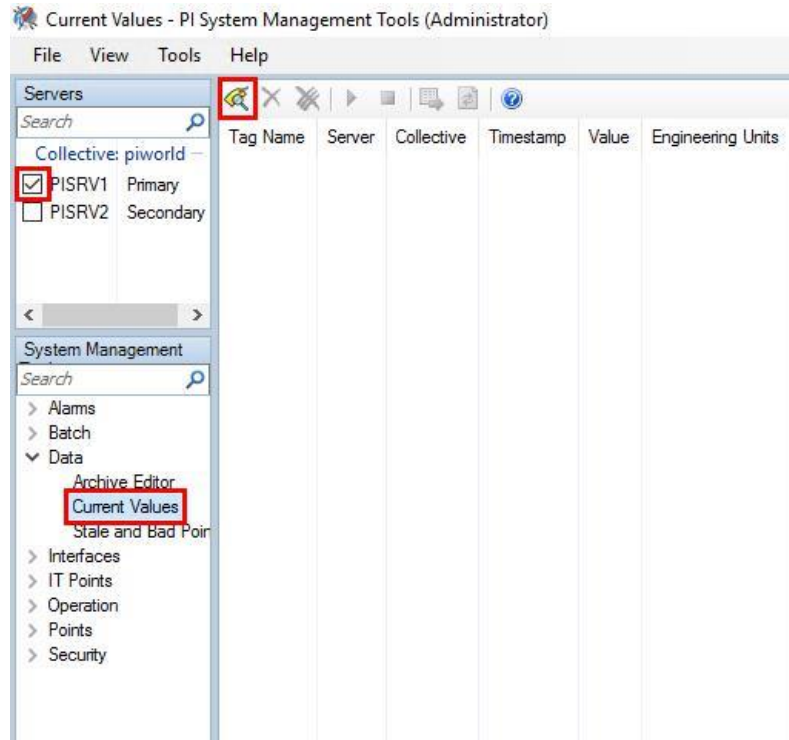
a. Rule out User Error by checking tag values

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.

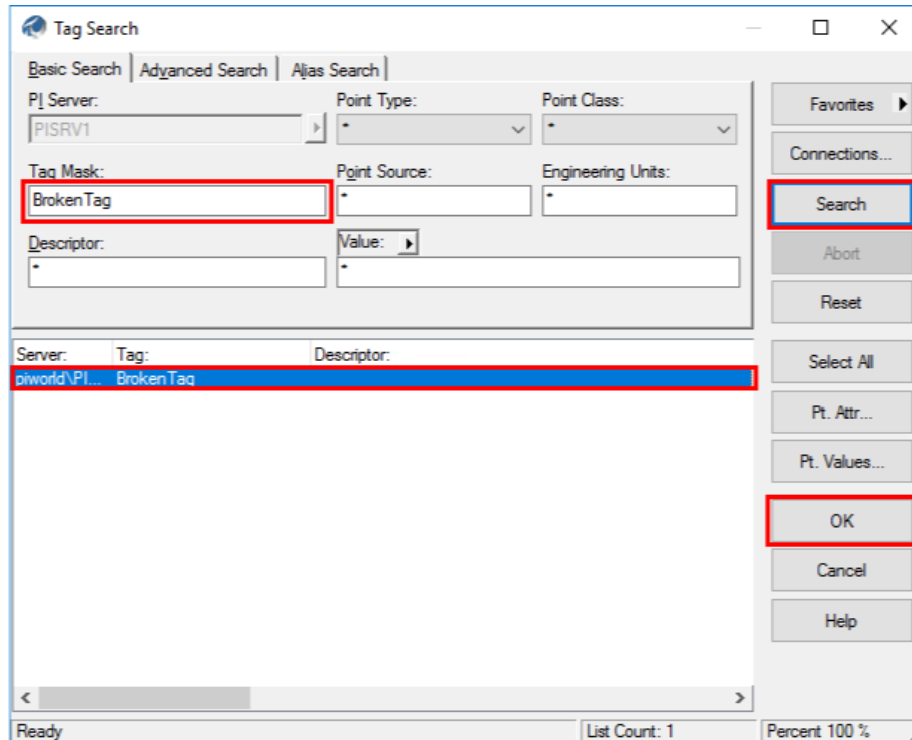
Open the Start Menu -> Expand PI System -> and run PI System Management Tools.



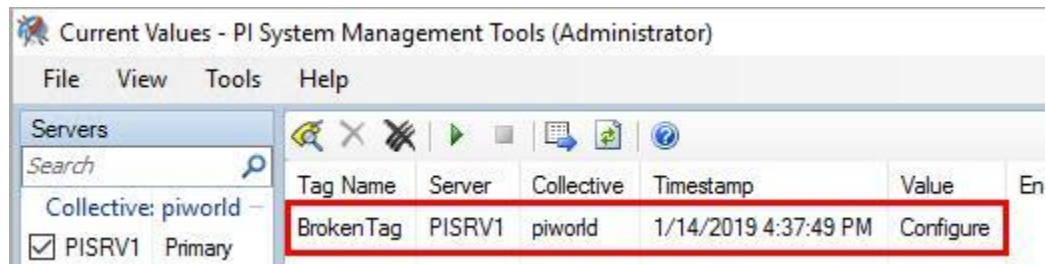
Check the box next to PISRV1 to connect to the PI Data Archive. Expand Data -> Current Values, then click the icon to search for Tags.



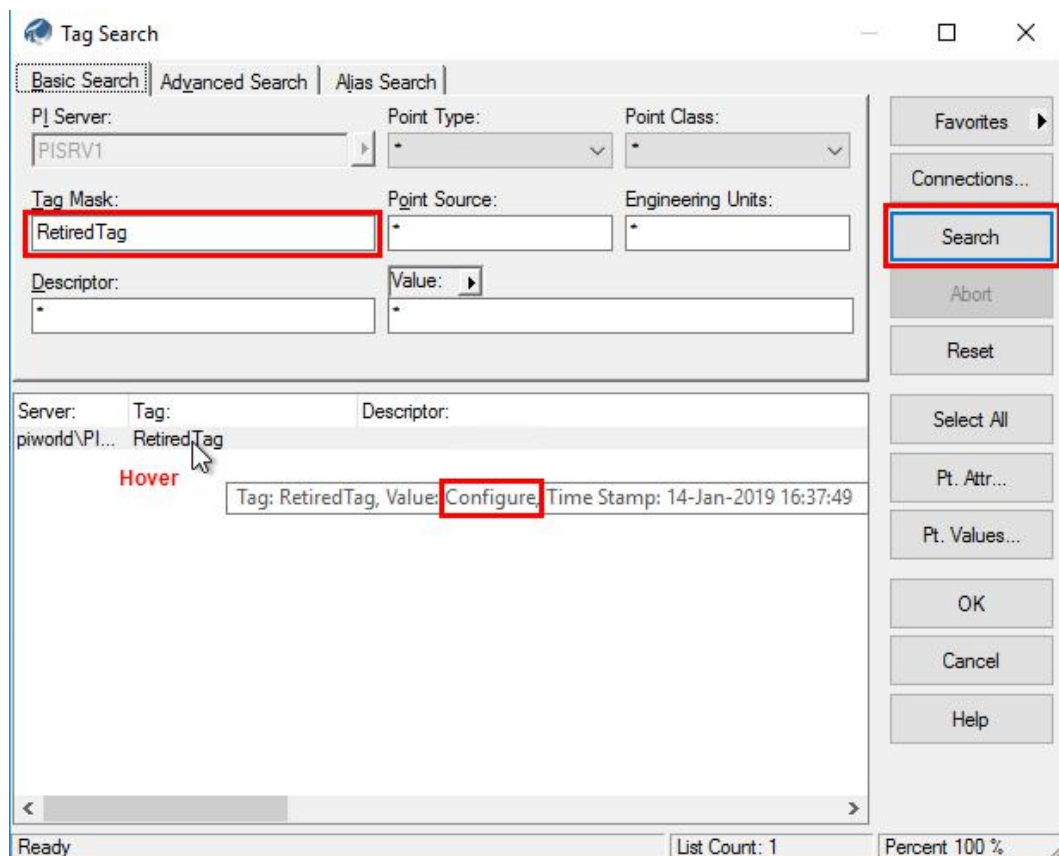
In the search dialog, enter BrokenTag, click search, then select BrokenTag in the result list and click OK.



Looks like the value is indeed Configure. The user informs you it should be a number.



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.**

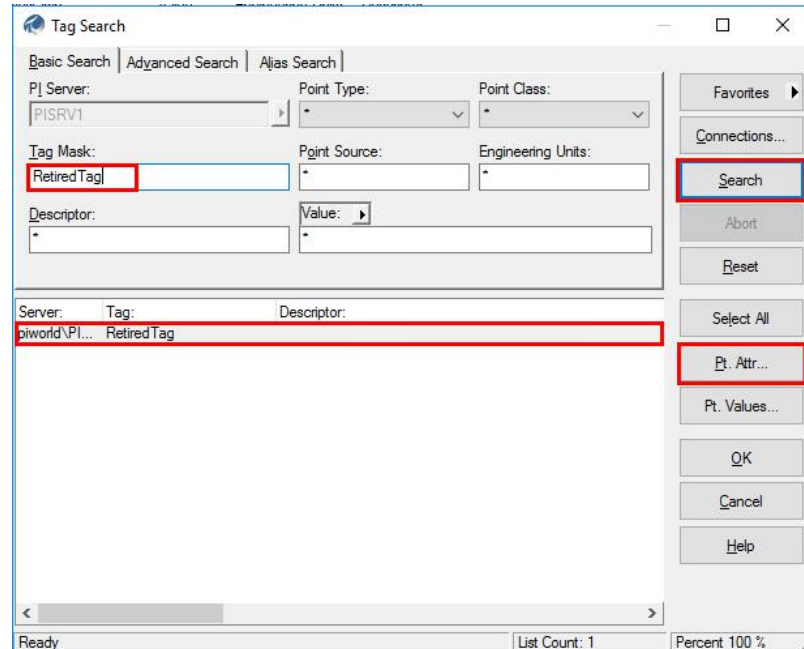


b. Identify the PI Interface Node

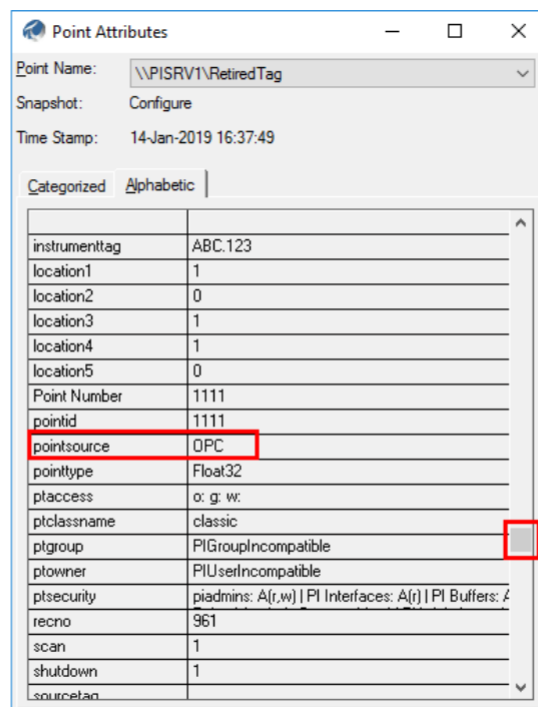
In time, you'll get a feel for how each tag gets its data, but let's say you know nothing at all about your PI System and have nobody to ask. How do you even get started tackling a bad tag? Generally, the first step is to get close to the source of the data, which often means logging in to the PI Interface 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?

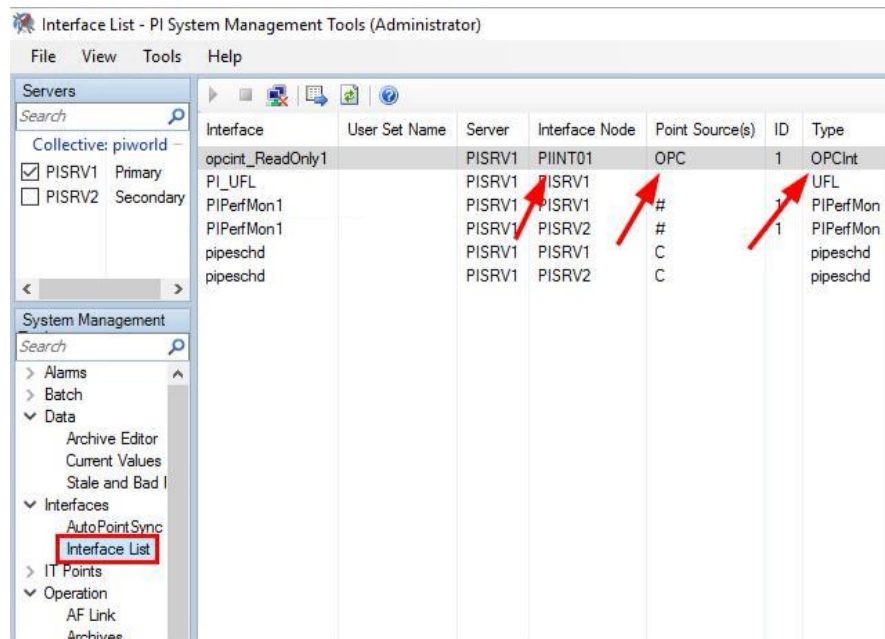
Go back to the Tag Search dialog, and search for one of the bad tags again. **Enter RetiredTag for the Tag Mask, Hit search, Click on RetiredTag, and Click Pt Attr....**



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**.



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.

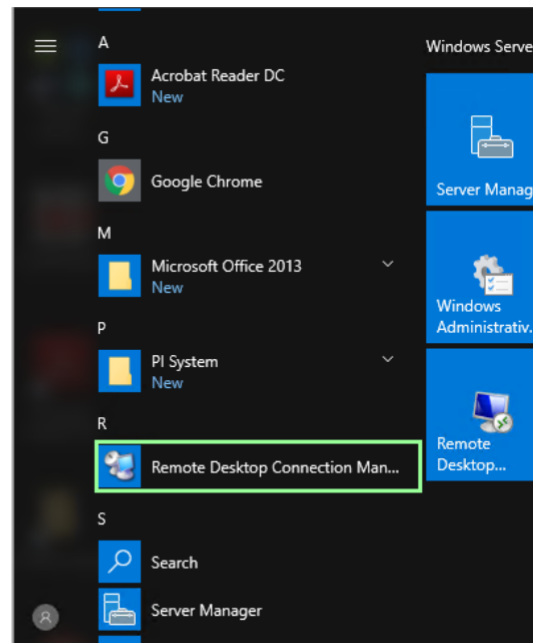


c. Set up Remote Desktop Connection Manager (RDCMan).

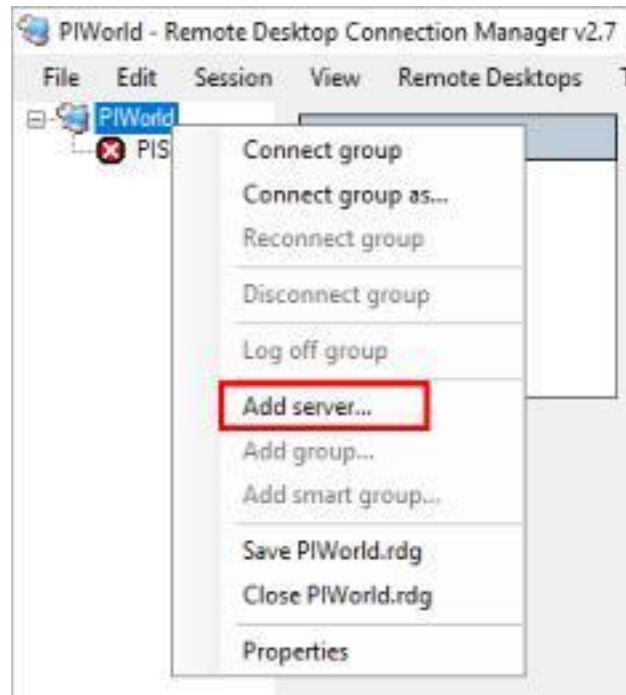
From this point, you could open up Remote Desktop and connect to PIINT01, however you will save time in the long run by using a free Microsoft tool called Remote Desktop Connection Manager (RDCMan). If you aren't using this tool (or something better) already then you've been doing it the hard way.

RDCMan has already been downloaded and installed and some initial configuration has been completed to save time during the lab. You'll have no trouble figuring it out if you have to start from scratch at your own workstation. There are many tutorials and YouTube videos available.

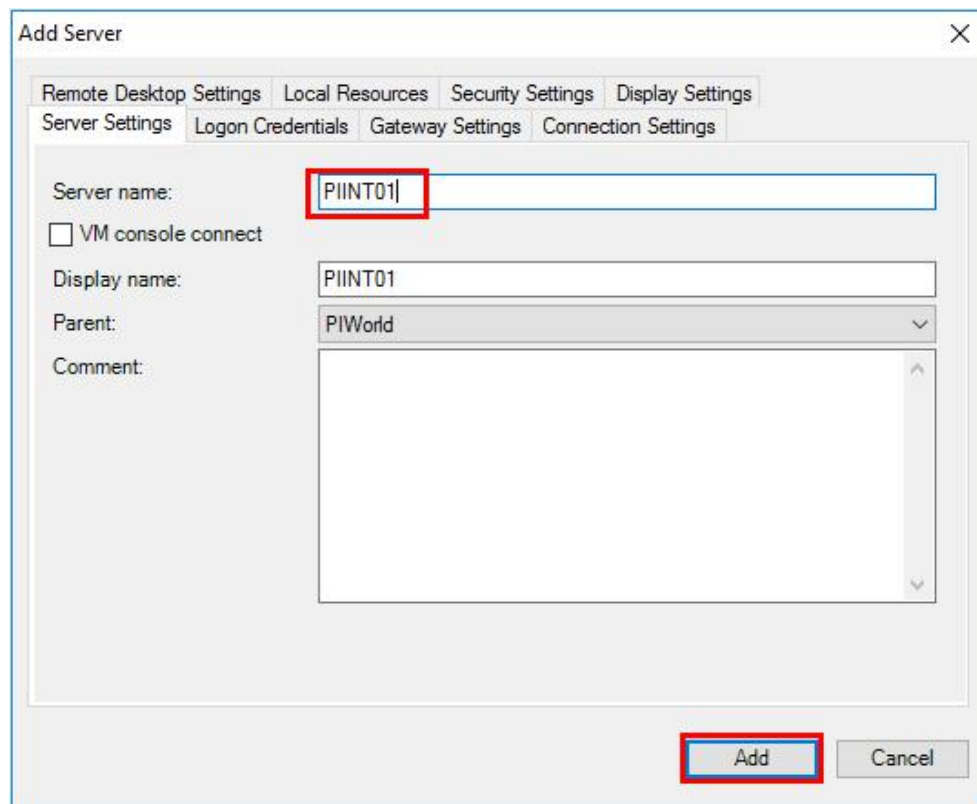
Open RDCMan from the Start Menu.



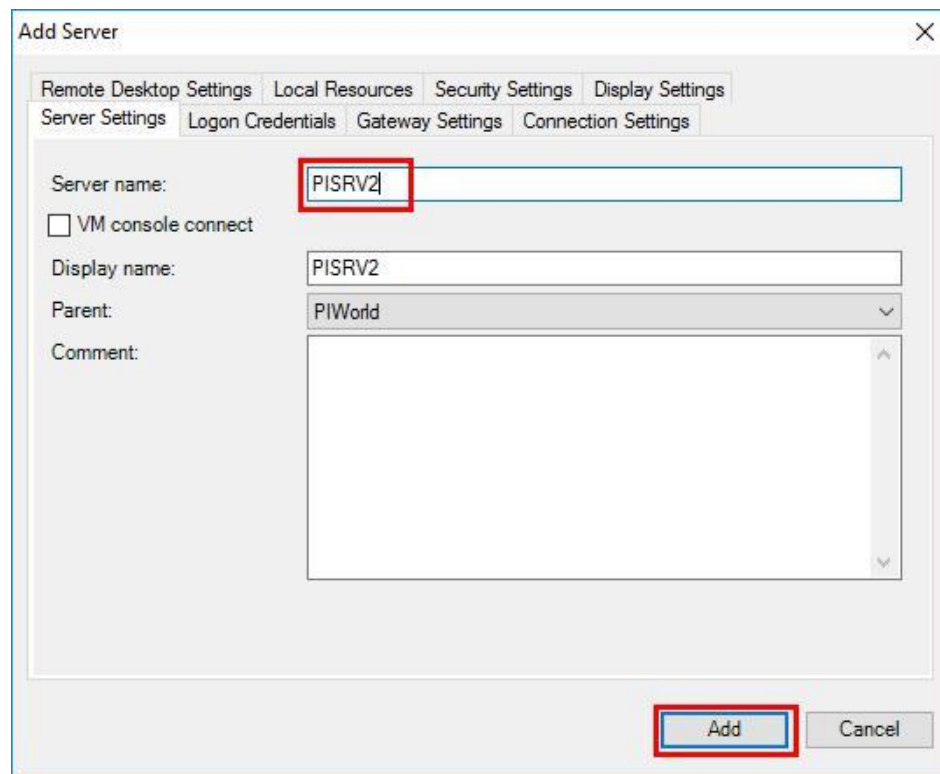
Right click on the **PIWorld** server group and select **Add Server**



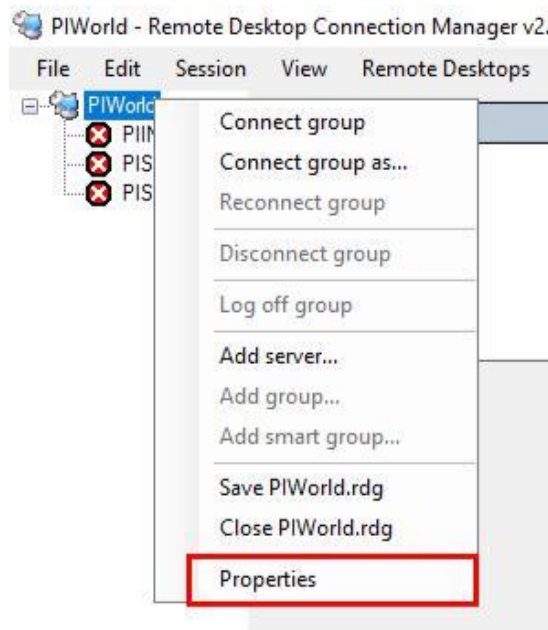
Enter **PIINT01** as the server name, which should populate the display name. **Click Add.**



While we're at it, **add PISRV2** since we may log into this machine later.



All servers in the PIWorld group inherit credentials from the group, which in turn inherits credentials from the Default settings group. New passwords are generated with each deployment, so the password needs to be updated in the Default settings group. **Right click the PIWorld group and select Properties.**



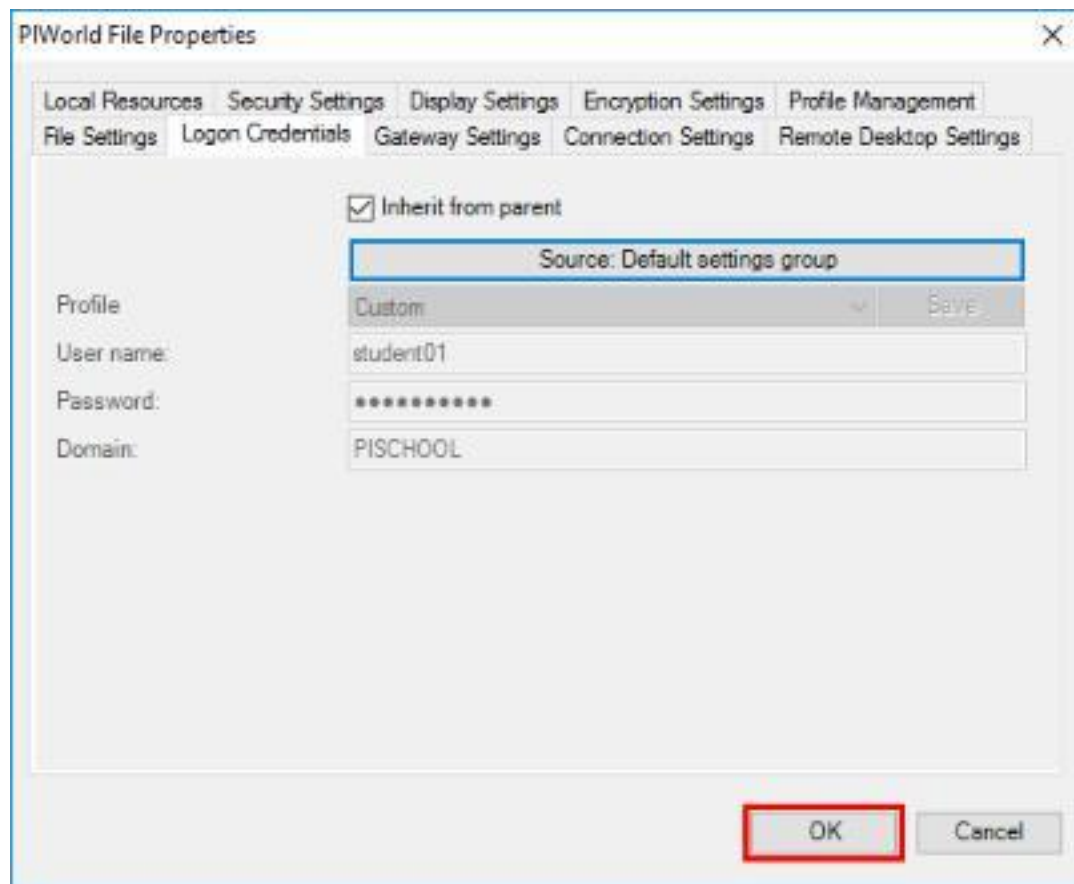
Then select the logon credentials tab and **click 'Source: Default settings group'**.

The screenshot shows the 'PIWorld File Properties' dialog box with the 'Logon Credentials' tab selected. The 'Inherit from parent' checkbox is checked. Below it, a red box highlights the text 'Source: Default settings group'. The 'Profile' dropdown is set to 'Custom'. The 'User name' field contains 'student01', the 'Password' field is masked with dots, and the 'Domain' field contains 'PISCHOOL'.

Enter the password for user **pischool\student01**. This password should have been supplied at the start of the lab.

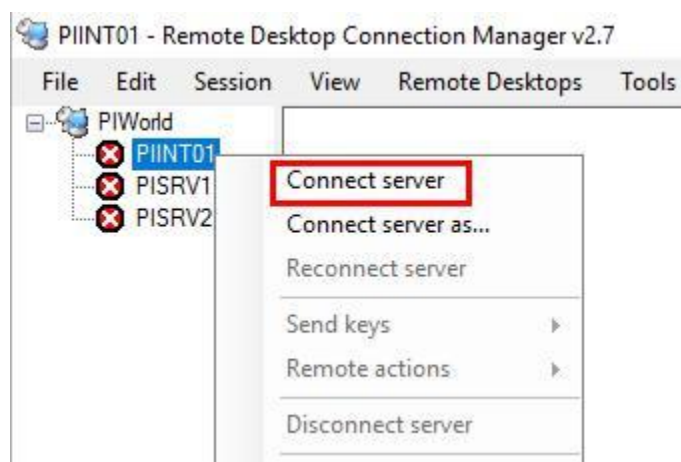
The screenshot shows the 'Default Settings Group Properties' dialog box with the 'Logon Credentials' tab selected. The 'Profile' dropdown is set to 'Custom'. The 'User name' field contains 'student01', the 'Password' field is masked with dots and highlighted with a red box, and the 'Domain' field contains 'PISCHOOL'. At the bottom, the 'OK' button is highlighted with a red box.

Finally, **Click OK in the PIWorld group properties**



Thanks to credential inheritance, setting the password only needs to be done once and not for every server.

Test the configuration by logging in to PIINT01. **Right click on PIINT01 and click 'Connect server'.** You should see the server desktop.



d. Discussion

Let's quickly review the main advantages of RDCMan over Remote Desktop:

- **The act of connecting is just faster:** Just right-click -> Connect without having to type or copy/paste the password. Right-click->Reconnect when your session times out.
- **Password Management:** It's common to use different credentials for different servers depending on which domain or network segment they exist in. Remote Desktop Connection Manager remembers IP addresses and alternate accounts for you.
- **Documentation for your PI System:** The list of all the servers you administer is right there in front of you.
- **Connection file sharing:** The .rdg file can be shared with other employees or contractors. The list of servers they might log into and all connection information is inherently stored in the file. The exception is passwords, which have to be re-entered for security reasons.

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 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

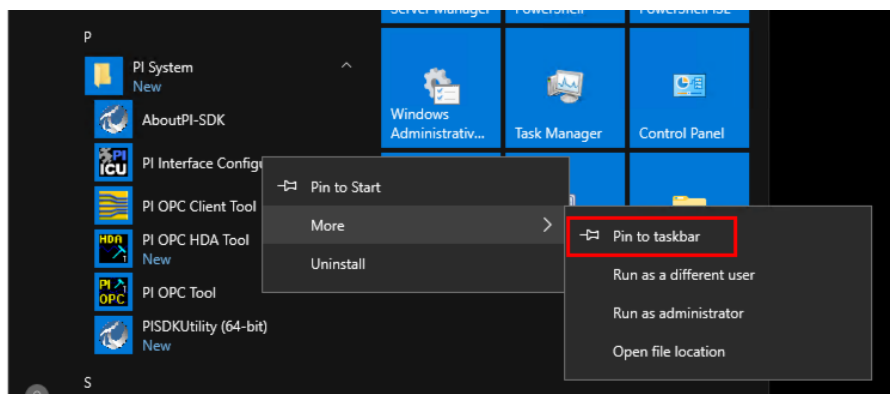
a. View the PI Interface startup logs

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, but it's much easier to restart the PI Interface service and watch the logs in real-time or look at a recent time period. Restarting the PI Interface service is generally safe and is done very frequently during troubleshooting and when adding new tags. Don't worry about breaking it.

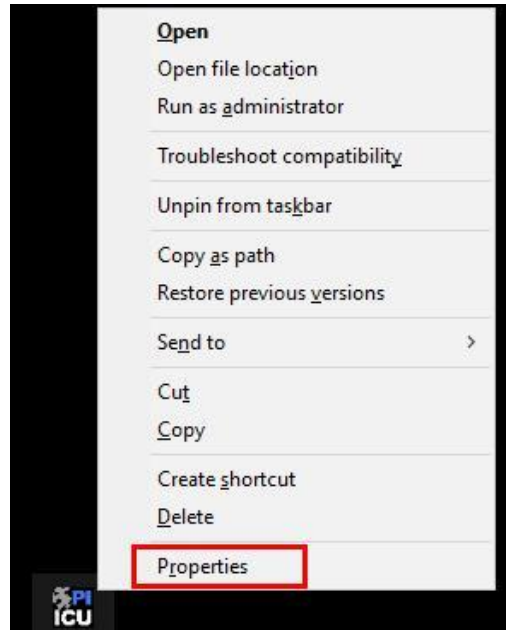
Perhaps the simplest way to view the logs is using the 'View current PI Message log continuously' button in the PI Interface Configuration Utility (PI ICU).

But first, you'll be using PI ICU a lot, so let's add it to the task bar. **Find PI ICU in the start menu and right click -> More -> Pin to taskbar**

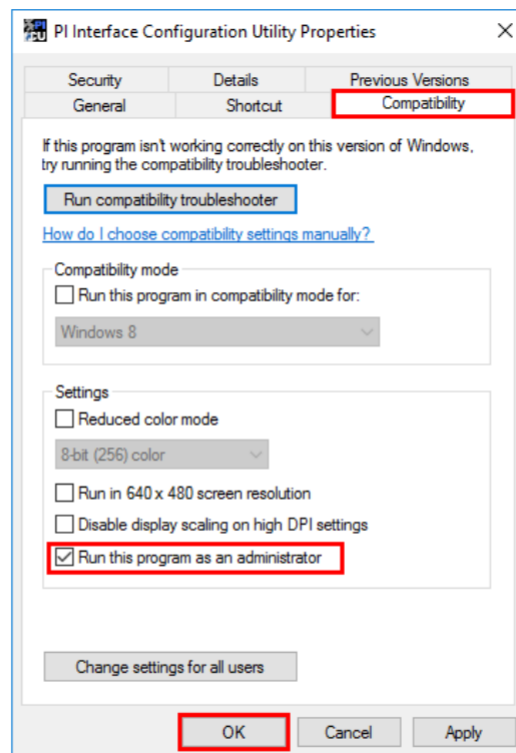


There are certain operations that will prompt you to re-open PI ICU as an administrator which can be annoying. Further, there are operations which will fail if you forget to run as administrator. Let's modify the taskbar item so that it runs as administrator every time.

Hold shift while right-clicking PI ICU in the task bar and enter the properties:

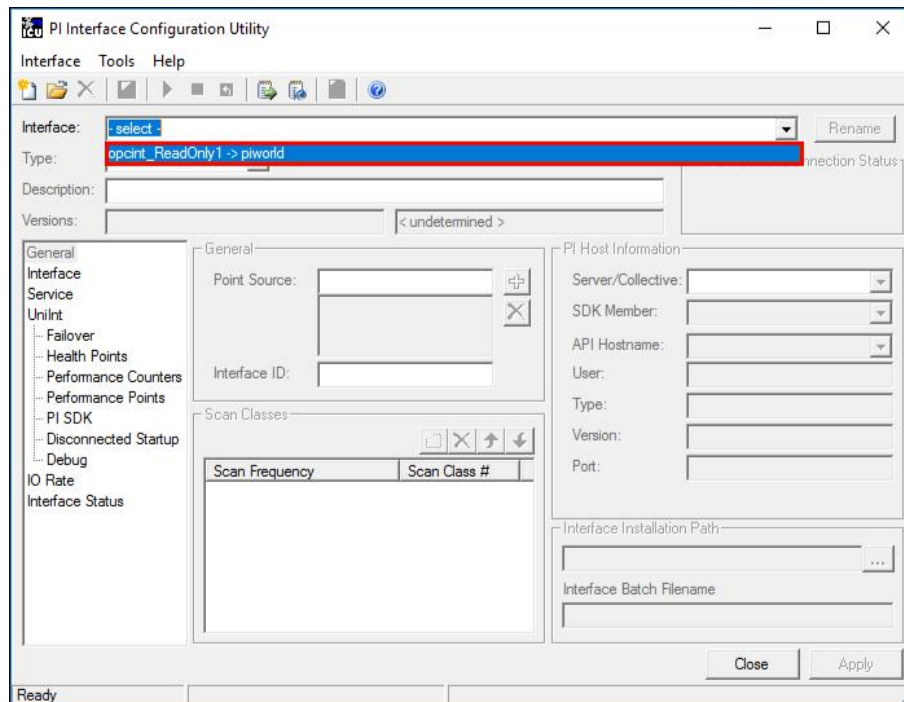


In the **compatibility** tab, check 'Run this program as an administrator' and click OK.

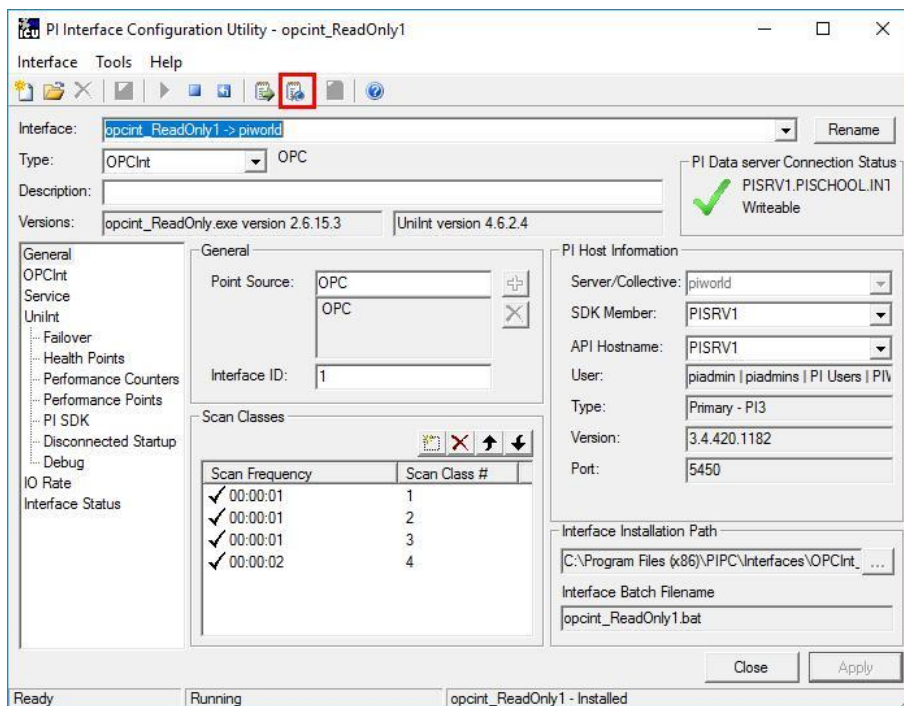


Now back to troubleshooting.

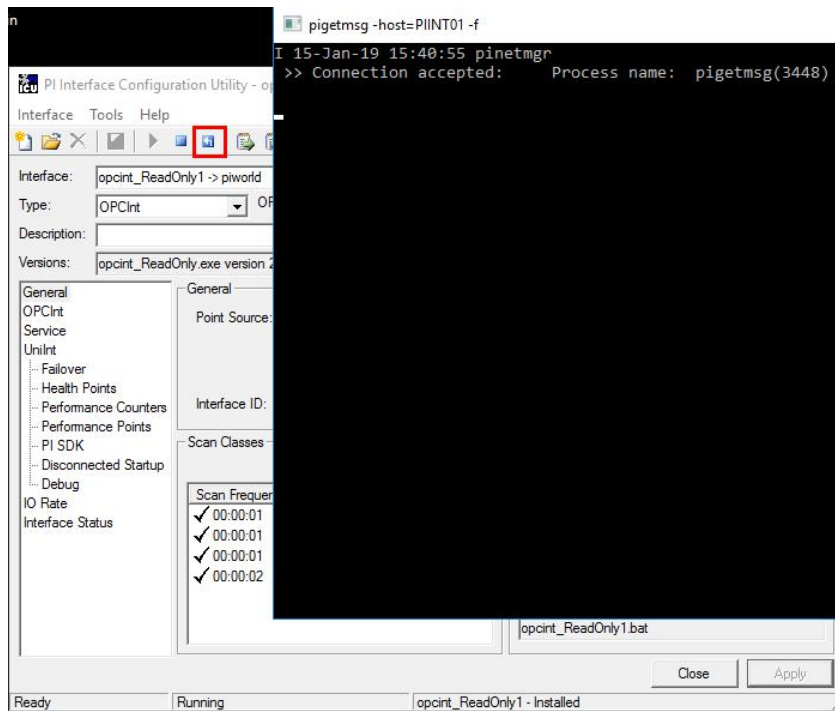
Launch PI ICU from the taskbar and select `opcint_ReadOnly1` from the drop-down



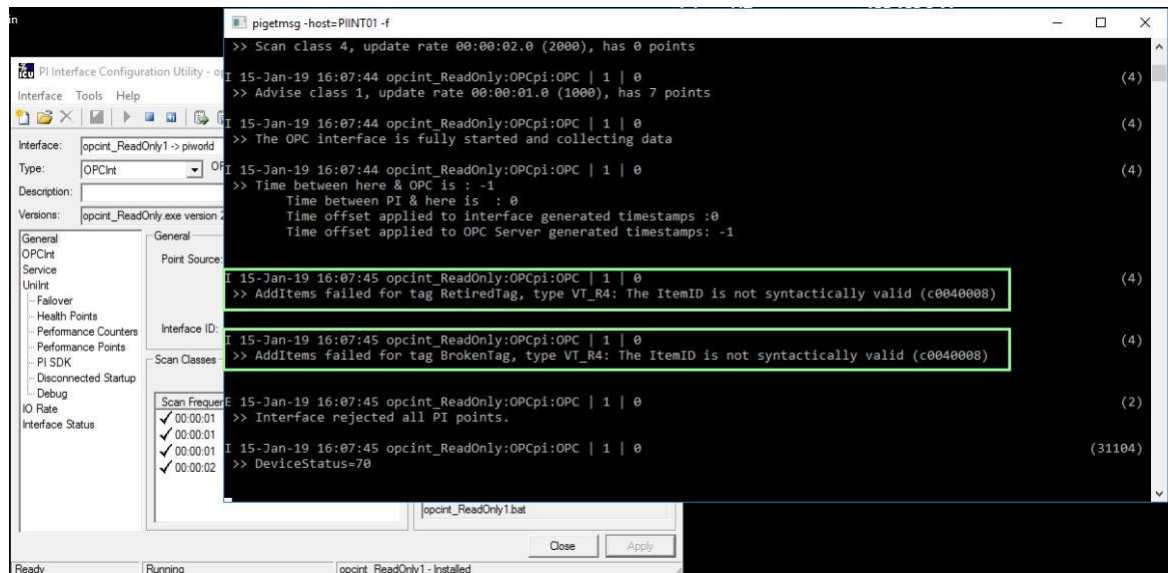
Click the 'View current PI Message log continuously' button.



Reposition the log window so that the 'restart interface service' button is visible, then click the 'restart interface service' button.



Inspect the output and you'll see what's wrong with BrokenTag and RetiredTag.



We can see the following error messages:

AddItems failed for tag RetiredTag, type VT_R4: The ItemID is not syntactically valid (c0040008)

AddItems failed for tag BrokenTag, type VT_R4: The ItemID is not syntactically valid (c0040008)

In real life, it would be a good idea to search these phrases on techsupport.osisoft.com if you didn't know what they meant.

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.

Regardless of the exact message, the problem is almost always the same: 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.

In fact, a very high percentage of the time, simply seeing a value of Configure in an OPC Tag means the ItemID is wrong, and you would mainly be going through the motions to prove it. [This KB Article](#) shows how you can do this with an OPC client.

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.Int3**.

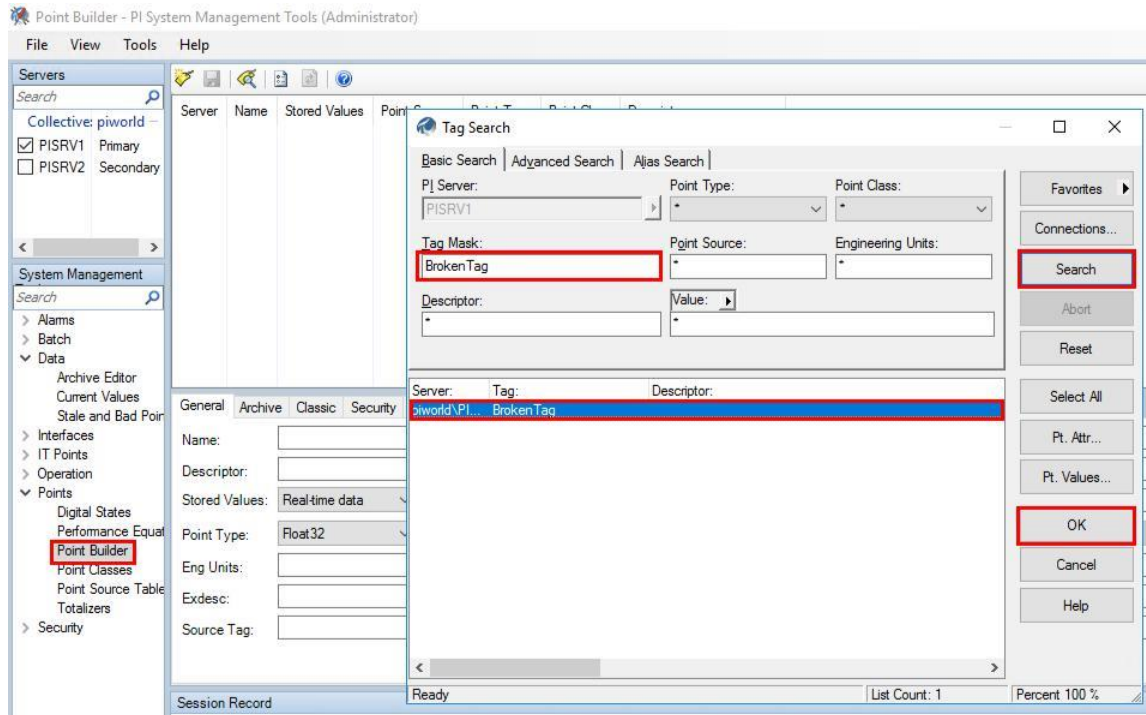
The instrumentation person also informs you that **RetiredTag no longer has any corresponding instrumentation. The equipment has been removed**. You now have a few options:

- **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. Minimize RDCMan and **go back to PI SMT on PICLIENT01**.

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



The ItemID is stored in the instrumenttag 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\OPCInt_ReadOnly for PI_OPCLnt.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.Int3** as the instrumenttag, and save.

The screenshot shows the PI Interface Classic tab. At the top, a table lists points. Below the table, the 'Classic' tab is selected, showing configuration fields for the 'BrokenTag' point. The 'Instrument Tag' field is highlighted with a red box and contains the text 'Random.Int3'.

Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor	Point Security
PISRV1	BrokenTag	Real-time data	OPC	Float32	classic		piadmins: A(r,w) PI

General Archive **Classic** Security System

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

Instrument Tag:

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

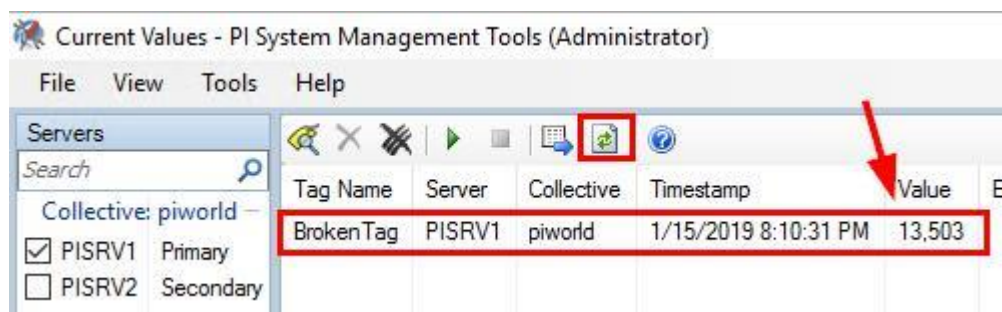
I 15-Jan-19 20:05:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> Have at least required percent good tags, clearing device status.

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 again, but you won't see the message in the previous screenshot during an interface startup.

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!



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.

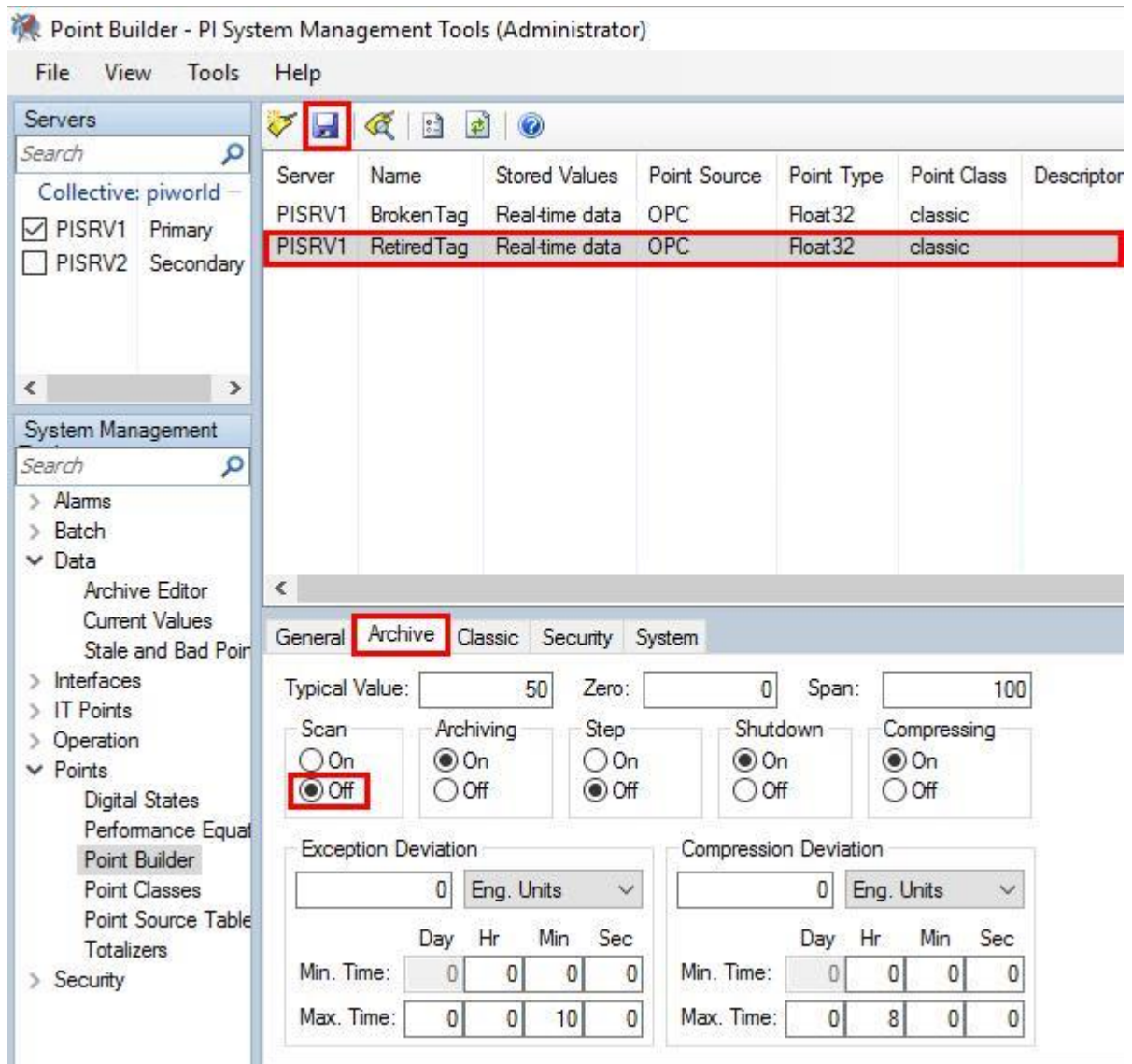


Retire tags by assigning an unused point source in addition to setting the scan attribute to off. This will prevent the PI Interface from loading and

Best Practice evaluating the tag upon startup.

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.



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

```
I 15-Jan-19 20:29:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> Deleted point RetiredTag

I 15-Jan-19 20:29:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> Scan Off: Point ID 1111: Tag 'RetiredTag'. Point will not be added

I 15-Jan-19 20:29:45 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> tag RetiredTag (1111) add point did not succeed. tag is removed from the Interface
```


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

The screenshot shows the PI Interface Configuration Utility. At the top, a table lists points:

Server	Name	Stored Values	Point Source	Point Type	Point Class	Descriptor	Point
PISRV1	BrokenTag	Real-time data	OPC	Float32	classic		piadr
PISRV1	RetiredTag	Real-time data	ZZZ_OPC	Float32	classic		piadr

Below the table, the 'General' tab is selected. The configuration fields are as follows:

- Name: RetiredTag
- Descriptor: (empty)
- Stored Values: Real-time data (dropdown)
- Point Source: ZZZ_OPC (dropdown, highlighted with a red box)
- Point Type: Float32 (dropdown)
- Digital Set: (empty)
- Eng Units: (empty)
- Exdesc: (empty)
- Source Tag: (empty)

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

The screenshot shows the PI Interface Configuration Utility on the left and a terminal window on the right. The terminal window displays the following logs:

```

pigetmsg -host=PIINT01 -f
>> Advise class 1, update rate 00:00:01.0 (1000), has 6 points
I 15-Jan-19 20:35:00 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> The OPC interface is fully started and collecting data
I 15-Jan-19 20:35:00 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> 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 15-Jan-19 20:35:00 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> Poll Group 4 named 'Poll3'.
I 15-Jan-19 20:35:01 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> Have at least required percent good tags, clearing device status.
I 15-Jan-19 20:35:01 opcint_ReadOnly:OPCpi:OPC | 1 | 0
>> DeviceStatus=0
I 15-Jan-19 20:35:02 pinetmgr
>> Deleting connection: opcint_ReadOnly.exe(3628), Connection lost(2), [109] The pipe has been
D 15-Jan-19 20:35:02 pinetmgr:Connection Statistics
>> ID: 10; Duration: 31.2 min.; kbytes sent: 4.96582; kbytes rcv: 25.25; app: ; user: ; osuse
; ip host:
  
```

The configuration utility shows the 'General' tab for the 'opcint_ReadOnly1' interface. The 'Point Source' is set to 'ZZZ_OPC'. The 'Interface ID' is 'opcint_ReadOnly1'. The 'Scan Classes' section shows a list of scan frequencies with checkboxes:

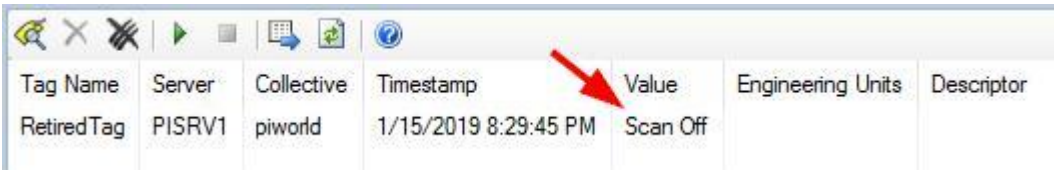
- Scan Frequency: 00:00:01 (checked)
- Scan Frequency: 00:00:01 (checked)
- Scan Frequency: 00:00:01 (checked)
- Scan Frequency: 00:00:02 (checked)

The 'Interface Batch Filename' is 'opcint_ReadOnly1.bat'. The status bar at the bottom indicates 'Ready' and 'Running'.

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.



The screenshot shows a software window with a toolbar at the top containing icons for zoom, pan, delete, play, stop, and help. Below the toolbar is a table with the following data:

Tag Name	Server	Collective	Timestamp	Value	Engineering Units	Descriptor
RetiredTag	PISRV1	piworld	1/15/2019 8:29:45 PM	Scan Off		

A red arrow points to the 'Value' column header.

Success!

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 there is no data in their PI ProcessBook display. The Trend is empty and only one of the Bars appear to be working.

The user informs you what one of the offending tags is called Random.Real9.

4.2 Tasks

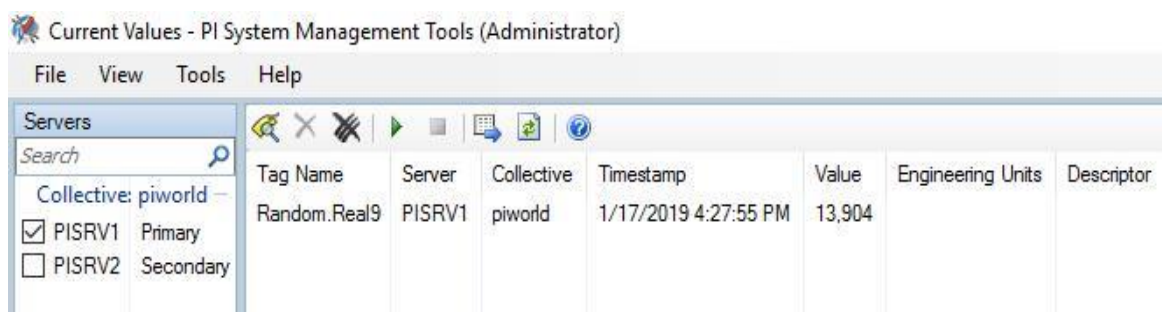
- Manually switch connections between PI Collective members
- Compare Current Values between PI Collective members
- Configure PI Buffer Subsystem
- Reinitialize a PI Collective

4.3 Step-by-Step Instructions

a. Rule out user error by checking the tag values

While on the phone with the user, let's quickly check the Current Value of Random.Real9

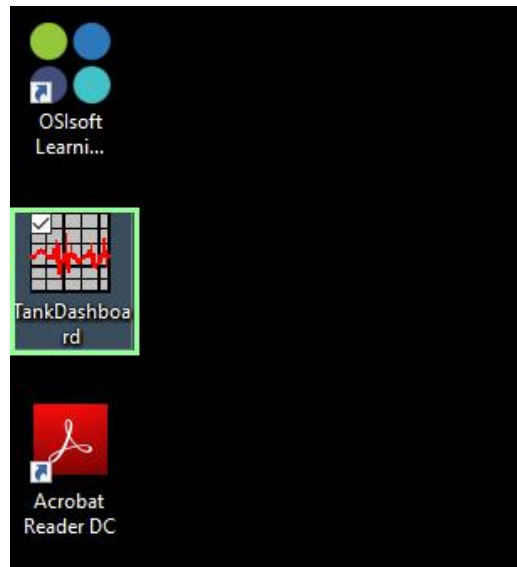
Everyone's values will be different and will not match the workbook because this is randomly generated data!



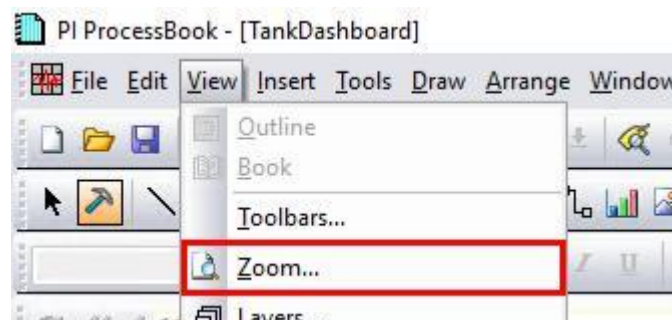
Tag Name	Server	Collective	Timestamp	Value	Engineering Units	Descriptor
Random.Real9	PISRV1	piworld	1/17/2019 4:27:55 PM	13,904		

Random.Real9 seems to have a value... what is the user complaining about?

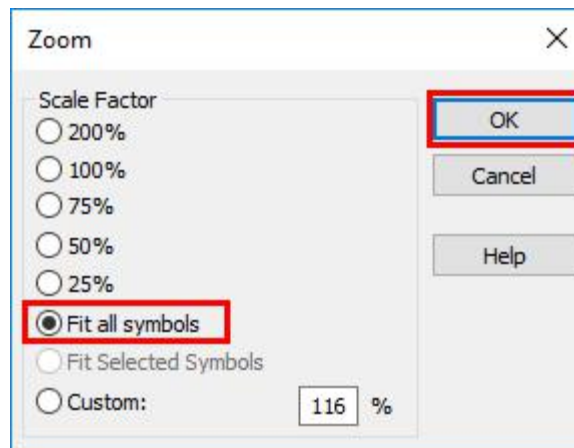
At this point we'd probably do a screen share with the user to see exactly what their problem is. In the lab we'll have to pretend a bit. Open the TankDashboard PI ProcessBook display on the desktop and pretend that you're looking at the user's desktop.



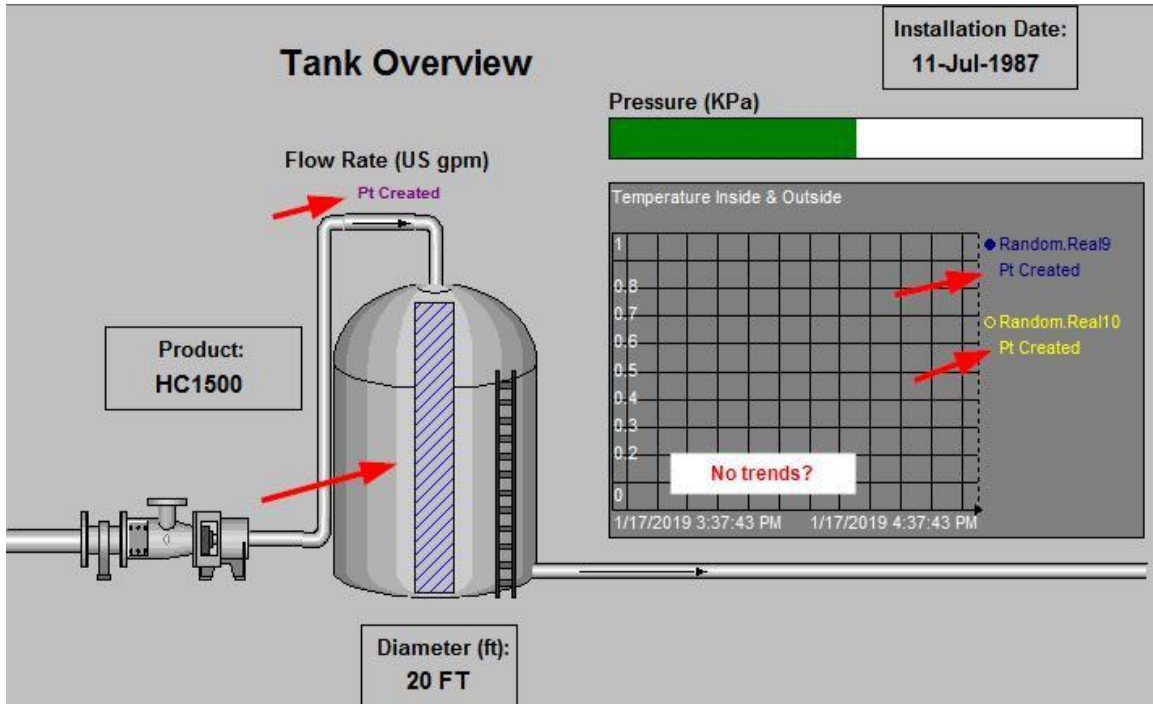
Let's quickly adjust the zoom so all symbols fit on screen. Do View -> Zoom



Fit all symbols, OK



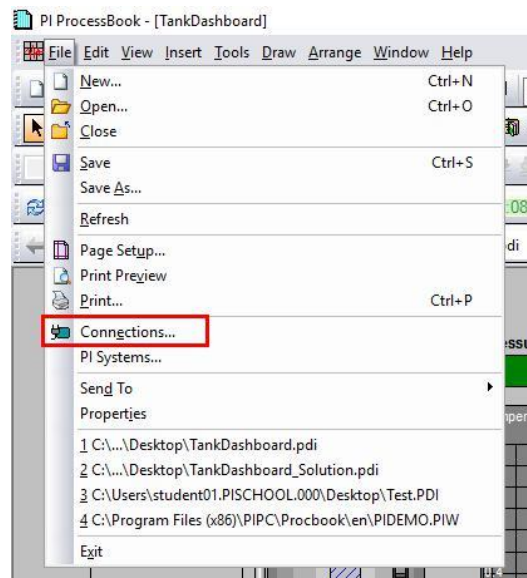
Ok, clearly there is something wrong.



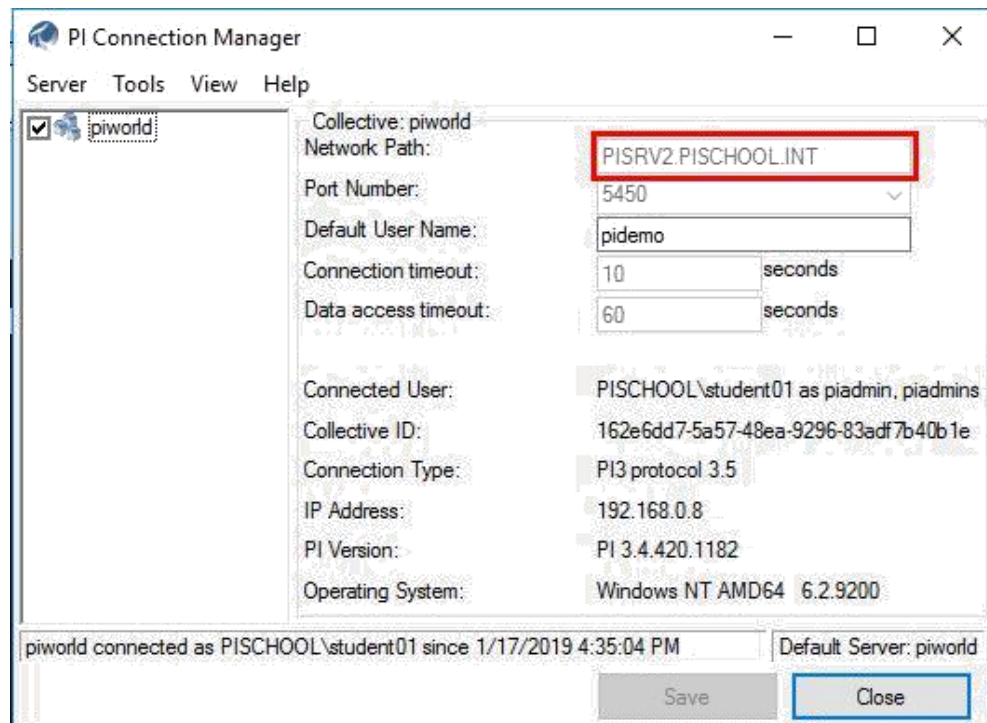
b. Begin Troubleshooting

You don't really know what to do, so you start flailing around with the buttons and menus. You hit Revert, doesn't help. You ask the user to reboot their machine. Still same problem.

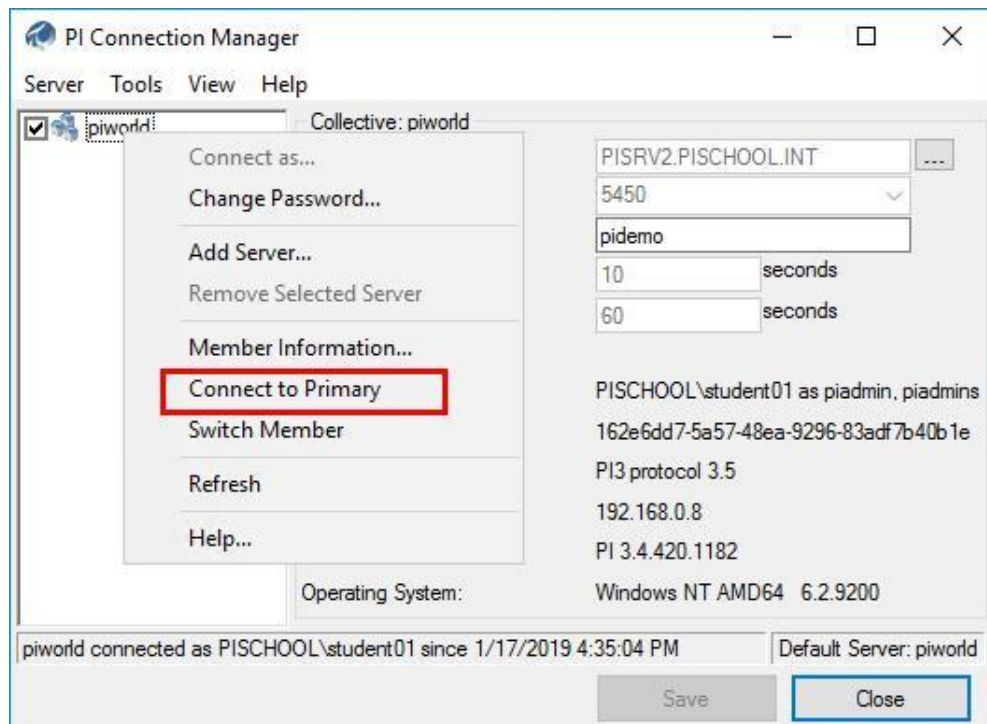
Eventually you stumble upon the Connections menu:



You see something a little odd. It looks like we're connected to PISRV2, the Secondary.

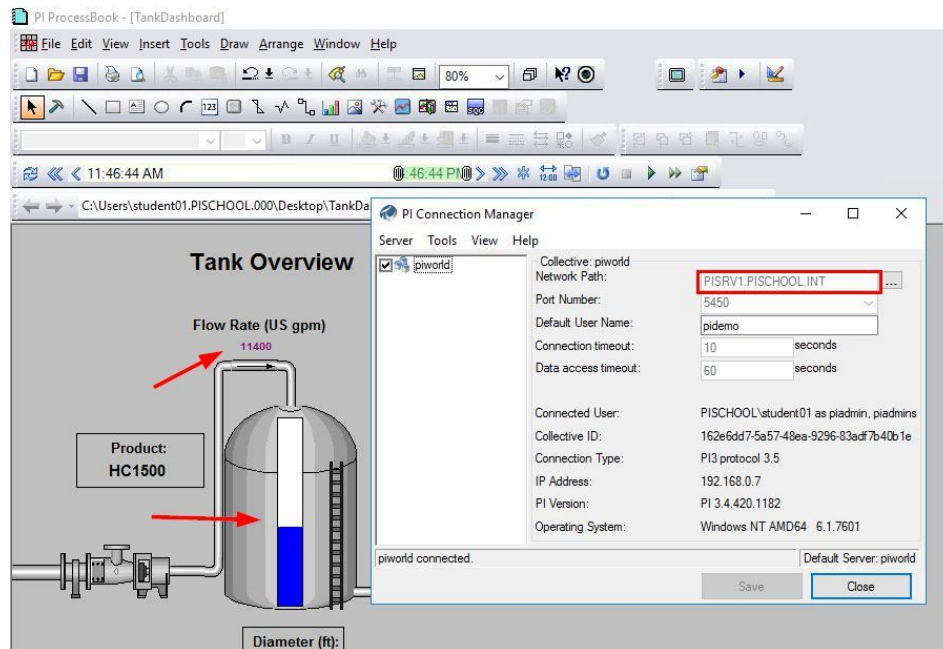


Let's try connecting to the Primary and see what happens. Right click the piworld PI Collective and connect to the Primary.



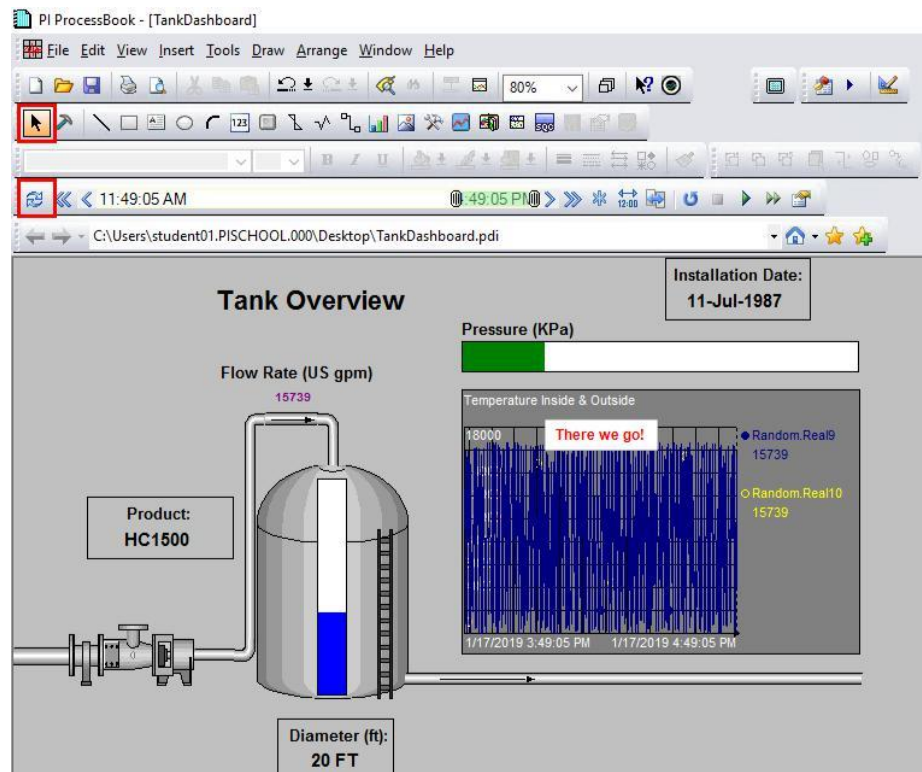
Looks like some stuff came back.

Beginning Administration and Management of the PI System



Close PI Connection Manager. The Trends are still not quite right. They should be full from left to right.

Click the mouse cursor to enter Run mode, and then **click Revert** to force a reload of the historical data.

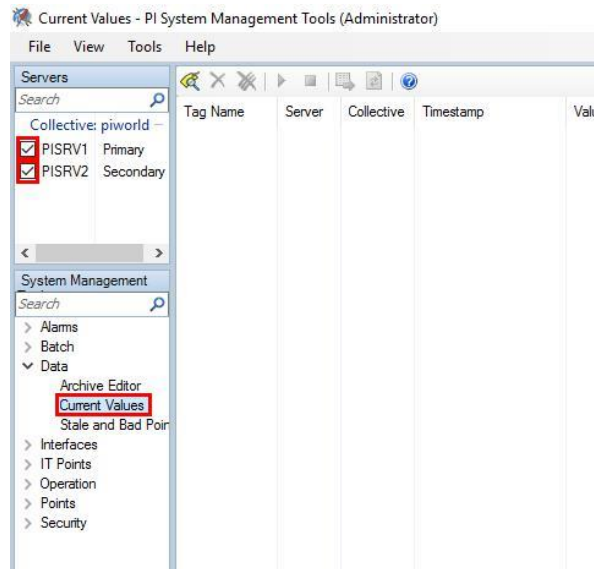


Alright, fixed... well not really. The Secondary PI Data Archive PISRV2 is not getting data, which would be a problem if the Primary ever went down. What good is a backup

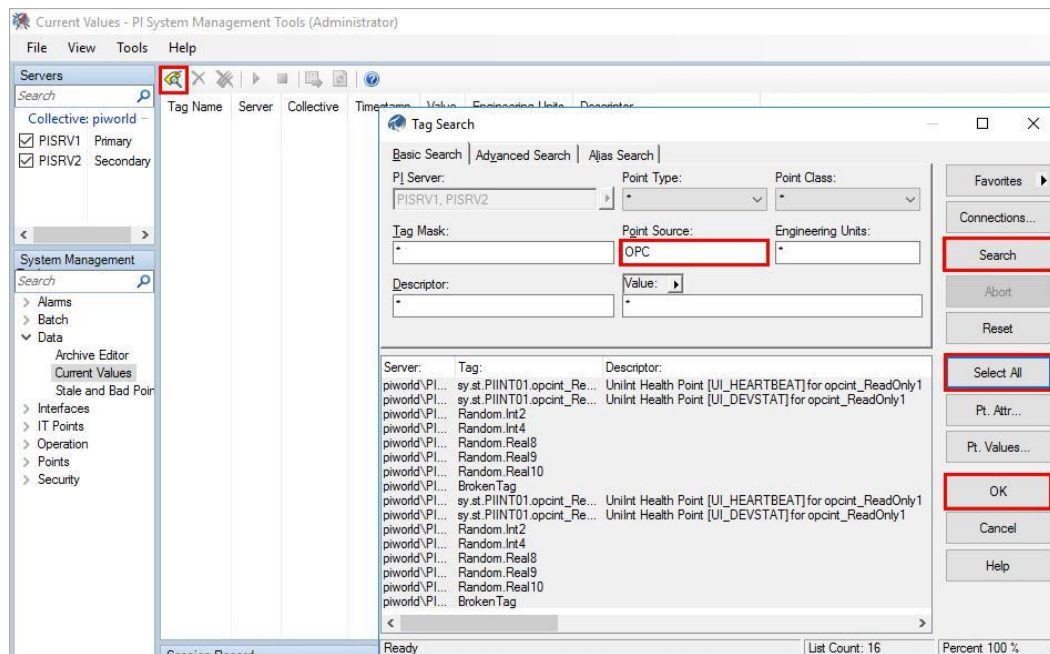
system when you can't use it?

Another way to confirm that the secondary is not getting data is to connect to PISRV1 and PISRV2 simultaneously and check the tag values side by side using the Current Values plugin.

Open PI SMT to the Current Values plugin and connect to PISRV1 and PISRV2.



You found that all the tags in the PI ProcessBook display had Point Source OPC, so search for all tags with Point Source OPC and add them to the display.



Sort by tag name to make the comparison easier, then take a closer look at the values.

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

Current Values - PI System Management Tools (Administrator)

Tag Name	Server	Collective	Timestamp	Value	Engineer
BrokenTag	PISRV1	piworld	1/17/2019 5:12:45 PM	10,655	
BrokenTag	PISRV2	piworld	1/11/2019 10:13:21 PM	Pt Created	
Random.Int2	PISRV1	piworld	1/17/2019 5:12:45 PM	23,142	
Random.Int2	PISRV2	piworld	1/11/2019 5:07:04 PM	14,015	
Random.Int4	PISRV1	piworld	1/17/2019 5:12:45 PM	24,887	
Random.Int4	PISRV2	piworld	1/11/2019 8:05:36 PM	Pt Created	
Random.Real10	PISRV1	piworld	1/17/2019 5:12:45 PM	1,934.1	
Random.Real10	PISRV2	piworld	1/11/2019 8:54:46 PM	Pt Created	
Random.Real8	PISRV1	piworld	1/17/2019 5:12:45 PM	1,934.1	
Random.Real8	PISRV2	piworld	1/11/2019 8:05:36 PM	Pt Created	
Random.Real9	PISRV1	piworld	1/17/2019 5:12:45 PM	1,934.1	
Random.Real9	PISRV2	piworld	1/11/2019 8:54:46 PM	Pt Created	
sy.st.PIINT01.opcint_ReadOnly1.Device Status	PISRV1	piworld	1/17/2019 2:39:16 PM	0 Good	
sy.st.PIINT01.opcint_ReadOnly1.Device Status	PISRV2	piworld	1/11/2019 4:53:57 PM	0 Good	
sy.st.PIINT01.opcint_ReadOnly1.Heartbeat	PISRV1	piworld	1/17/2019 5:12:45 PM	8	seconds
sy.st.PIINT01.opcint_ReadOnly1.Heartbeat	PISRV2	piworld	1/11/2019 5:07:04 PM	7	seconds

Looks like in some cases it used to be working? Why are the data and timestamps different between the 2 servers?

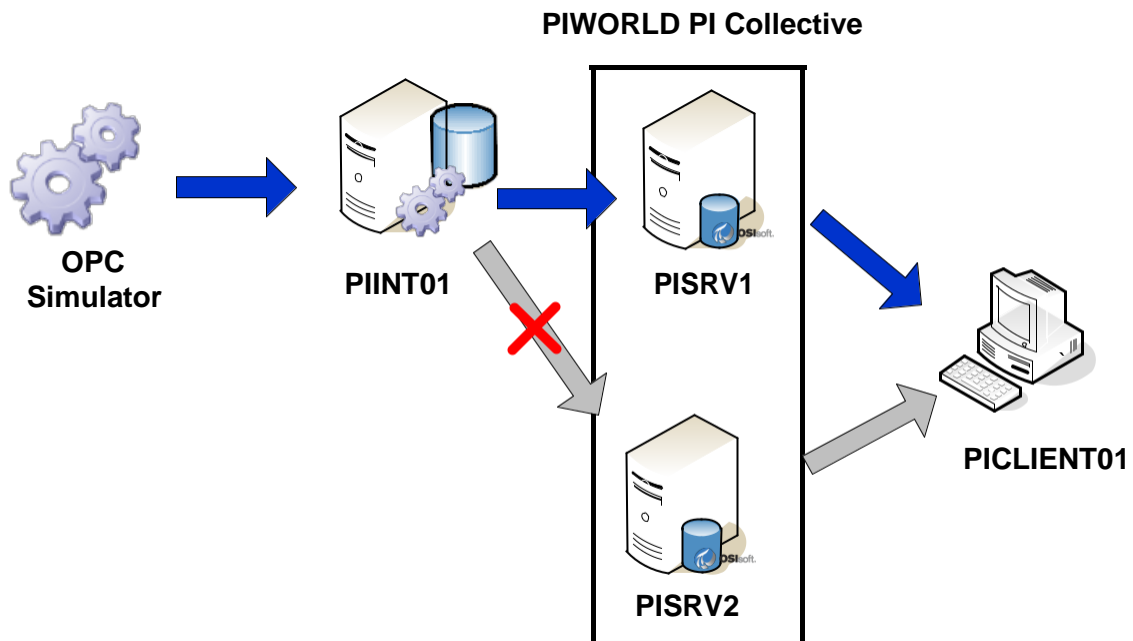
Tip

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.

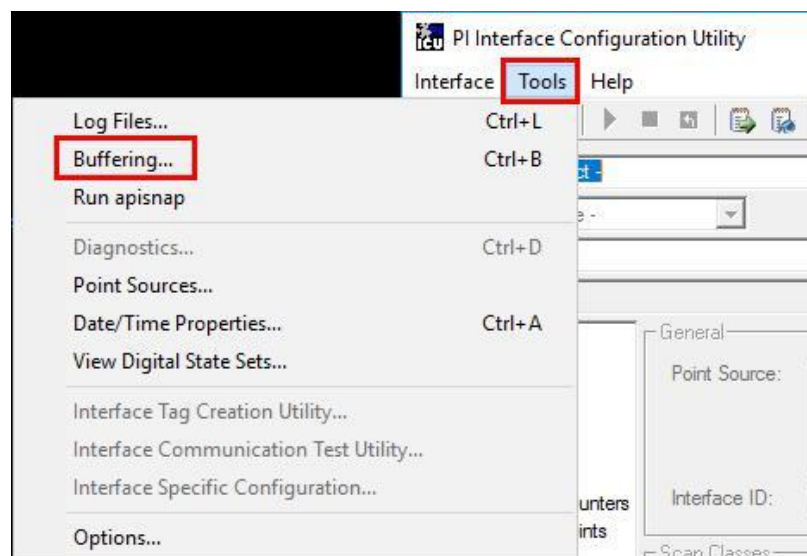


PI Buffer Subsystem must be configured on every node with an application that writes to the PI Data Archives. This of course includes PI Interfaces.

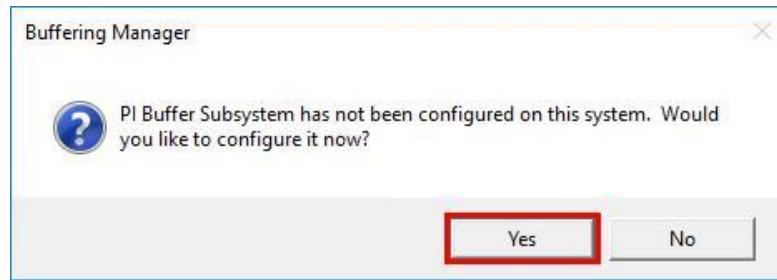
c. Check the PI Interface Node and configure PI Buffer Subsystem

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

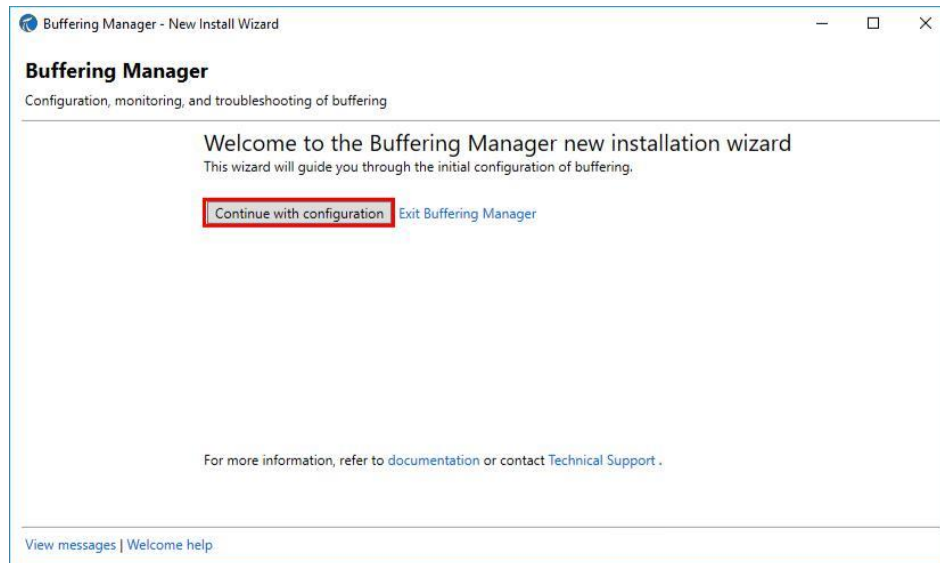
On PIINT01, open PI ICU and do Tools -> Buffering:



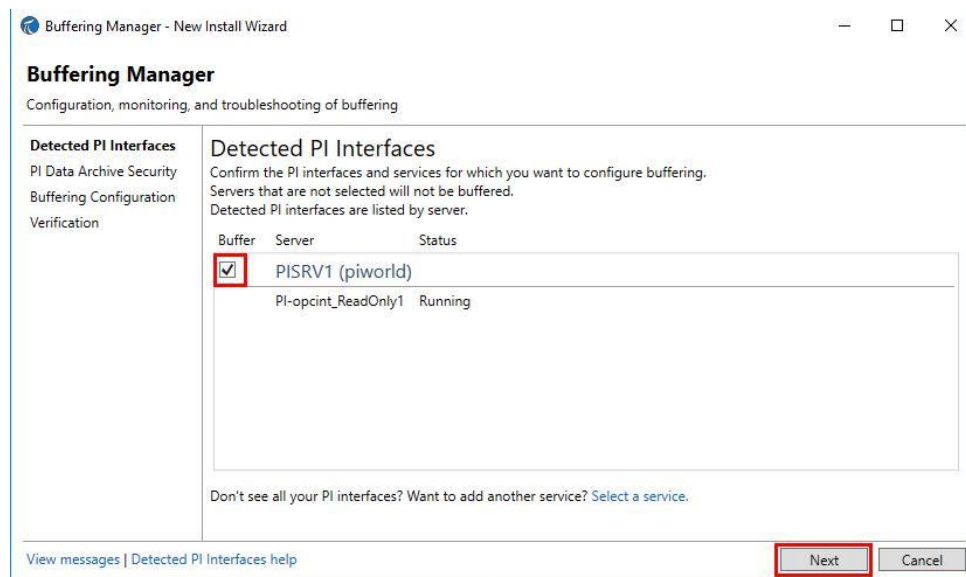
Well that would explain it. PI Buffer Subsystem is not configured. Let's configure it:



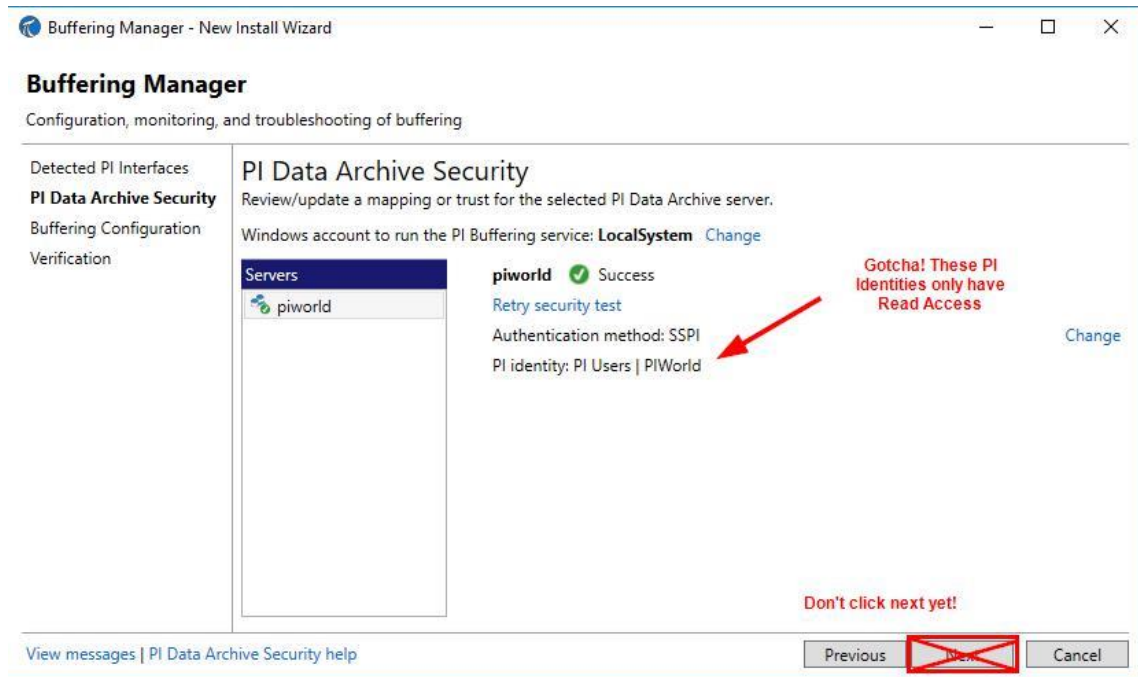
Click Continue with configuration:



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



Stop at the below screen, there is some configuration required.



Even though there is a green checkmark, we need to make a change. As-is, PI Buffer Subsystem is connecting to the PI Collective with Read Access, but it needs to Write to Tags.

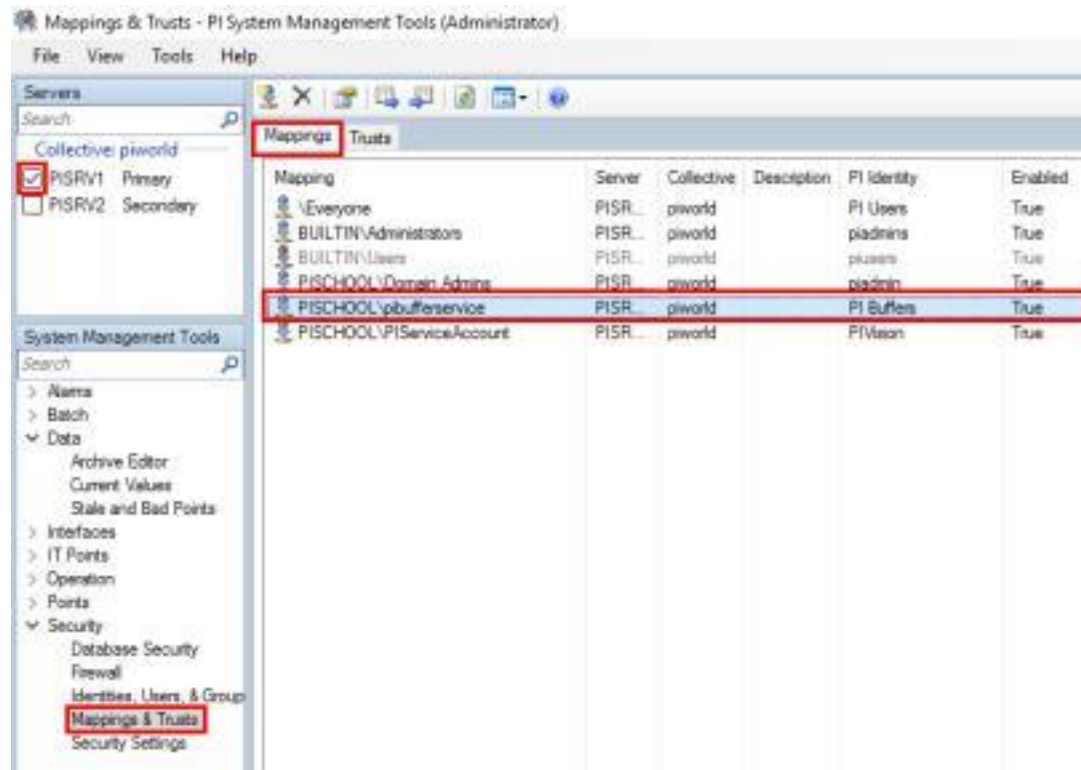
Tip

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.

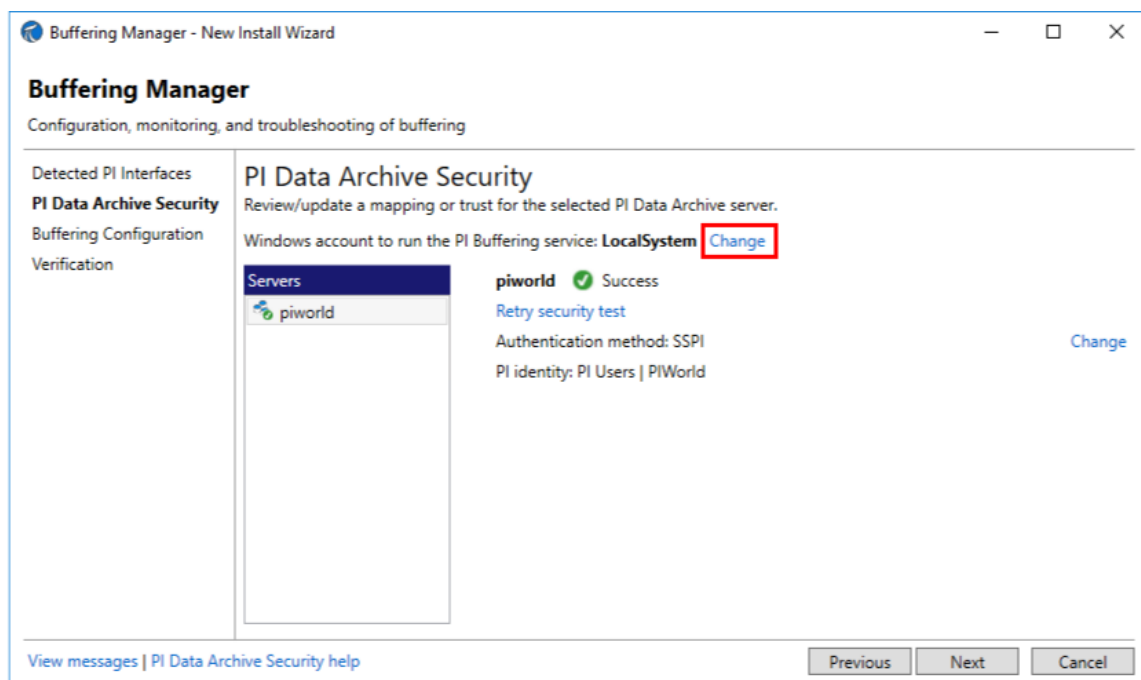
There is already a PI Identity configured for use with PI Buffer Subsystem. There's also a mapping that will grant the service account **pischool\pibufferservice** the access associated with the PI Identity.

This can be inspected in the Mappings & Trusts plugin in PI SMT.

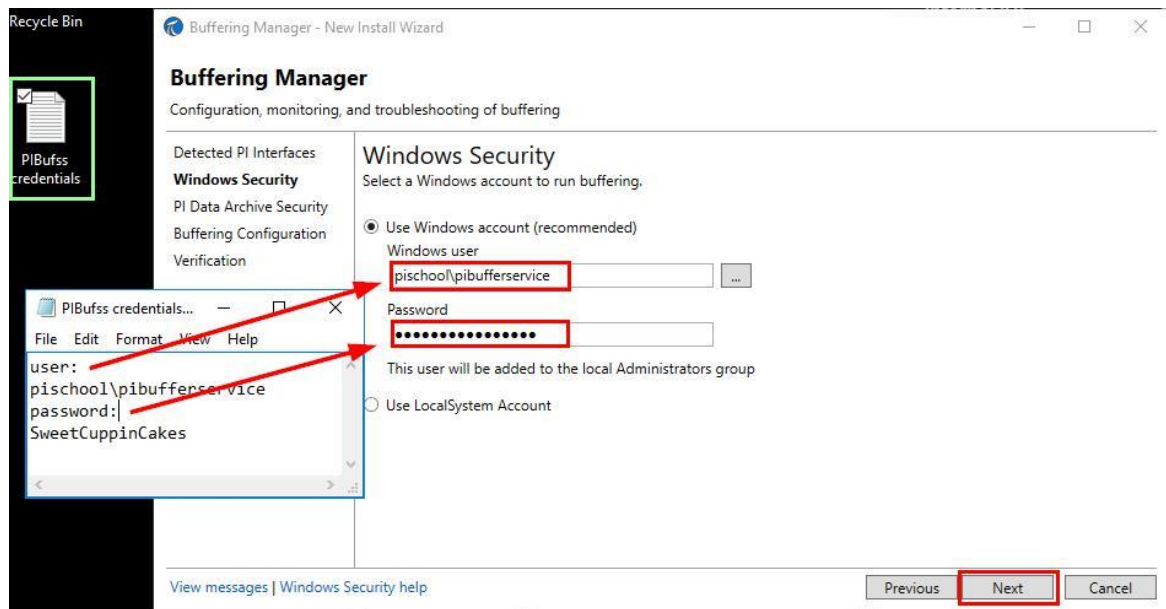
Nothing to configure here, we're just looking.



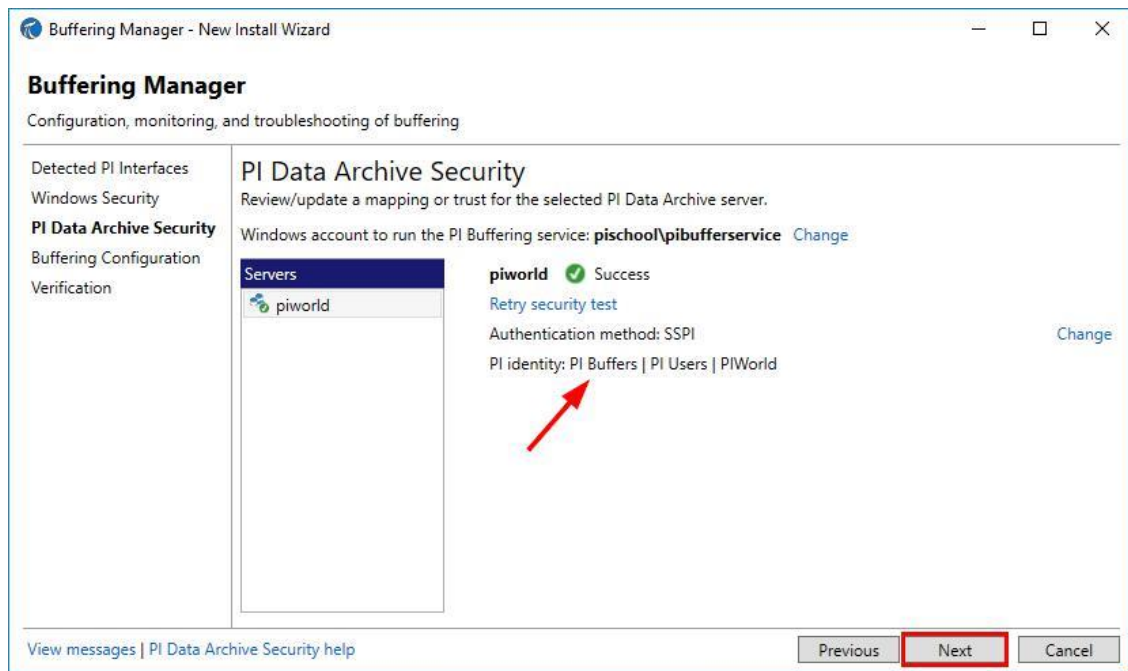
Back to PIINT01, we just need to change the service account used by PI Buffer Subsystem. Click Change:



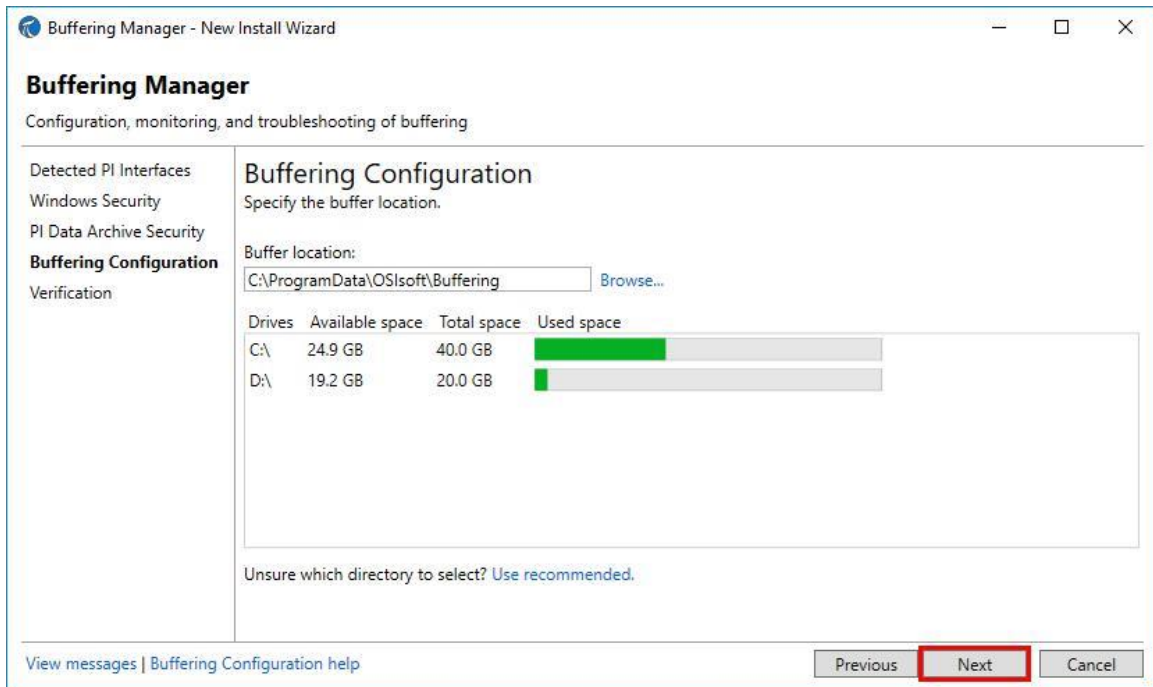
Check 'Use Windows account (recommended)', then open the text file called 'PIBufss credentials' on the desktop and copy/paste the user and password into the fields. Click Next.



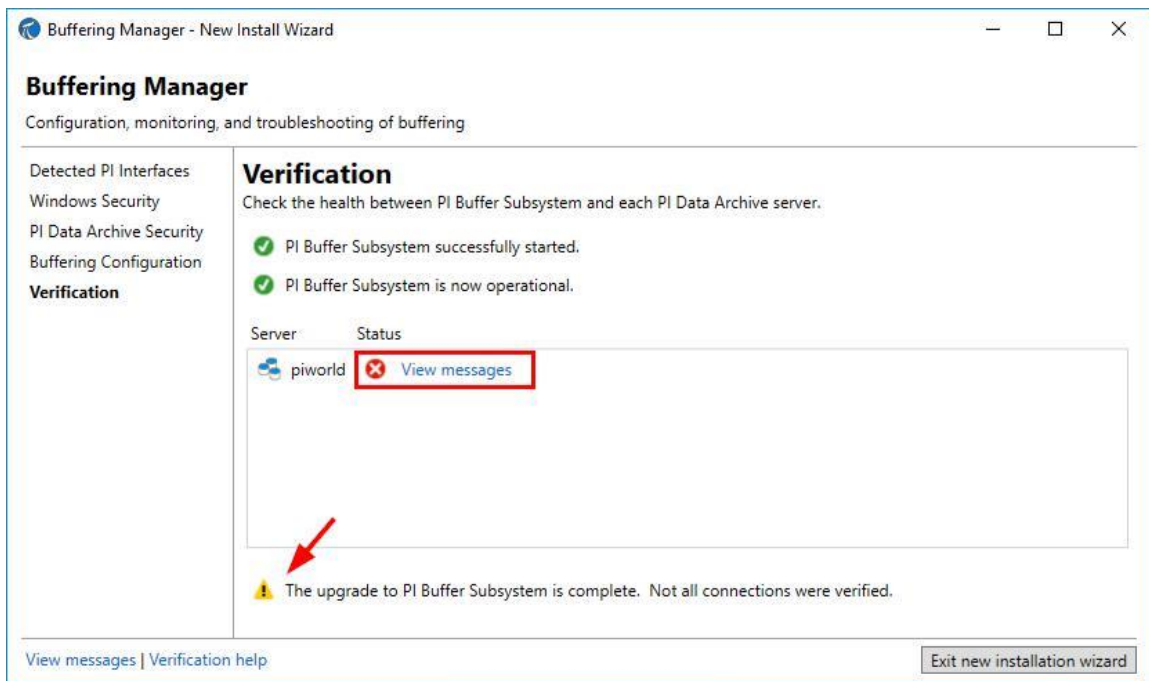
It should now show the **PI Buffers** identity, click Next



Leave the defaults, click Next

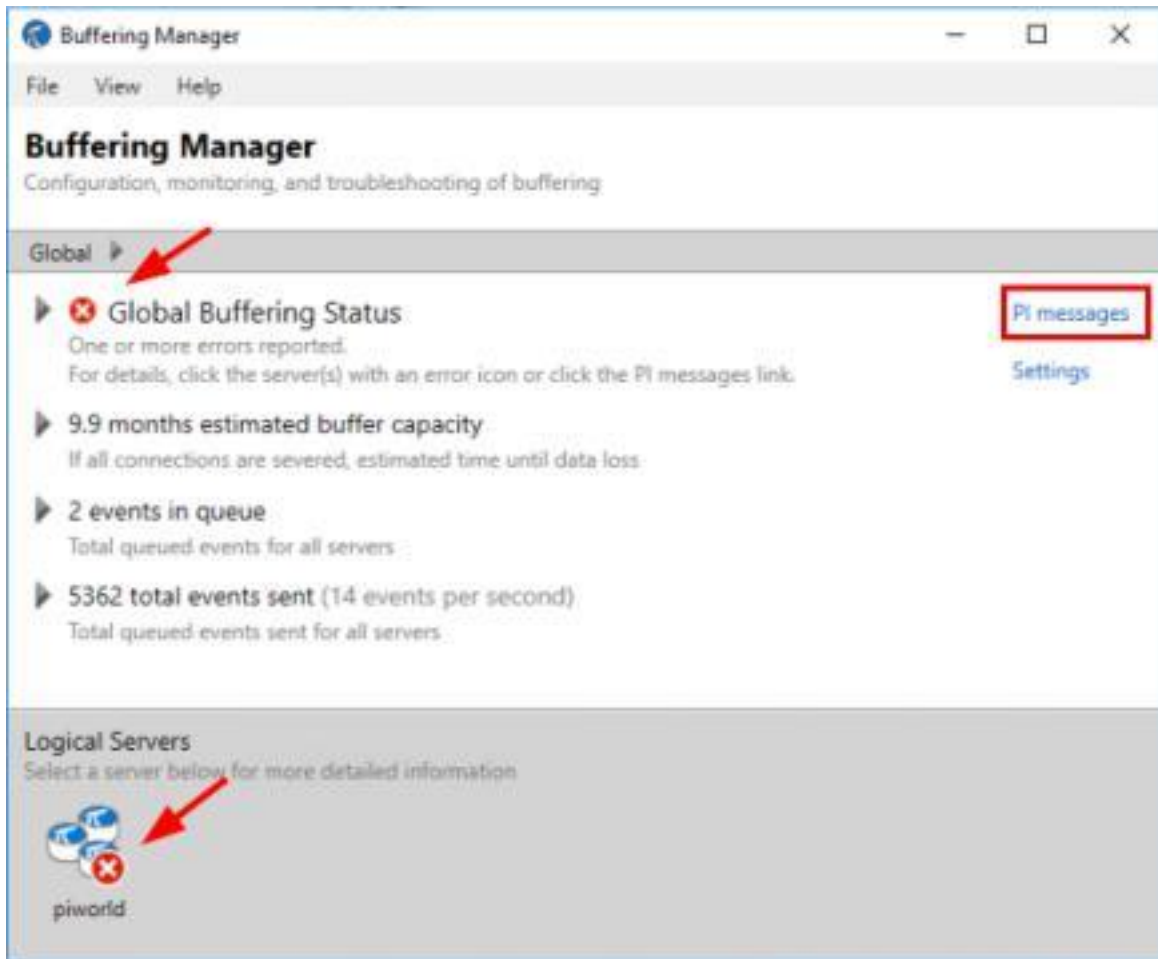


After a minute or so configuration should be complete, but it looks like there's still a problem. Click 'View messages'. **The screen might not show a problem for you depending on the timing. If so then click Exit new installation wizard**

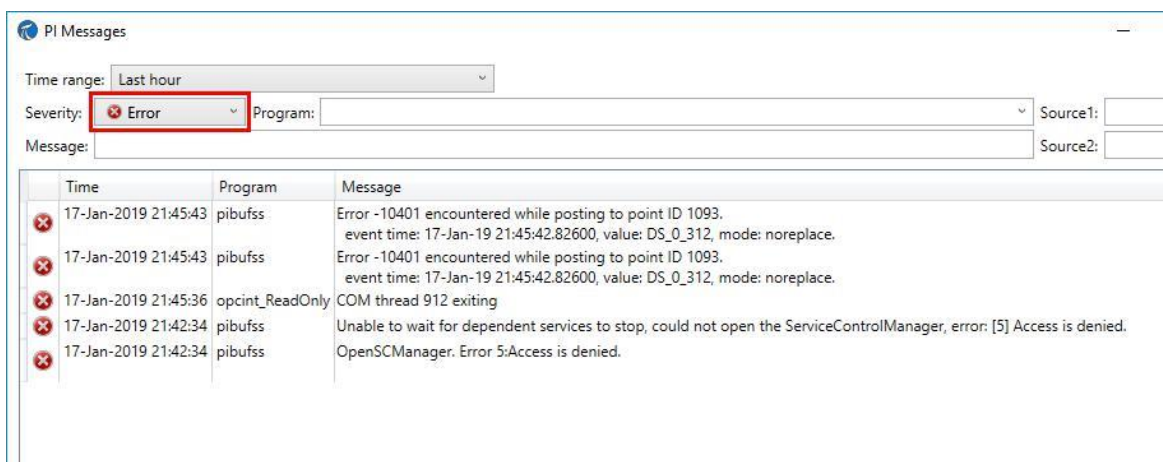


Again, the previous screen might not show a problem depending on the timing. If so then click Exit new installation wizard

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



Filter for Errors:



d. Troubleshoot Errors

Error -10401 is not giving much detail, but we can at least look up point ID 1093 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**.

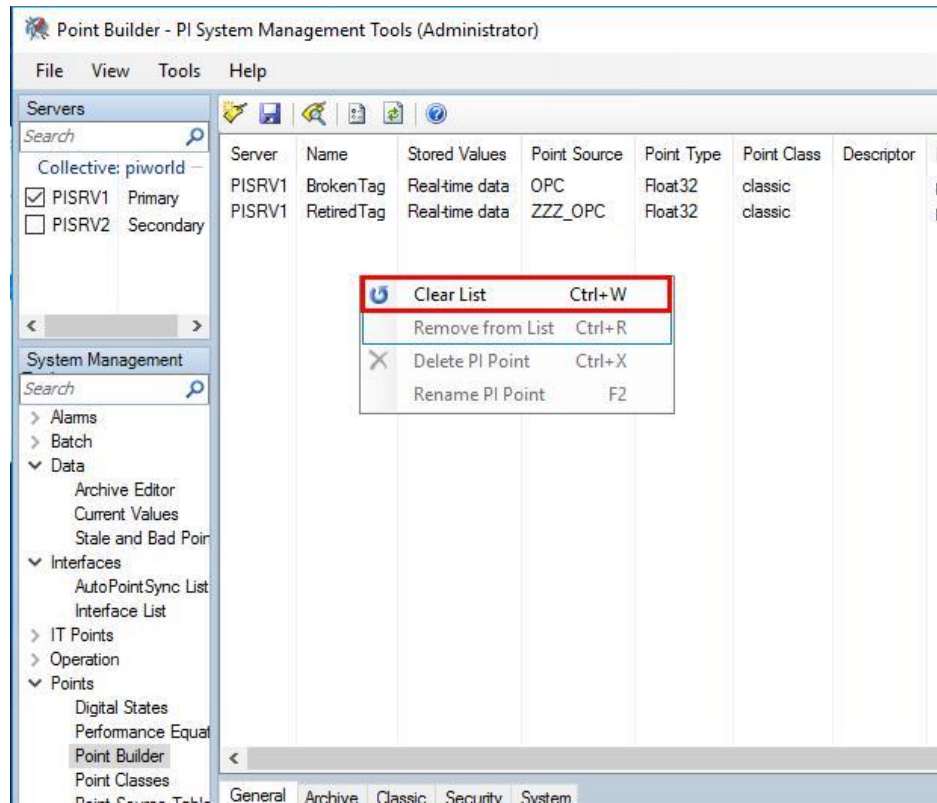
It would probably be more effective to search <https://techsupport.osisoft.com> or [PI Square](#) instead of google, but that requires a sign up. The required SSO account is free and highly recommended but we don't want to spend any time on this during the lab.

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.

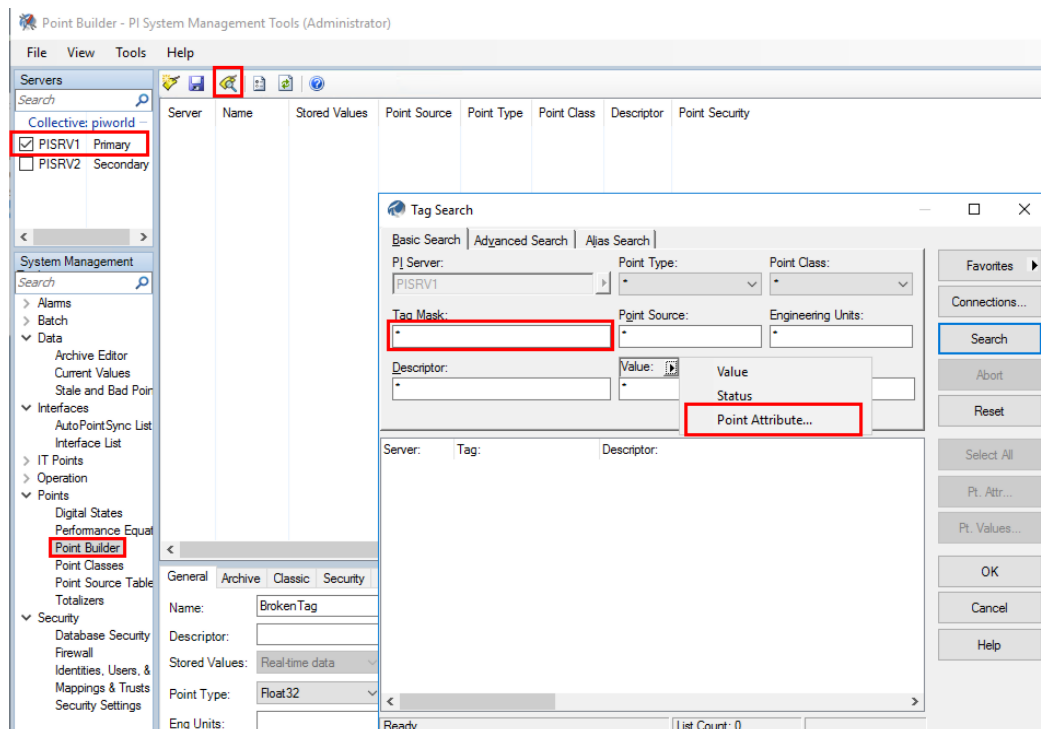
We've deduced that something doesn't have write access to a tag with pointid 1093. It stands to reason that it's probably PI Buffer Subsystem, since the tag was working before we configured buffering.

Which tag is pointid 1093? 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**.

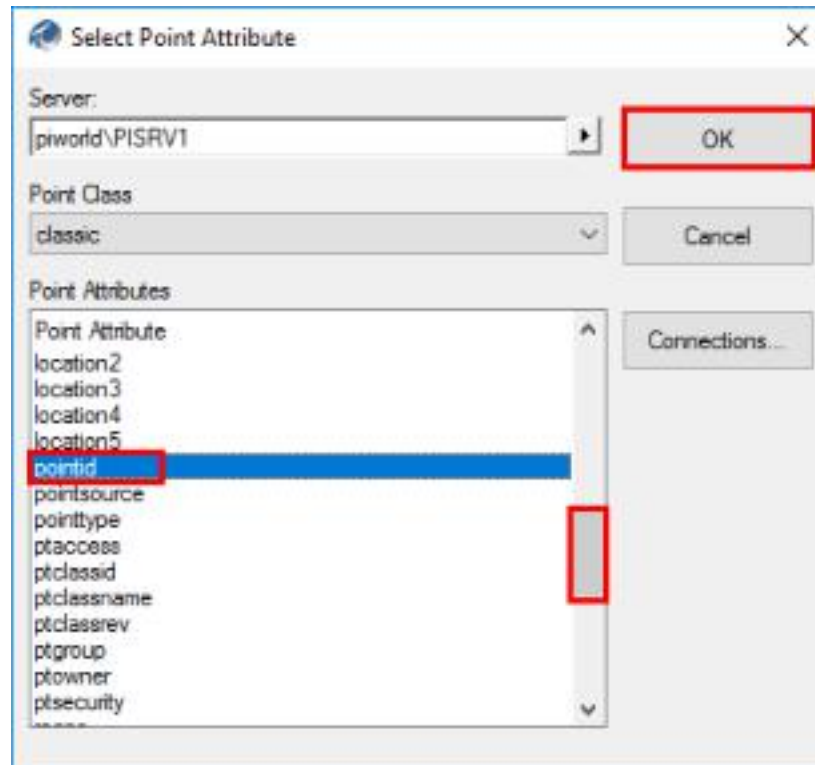
From PICLIENT01, Open PI SMT head to Point Builder. Clear the list if it's not empty.



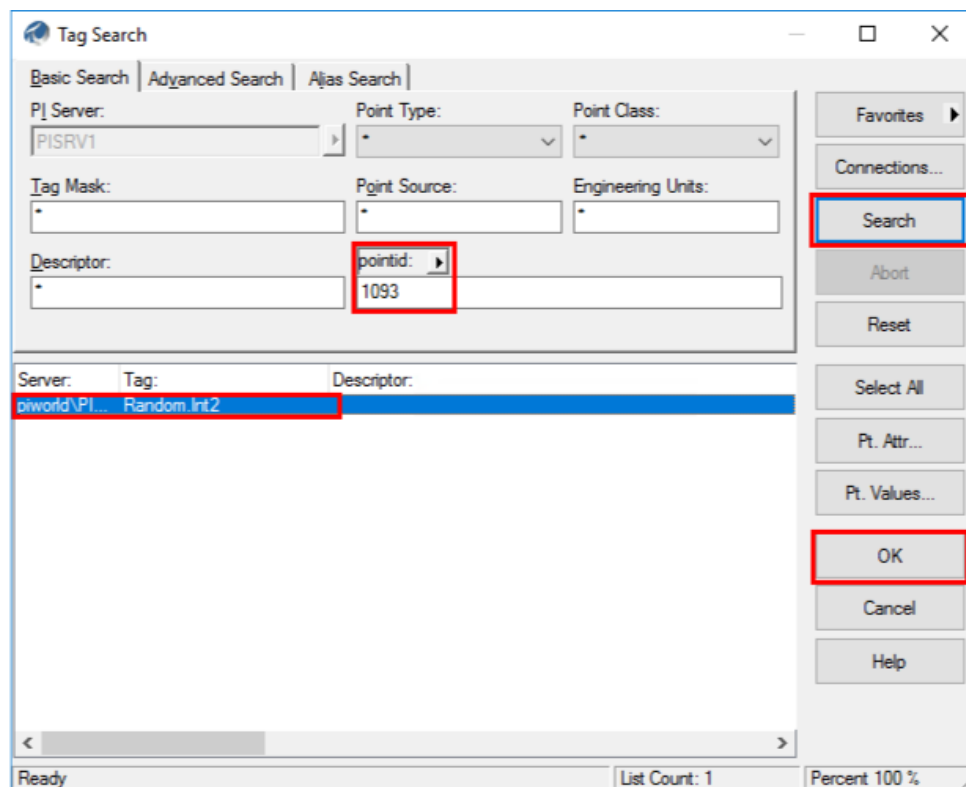
Do a tag search, ensure * is set for the Tag Mask, and change the customizable field to Point Attribute...



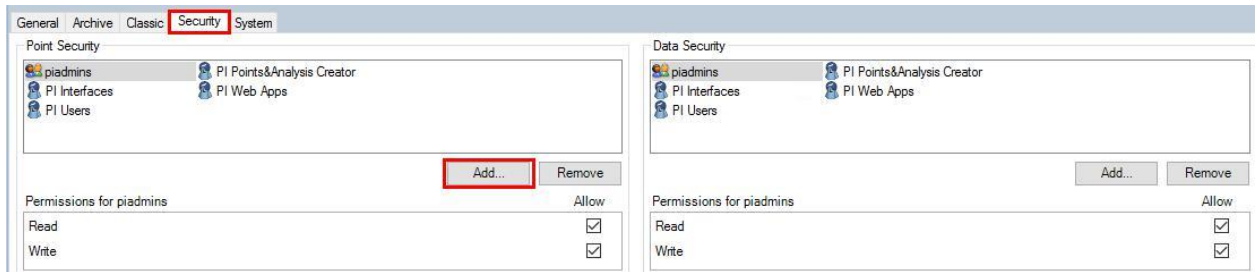
Scroll down to pointid, select pointid, click OK.



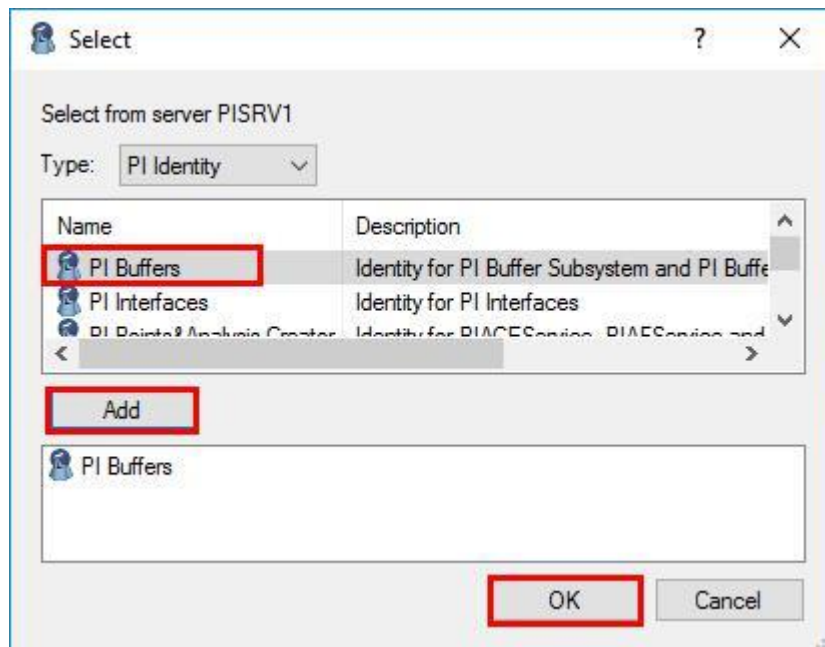
Search for pointid 1093, select tag Random.Int2, click OK



Ensure Random.Int2 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.



Click PI Buffers, Add..., OK



Also add PI Buffers to data security.

The screenshot shows the 'Security' tab of the PI System configuration window. It is divided into two main sections: 'Point Security' and 'Data Security'. In the 'Point Security' section, a list of identities includes 'piadmins', 'PI Interfaces', 'PI Users', 'PI Points&Analysis Creator', 'PI Web Apps', and 'PI Buffers'. The 'PI Buffers' identity is highlighted. Below this list, the 'Permissions for PI Buffers' are shown with 'Read' checked and 'Write' unchecked. In the 'Data Security' section, the same list of identities is shown, but 'PI Buffers' is not yet added. The 'Add...' button is highlighted with a red box. Below this, the 'Permissions for piadmins' are shown with both 'Read' and 'Write' checked.

The screenshot shows a 'Select' dialog box titled 'Select from server PISRV1'. The 'Type' is set to 'PI Identity'. A list of identities is displayed with columns for 'Name' and 'Description'. The 'PI Buffers' identity is highlighted with a red box. Below the list, the 'Add' button is highlighted with a red box. At the bottom of the dialog, the 'OK' button is highlighted with a red box.

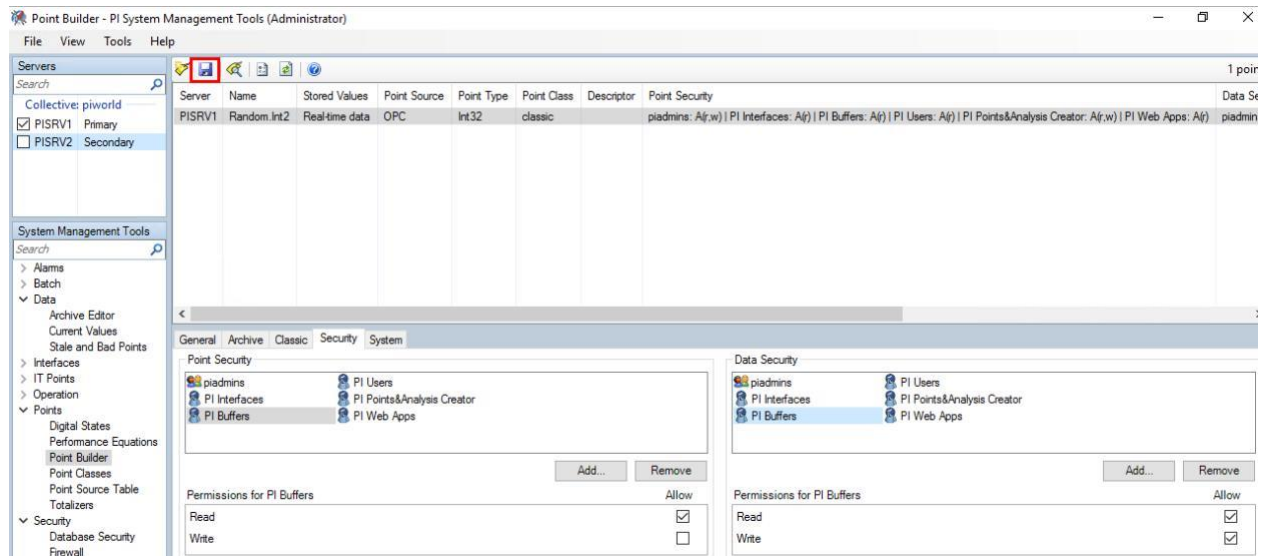
Write is required for Data Security.

This screenshot is similar to the first one, but it shows the 'Data Security' tab after the 'PI Buffers' identity has been added. In the 'Permissions for PI Buffers' section, both 'Read' and 'Write' permissions are now checked. A red arrow points to the 'Write' checkbox, which is highlighted with a red box.

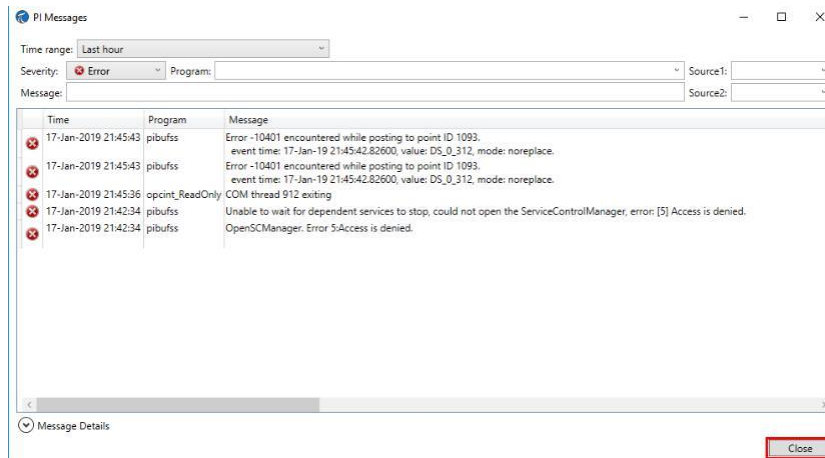
In simple terms, **Point Security** relates to the ability to search for, view, and edit the **configuration of tags**. For example, Write in Point Security was required to change the instrumenttag of BrokenTag and is required here to edit tag security settings.

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

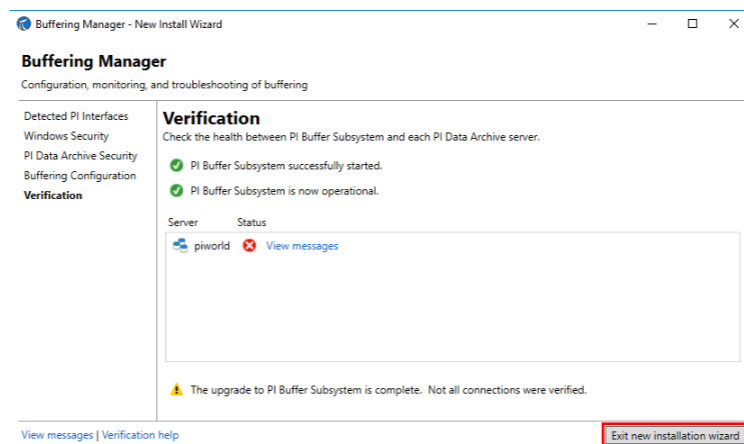
Be sure to Save!



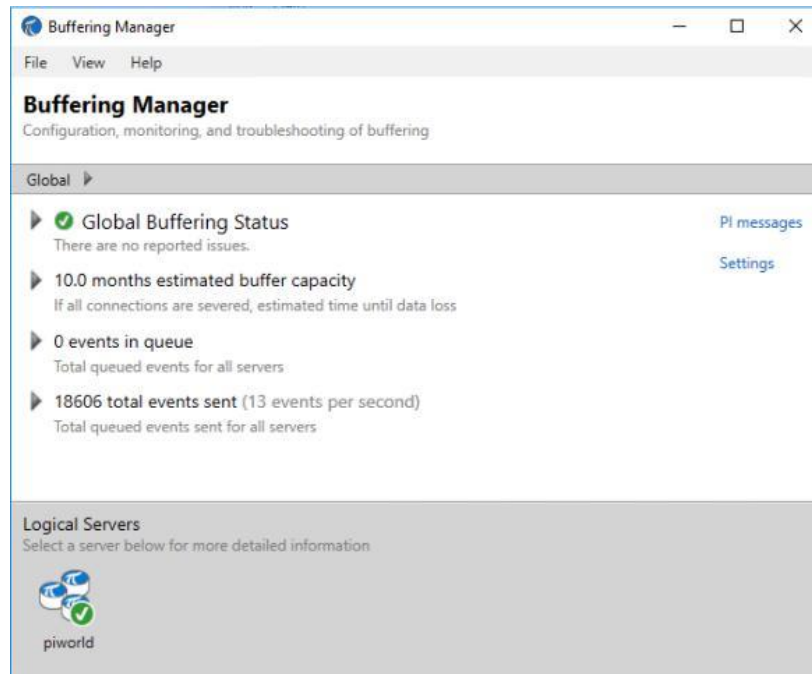
Let's go back to PIINT01. Close the message window.



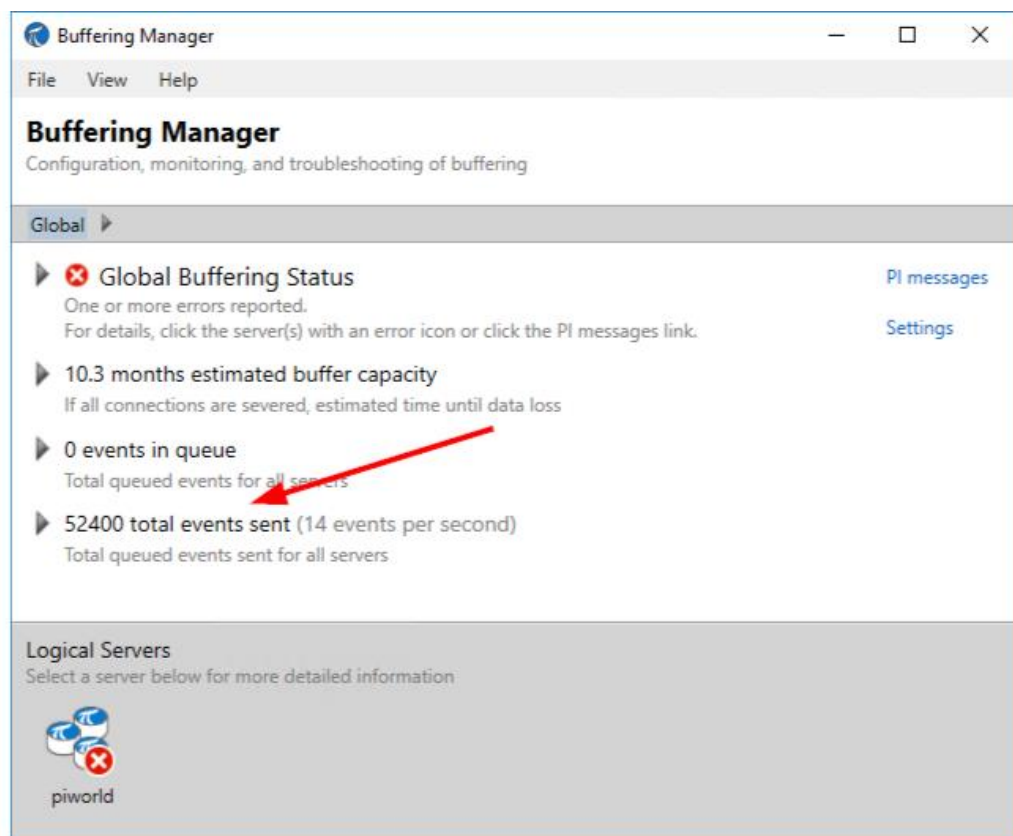
Exit new installation wizard if you haven't already.



The PI Buffer Subsystem status should be green.



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

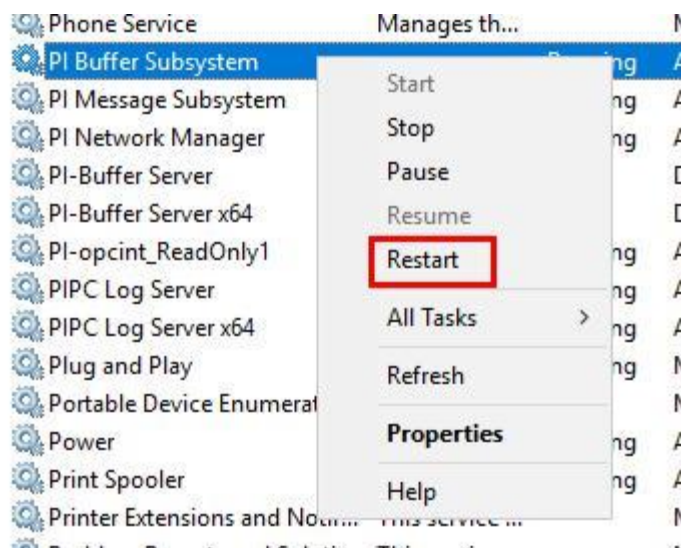


If it's still showing errors. Restart the PI Buffer Subsystem service and the PI Interface Service and see if that fixes it.

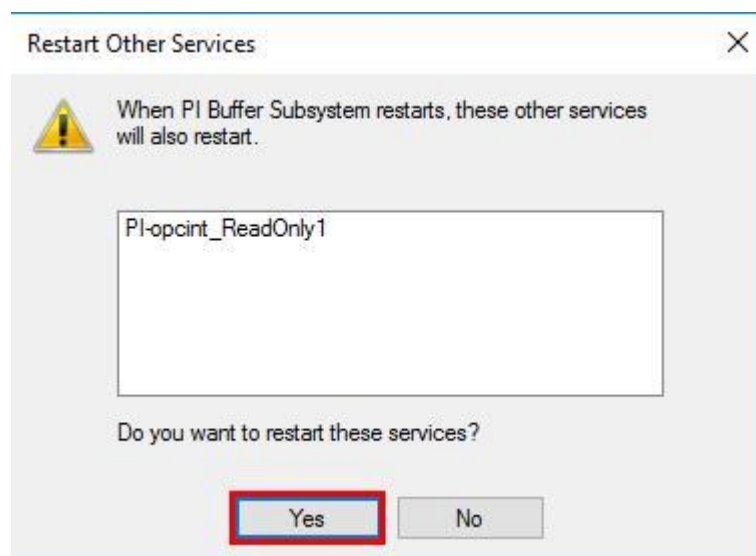
Open windows services from the taskbar.



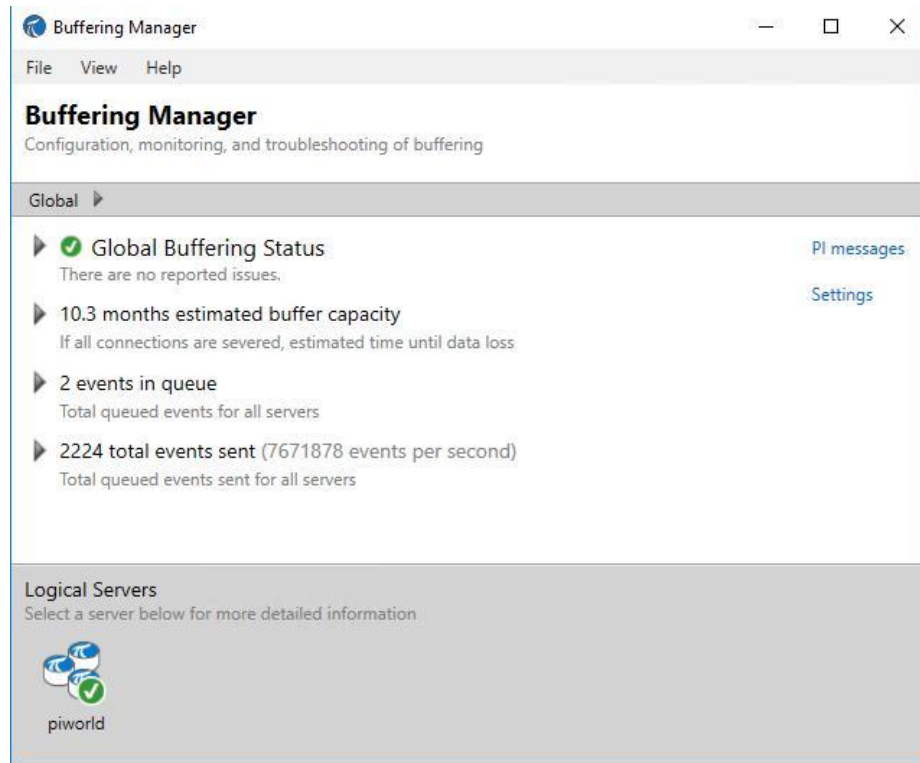
Restart PI Buffer Subsystem



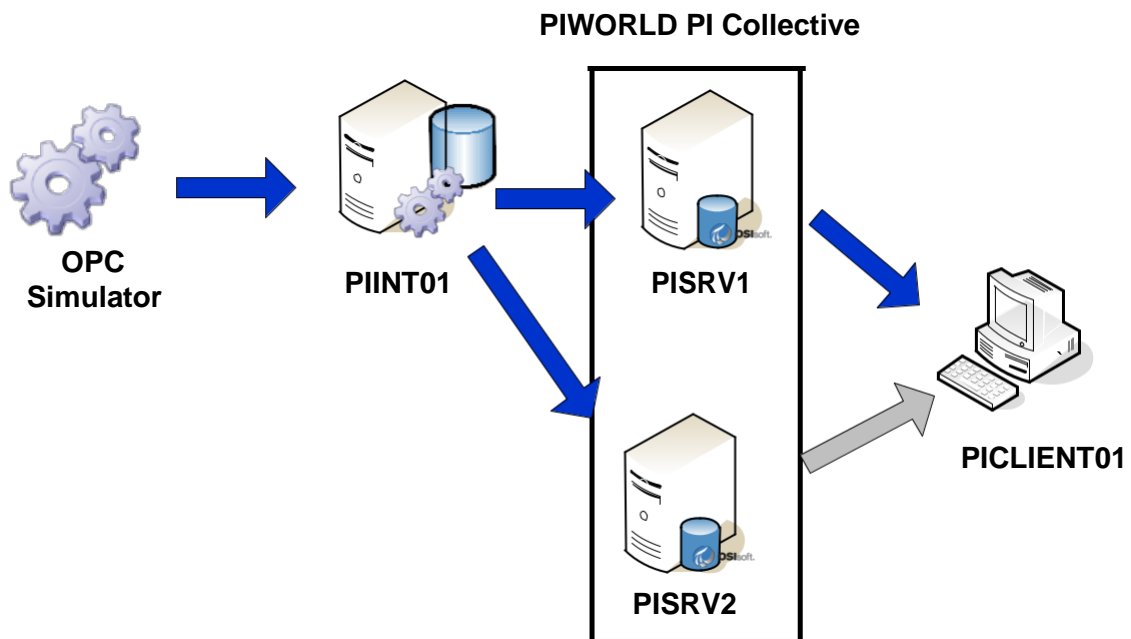
The wizard secretly added a dependency on PI Buffer Subsystem to the OPC Interface service, so it must restart too.



All Green!



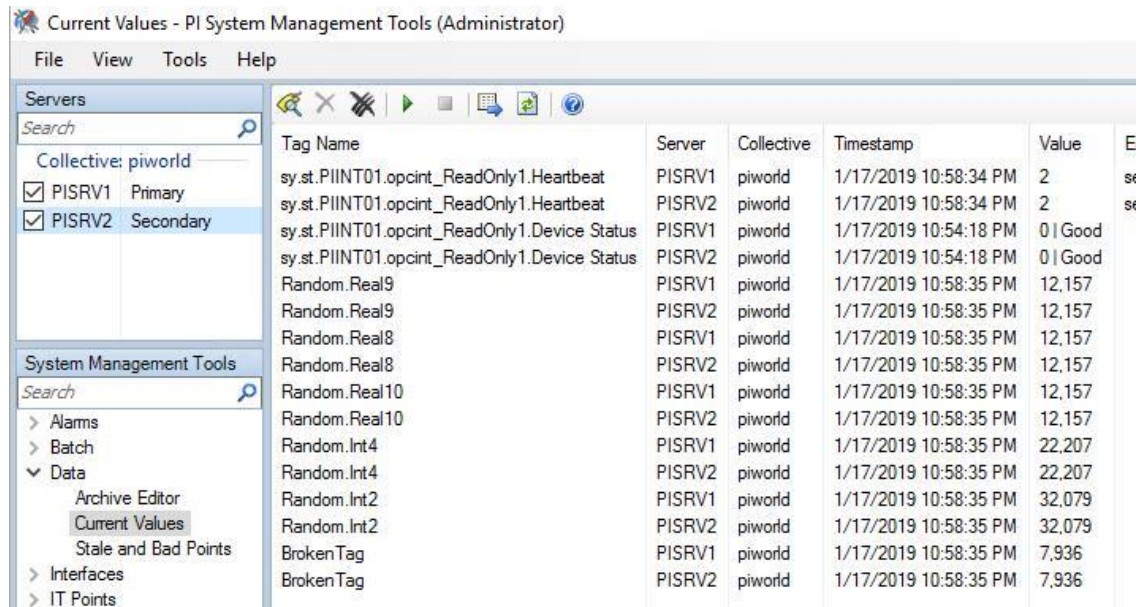
Now that Buffering is configured, the OPC Interface is sending new data to PISRV1 and PISRV2.



e. Verify Correct Operation

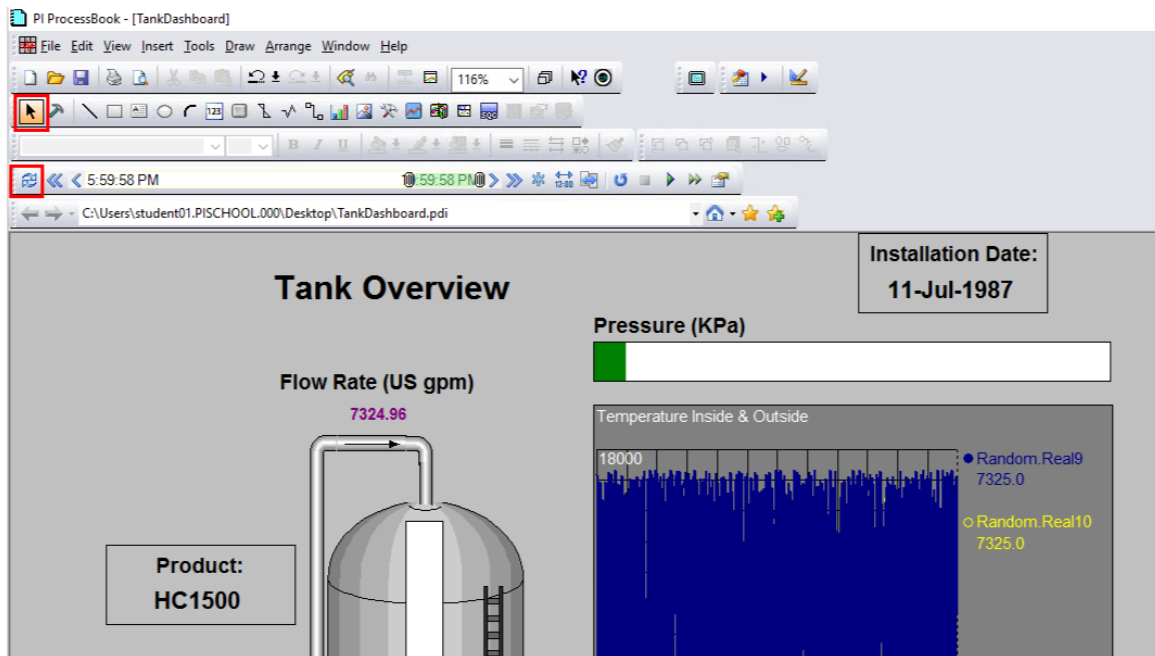
On PICLIENT01, Go to PI SMT, connect to PISRV1 and PISRV2, and add all OPC tags to the **Current Values** plugin. We should see identical timestamps and values on both servers now.

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

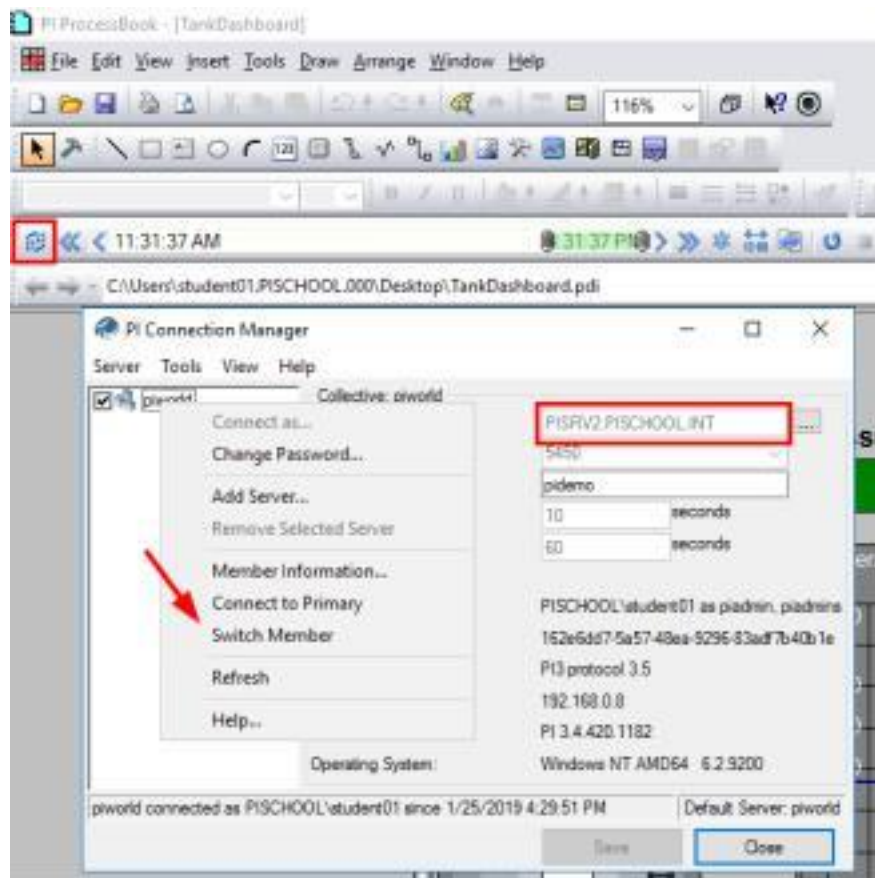


Tag Name	Server	Collective	Timestamp	Value
sy.st.PIINT01.opcint_ReadOnly1.Heartbeat	PISRV1	piworld	1/17/2019 10:58:34 PM	2
sy.st.PIINT01.opcint_ReadOnly1.Heartbeat	PISRV2	piworld	1/17/2019 10:58:34 PM	2
sy.st.PIINT01.opcint_ReadOnly1.Device Status	PISRV1	piworld	1/17/2019 10:54:18 PM	0 Good
sy.st.PIINT01.opcint_ReadOnly1.Device Status	PISRV2	piworld	1/17/2019 10:54:18 PM	0 Good
Random.Real9	PISRV1	piworld	1/17/2019 10:58:35 PM	12,157
Random.Real9	PISRV2	piworld	1/17/2019 10:58:35 PM	12,157
Random.Real8	PISRV1	piworld	1/17/2019 10:58:35 PM	12,157
Random.Real8	PISRV2	piworld	1/17/2019 10:58:35 PM	12,157
Random.Real10	PISRV1	piworld	1/17/2019 10:58:35 PM	12,157
Random.Real10	PISRV2	piworld	1/17/2019 10:58:35 PM	12,157
Random.Int4	PISRV1	piworld	1/17/2019 10:58:35 PM	22,207
Random.Int4	PISRV2	piworld	1/17/2019 10:58:35 PM	22,207
Random.Int2	PISRV1	piworld	1/17/2019 10:58:35 PM	32,079
Random.Int2	PISRV2	piworld	1/17/2019 10:58:35 PM	32,079
BrokenTag	PISRV1	piworld	1/17/2019 10:58:35 PM	7,936
BrokenTag	PISRV2	piworld	1/17/2019 10:58:35 PM	7,936

Go back to the Tank Dashboard and click Revert while in run mode, everything should have data.



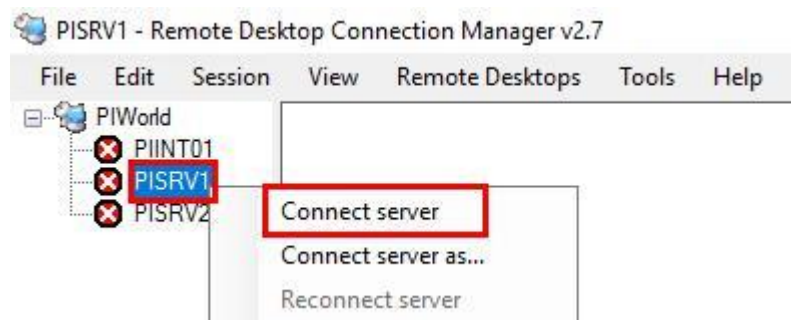
Confirm that you're connected to PISRV2 in File -> Connections. If not, switch members and click Revert again.



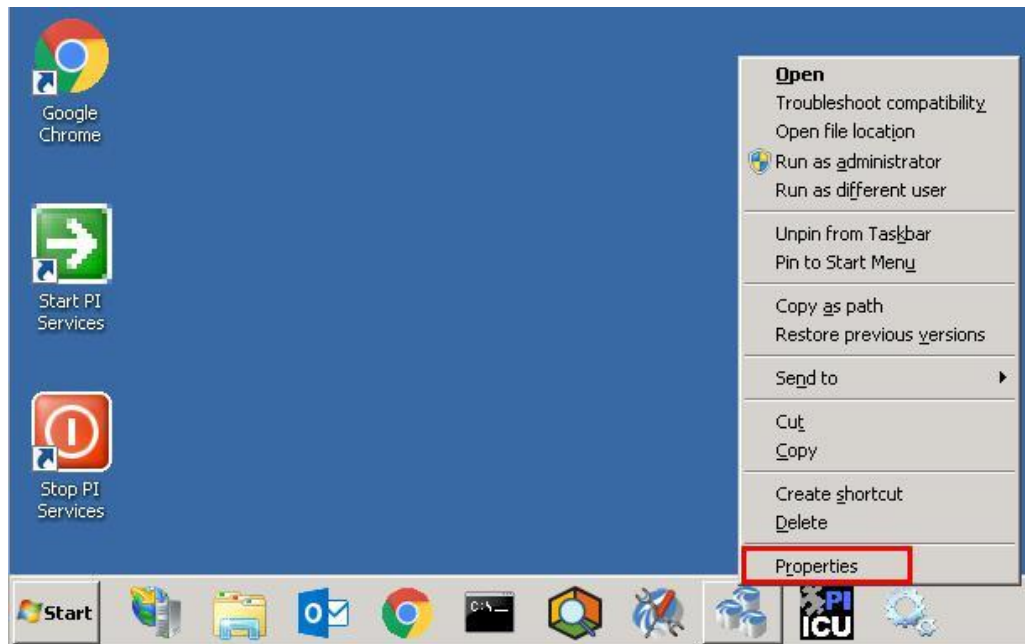
f. Copy Archive data to PISRV2 by Reinitializing the PI Collective

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 **connect to PISRV1** and Reinitialize the PI Collective.

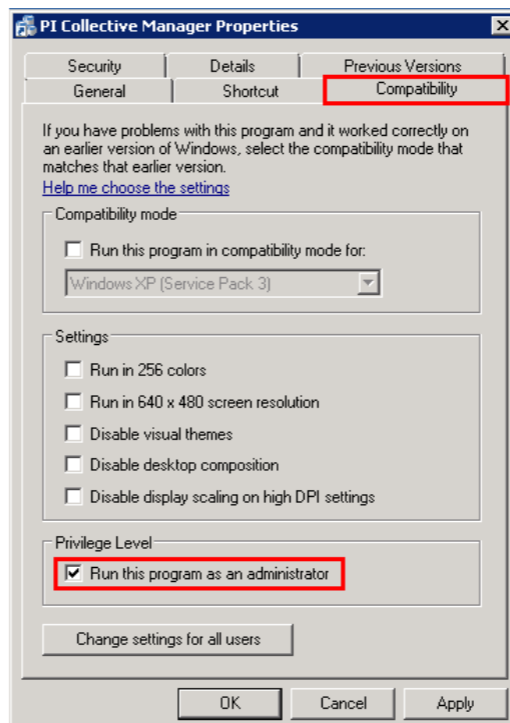


Before Reinitializing, let's implement a quick UI tweak. Hold shift while right-clicking PI Connection Manager in the taskbar and go to properties:



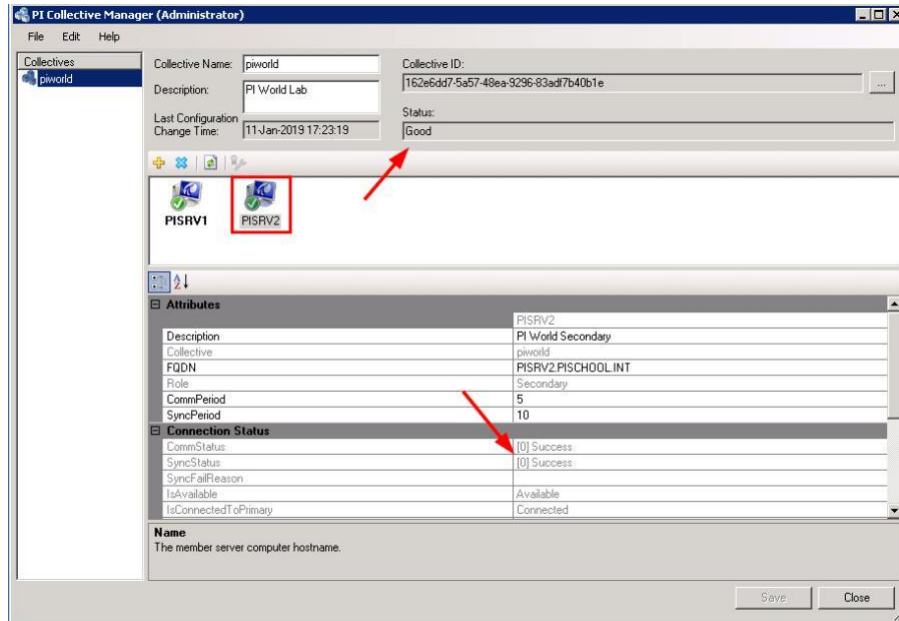
In the compatibility tab, check 'Run this program as an administrator'. You can of course just remember to Run as administrator every time instead, but this is a little more foolproof.

Reinitialization may fail mid-way if PI Collective Manager is not run as an administrator.



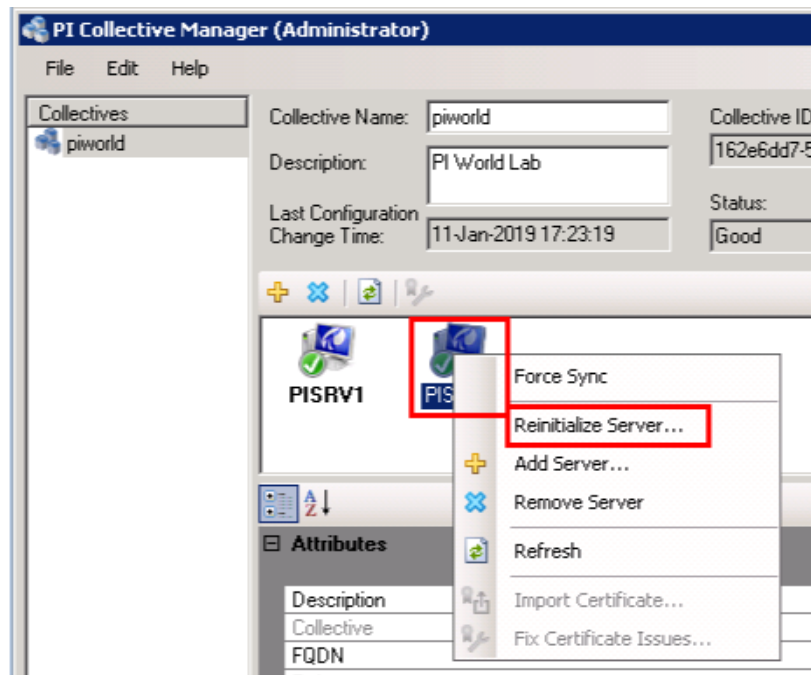
Launch PI Collective Manager from the taskbar and resize the window.

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.



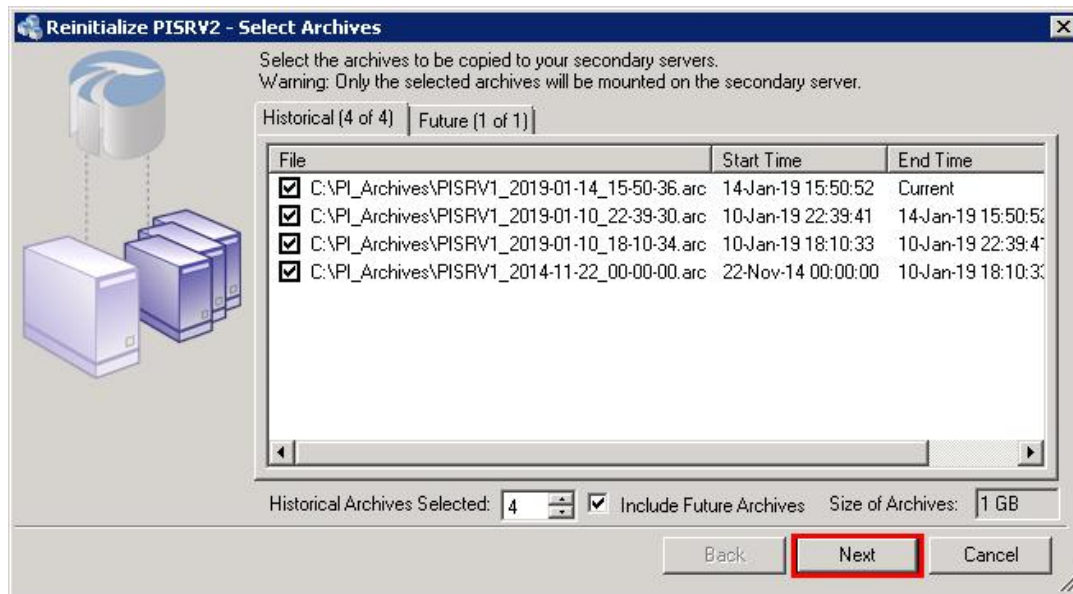
Provided the administrator has the necessary permissions, Reinitialization is fairly straightforward.

Right-click PISRV2 and Reinitialize Server...



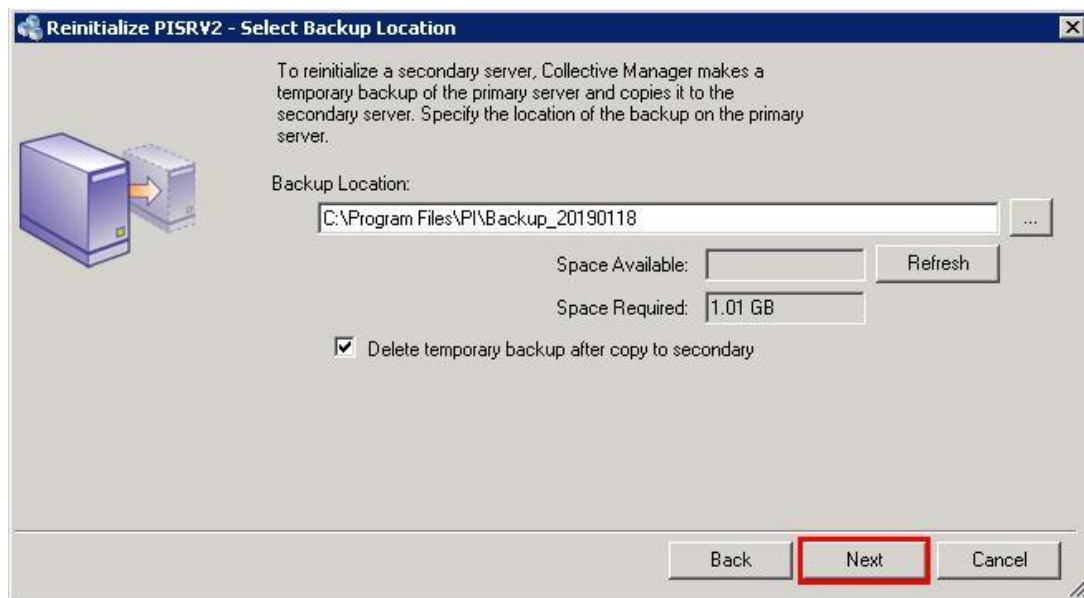
At this point you have the option to choose which Archive files are copied. Typically, you only have to go as far back as the missing data. In a production environment, you may choose to copy only the first archive using the software and manually copy and register the remaining archives, since there 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.

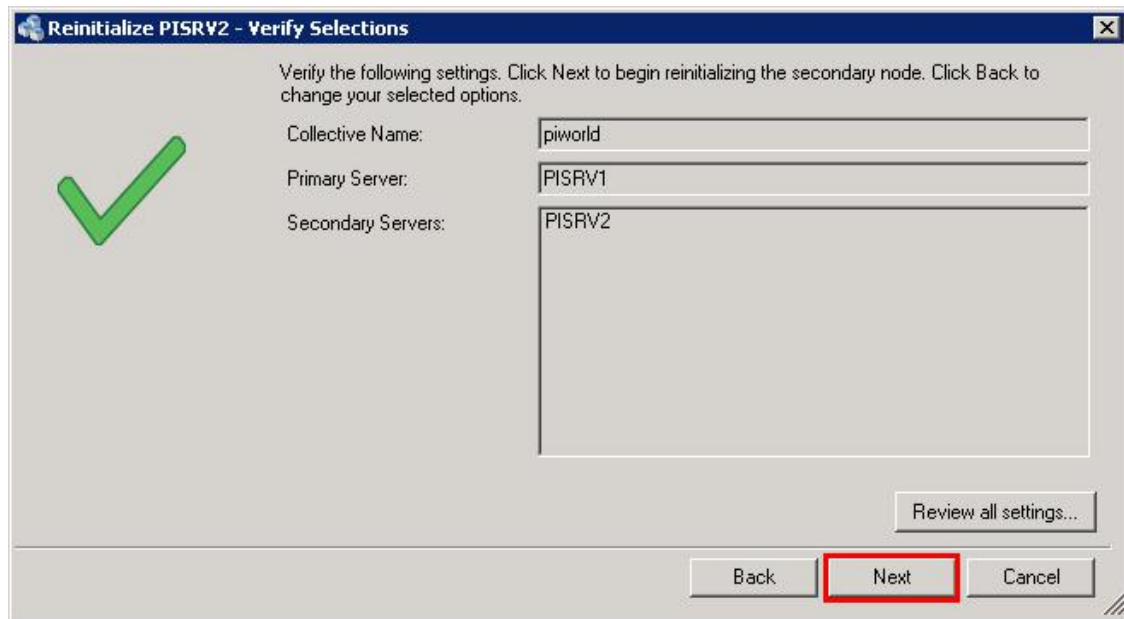


Reinitialization simply takes a backup of the Primary and restores it to the Secondary. This next screen lets you decide what to do with the temporary backup.

Again defaults are fine. Next.



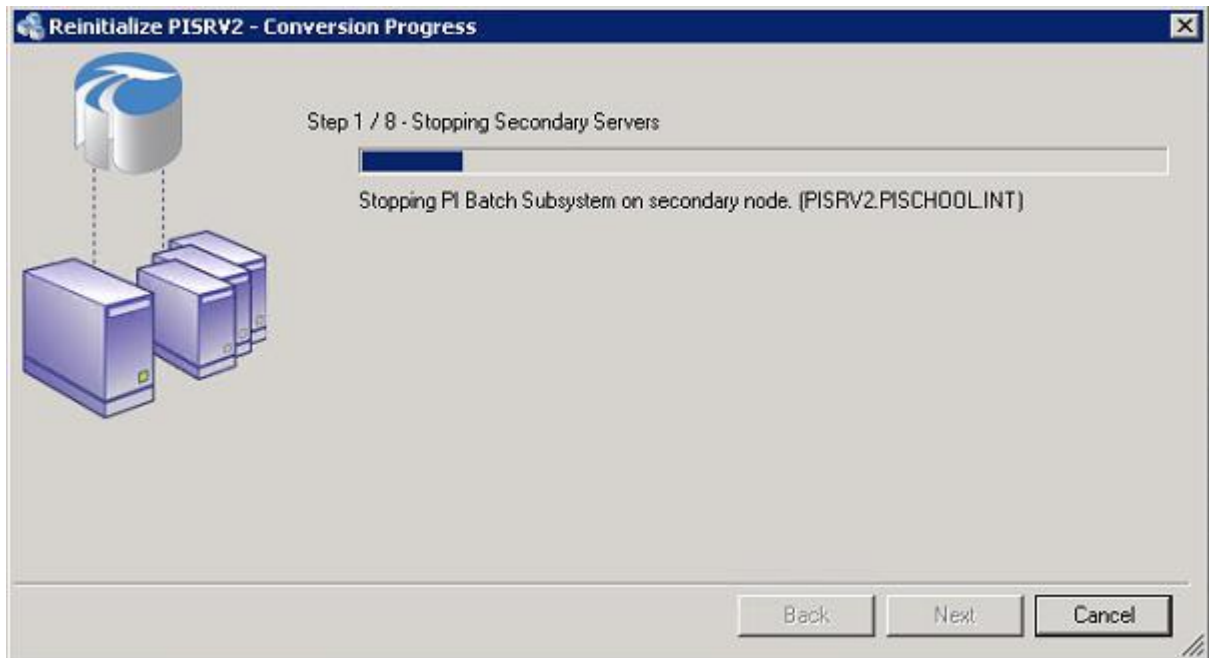
Typically there's no need to review the settings. Leave the defaults and click Next.



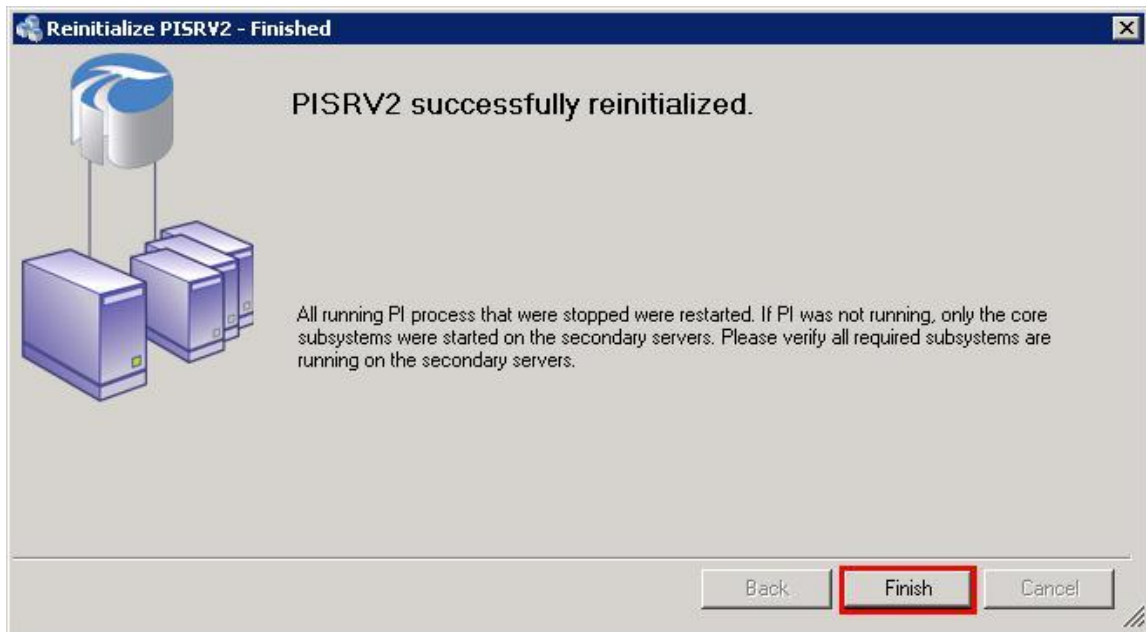
The Reinitialization process will:

- 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

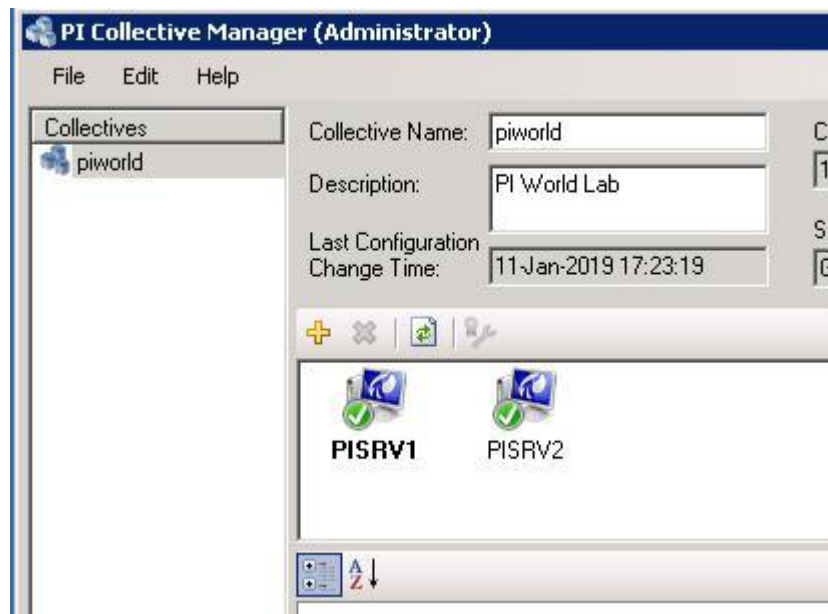
This will take 5 minutes or so.



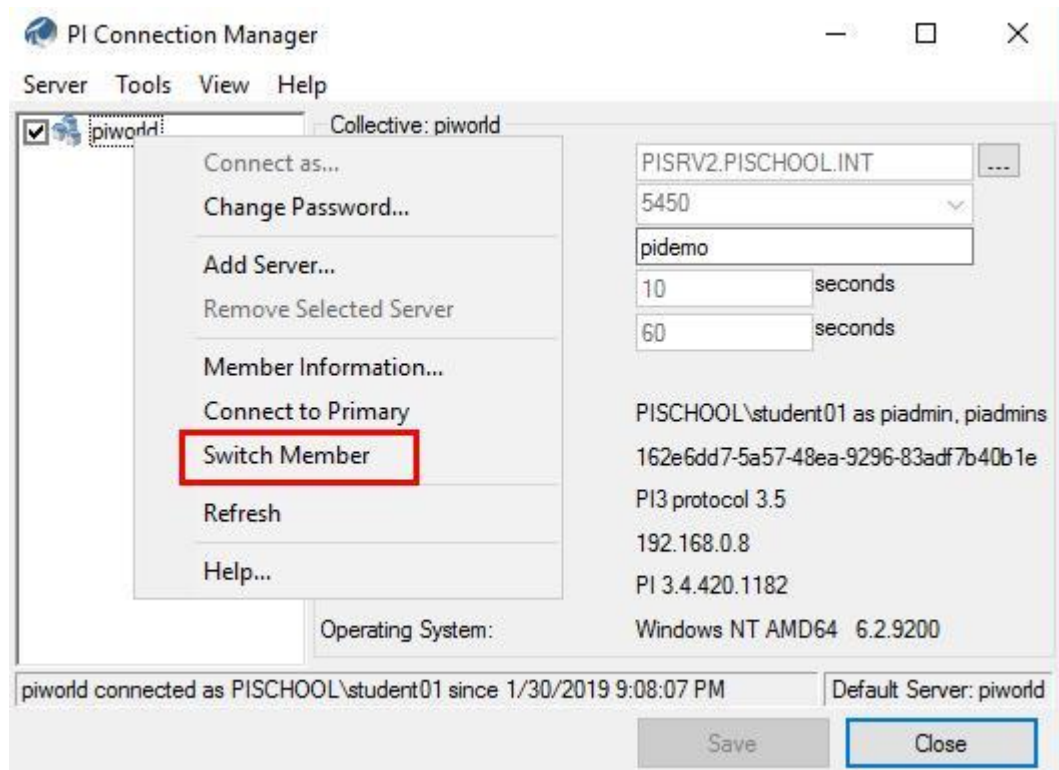
Eventually it should complete successfully.



Status should be green after a few minutes or so of closing the previous window.



Finally, we can verify that the data exists on PISRV2 by switching members in the PI ProcessBook display.



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.
- Someone makes a decision 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

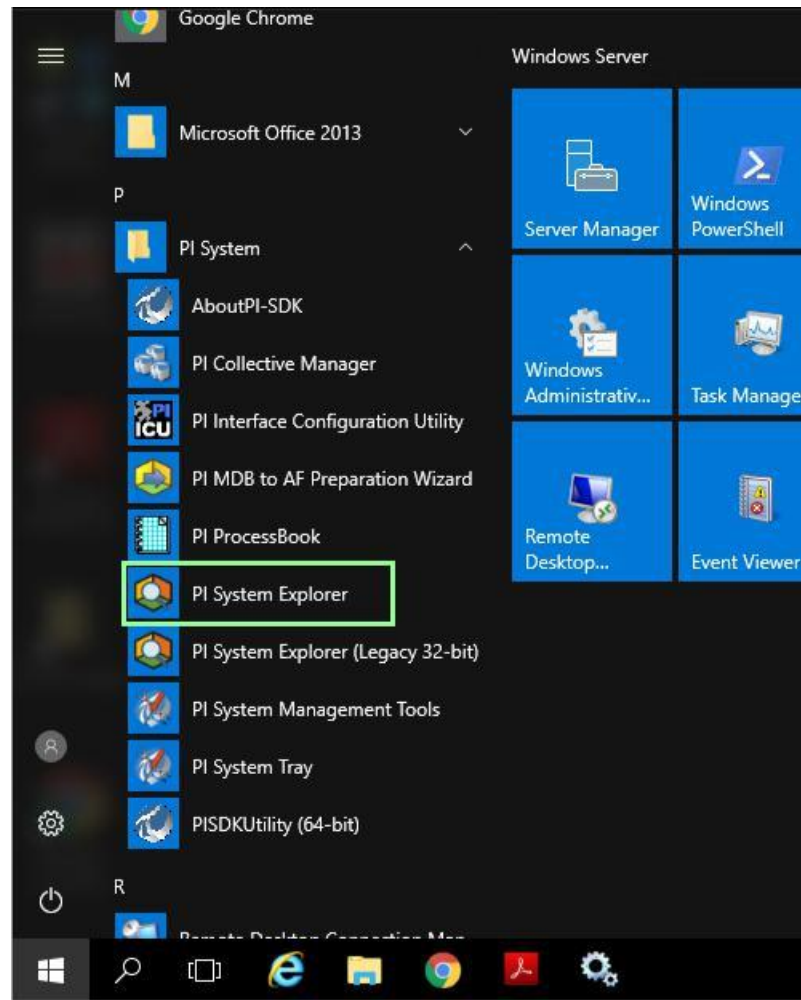
Ultimately, we want to get to the point of setting up a Notification. However, the following prerequisites must be complete first:

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

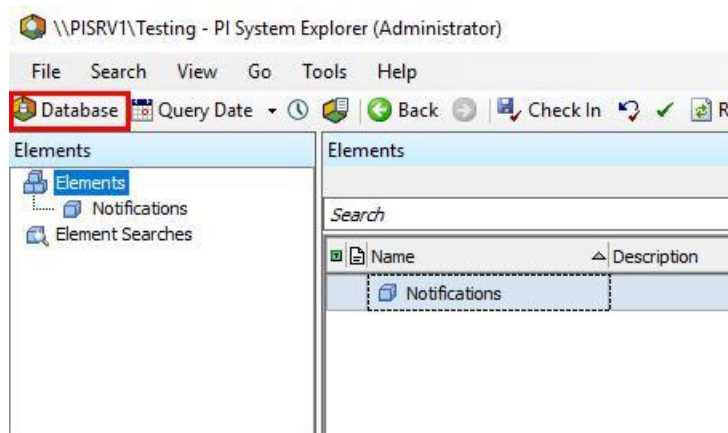
Let's start by choosing a tag to monitor. Random.Int2 is as good as any.

First we'll build a simple PI AF hierarchy and map Random.Int2.

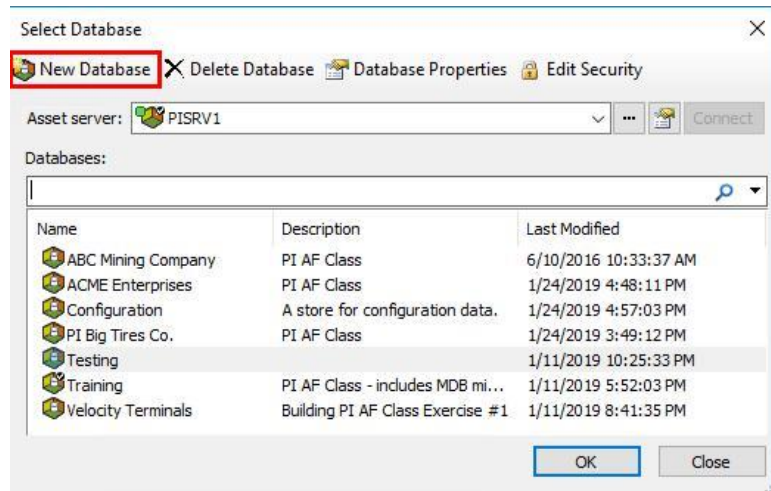
On PICLIENT01, Open PI System Explorer.



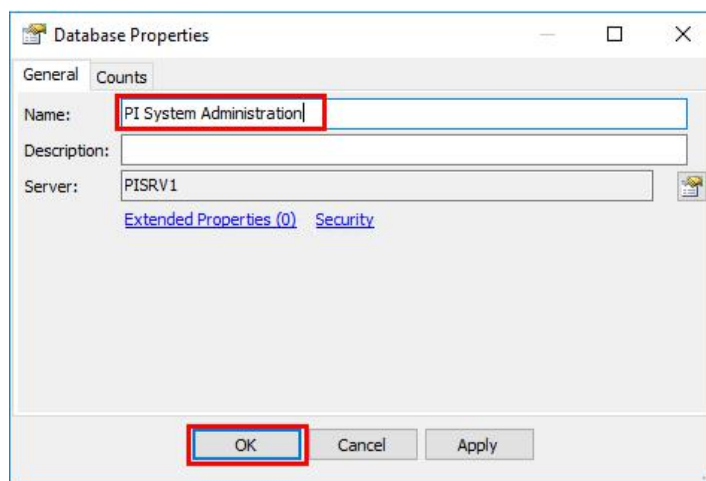
We'll create a new database dedicated to PI System Administration. **Click Database.**



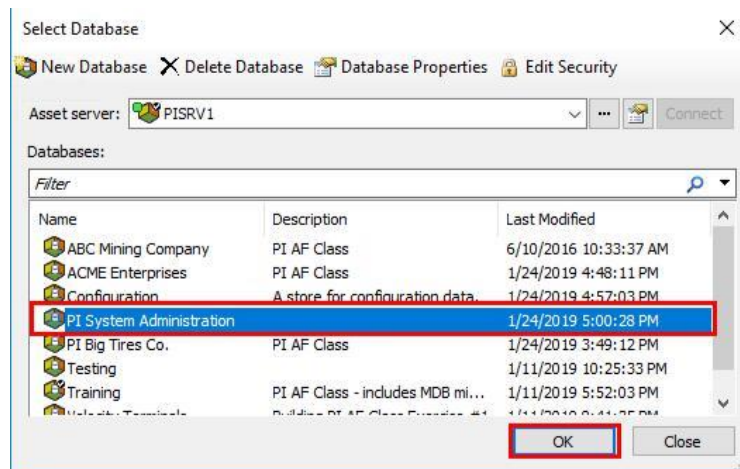
Click **New Database**.



Name it "PI System Administration" and click OK.

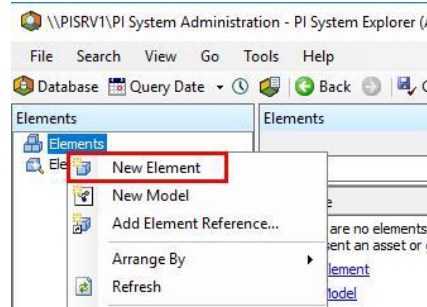


Select your new Database and click OK.

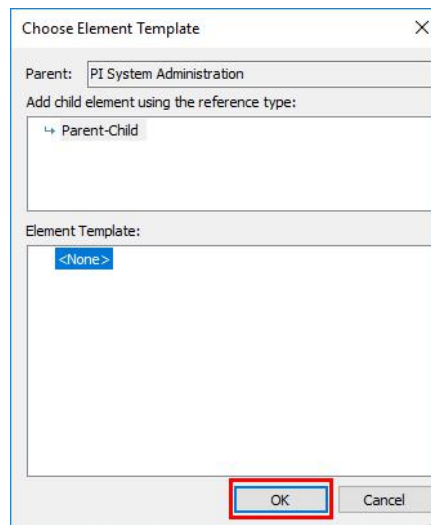


Next, we'll create a New top level Element to contain Stale and Bad Data alerts.

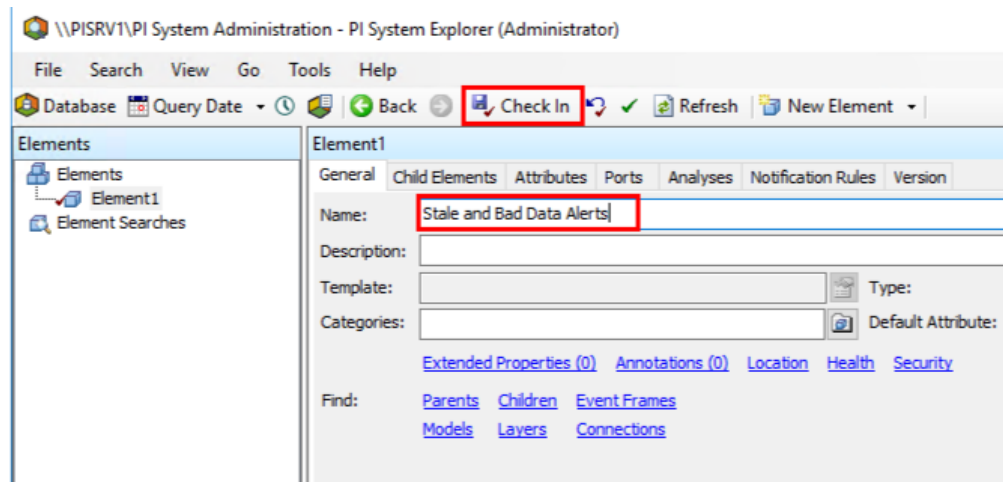
Right-click Elements and select New Element. **If you click the New Element hyperlink, you will see a slightly different set of dialogs, which will not match the workbook exactly.**



Leave the defaults and click OK.

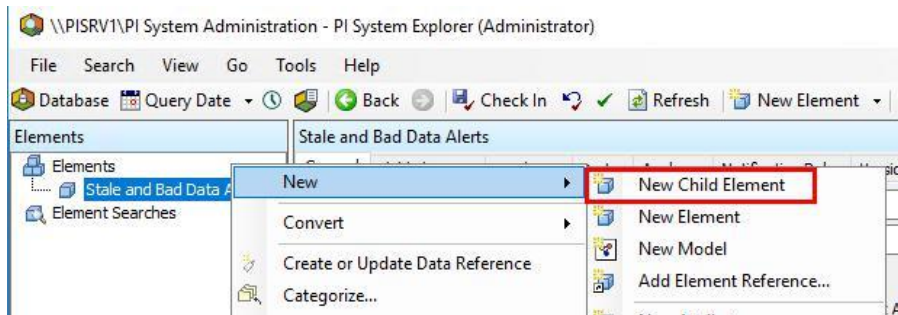


Enter Stale and Bad Data Alerts as the name and click Check In.

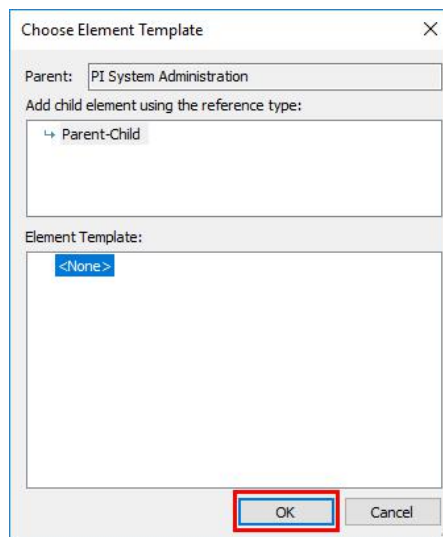


Now we'll create an element for Random.Int2.

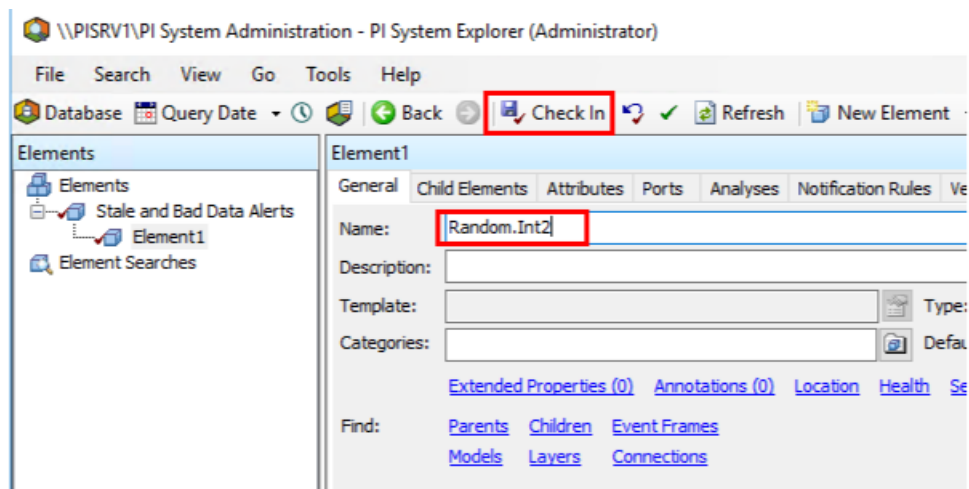
Right-click the Stale and Bad Data Alerts Element and create a new child Element.



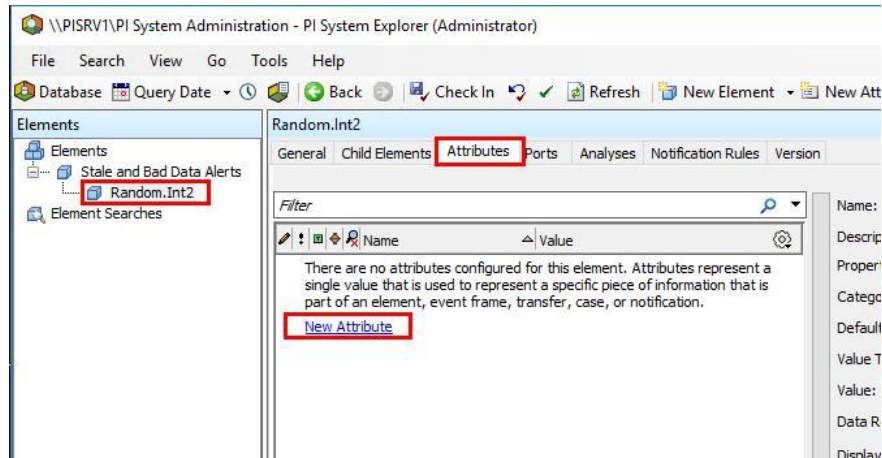
Again, leave the defaults and click OK.



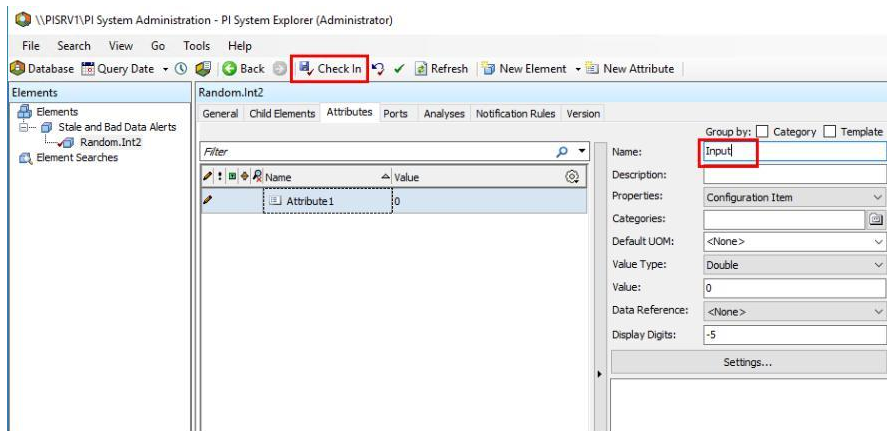
Name the Element Random.Int2 and check in.



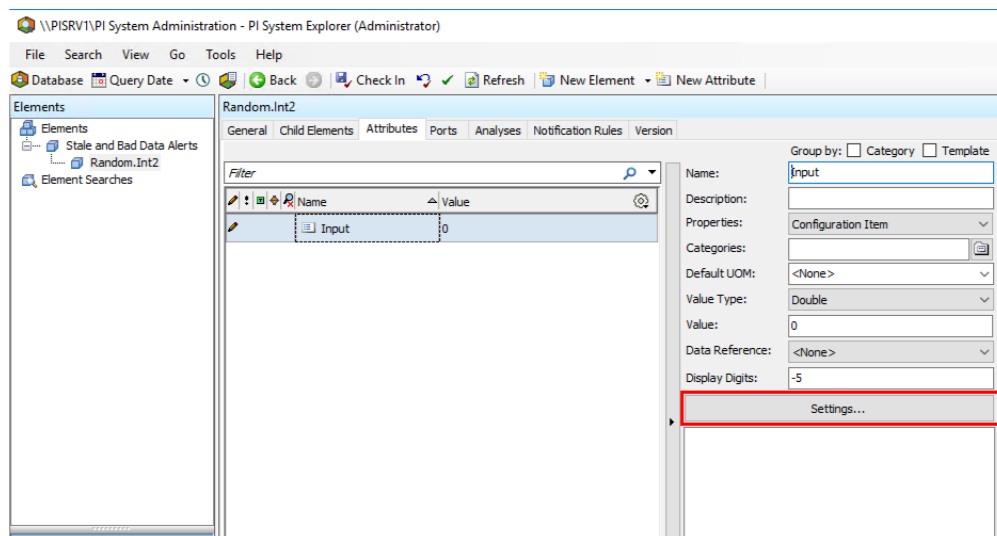
With the Random.Int2 Element selected, select the Attributes tab, and create a new attribute.



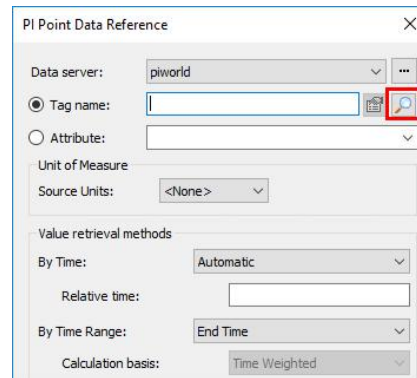
Name the attribute **Input** and check in.



Click Settings.

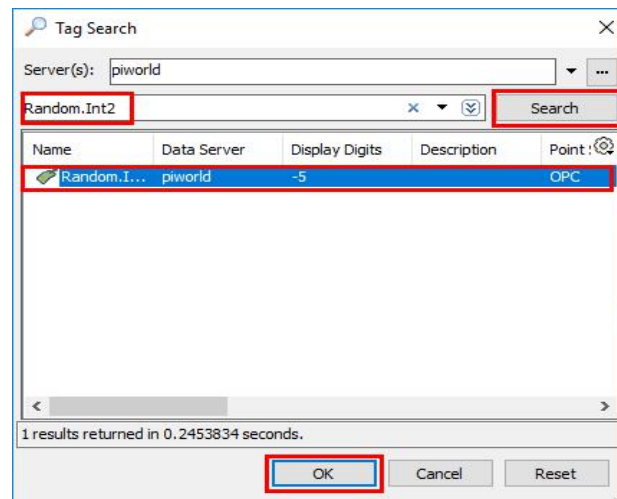


Click the search icon next to the tag name field.



The PI Point Data Reference dialog box is shown. The 'Tag name' field is selected, and the search icon (magnifying glass) next to it is highlighted with a red box. Other fields include 'Data server' (piworld), 'Attribute' (empty), 'Unit of Measure' (empty), 'Source Units' (<None>), 'Value retrieval methods' (Automatic), 'Relative time' (empty), 'By Time Range' (End Time), and 'Calculation basis' (Time Weighted).

Search for Random.Int2, select it, and click OK.

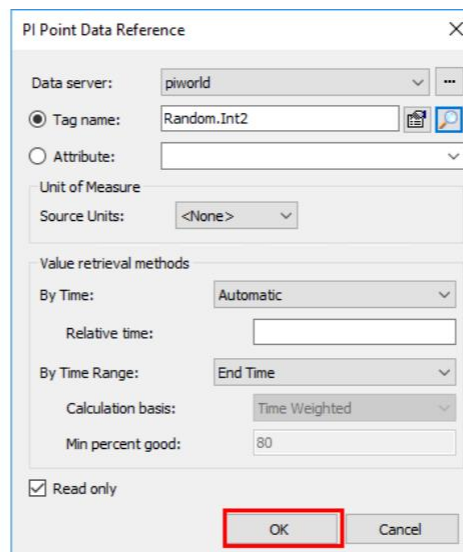


The Tag Search dialog box is shown. The 'Server(s)' field is set to 'piworld'. The search text 'Random.Int2' is entered in the search bar, and the 'Search' button is highlighted with a red box. Below the search bar is a table with the following data:

Name	Data Server	Display Digits	Description	Point
Random.I...	piworld	-5		OPC

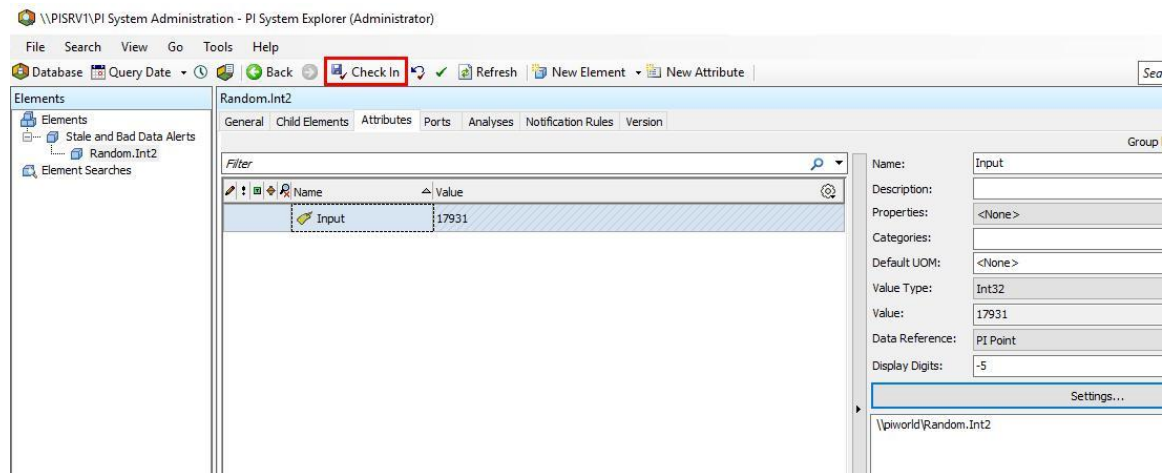
The first row is highlighted with a red box. At the bottom, it says '1 results returned in 0.2453834 seconds.' and the 'OK' button is highlighted with a red box.

Random.Int2 should now be in the Tag name field. Of course, we could have just typed "Random.Int2" here directly without searching. Leave the defaults and click OK.



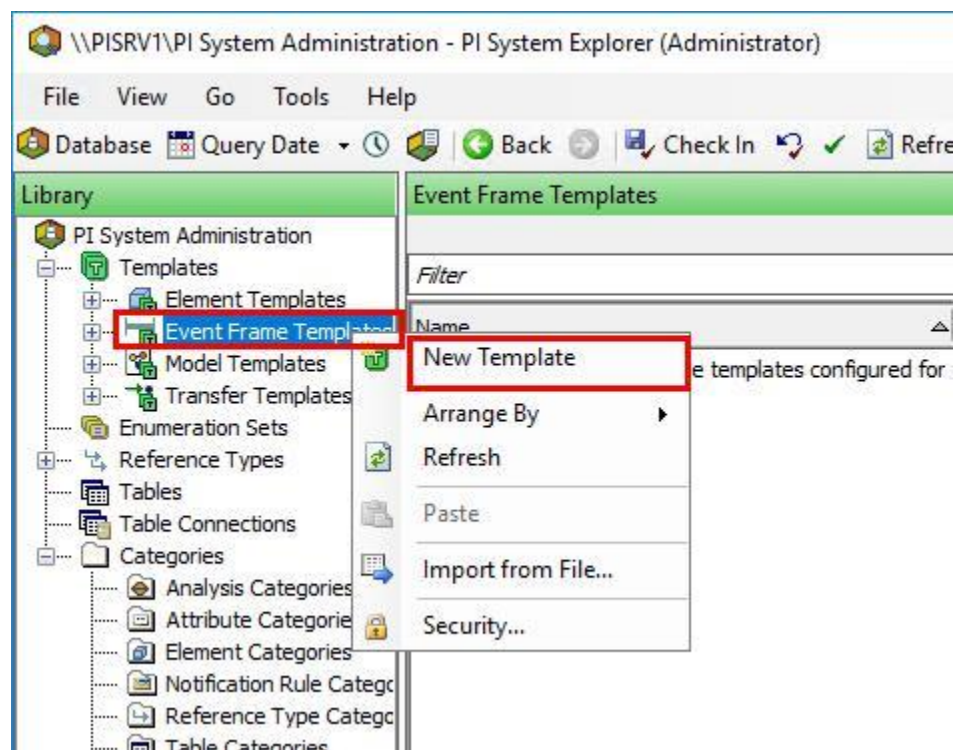
The PI Point Data Reference dialog box is shown again. The 'Tag name' field now contains 'Random.Int2'. The 'Read only' checkbox is checked. The 'OK' button is highlighted with a red box.

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.

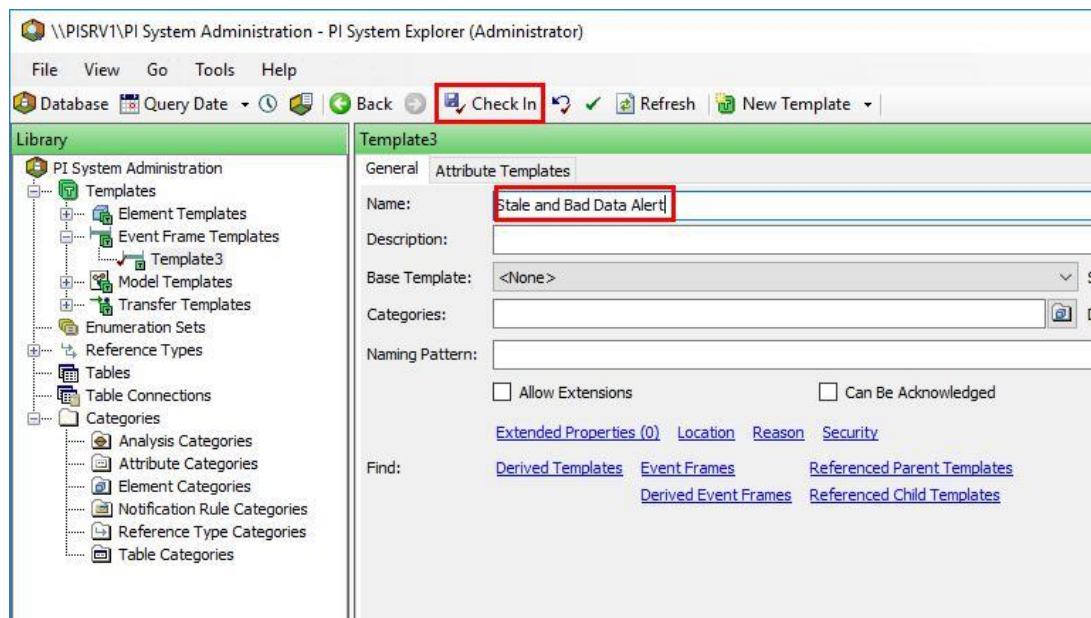


Next we'll create an Event Frame template. If this is your first experience with Event Frames you'll just have to accept the mystery for now. For our purposes it's a prerequisite for configuring PI Notifications that can't be avoided. We just don't have time to cover Event Frames in sufficient detail as the learning curve is too steep.

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

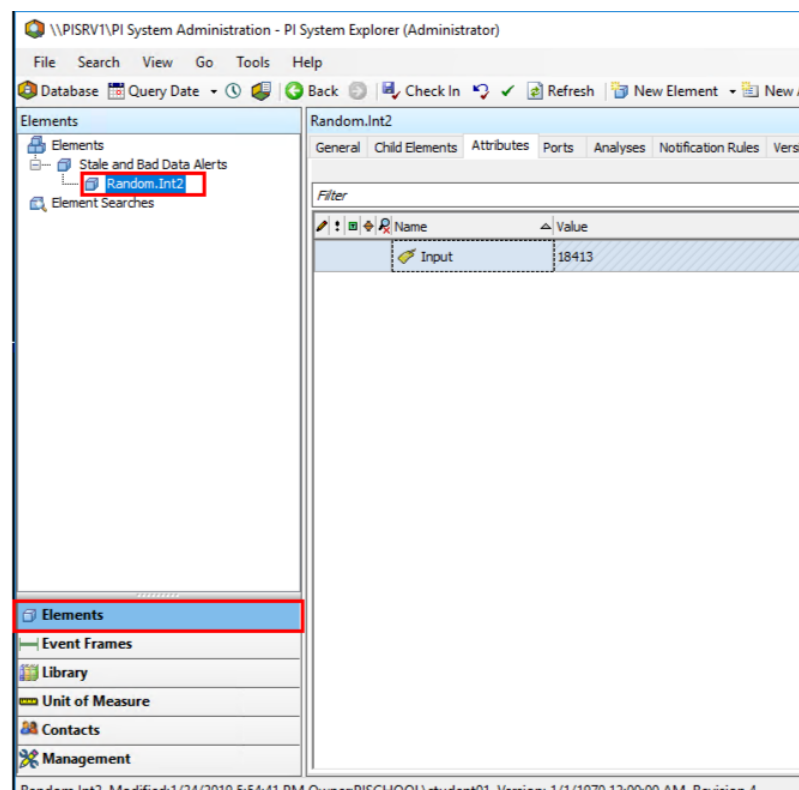


Name it Stale and Bad Data Alert, Check in

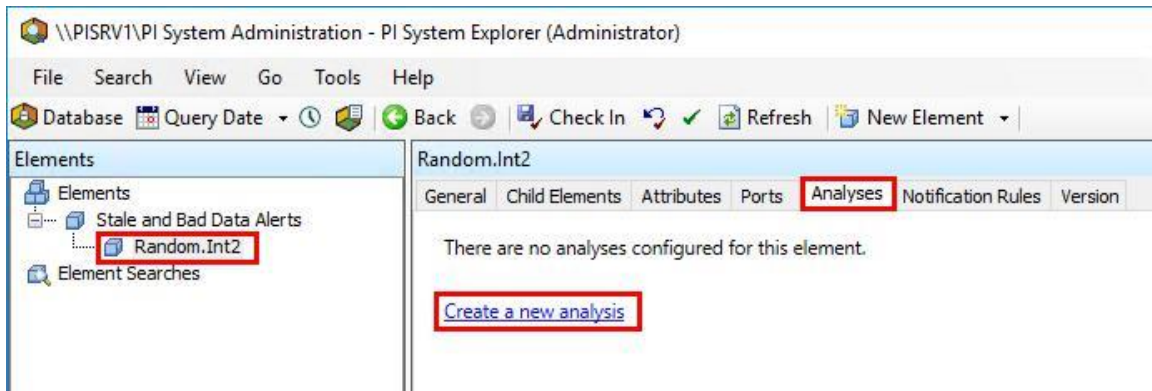


The Library warrants some explanation. This is where all the Templates and other building blocks used in PI AF are configured.

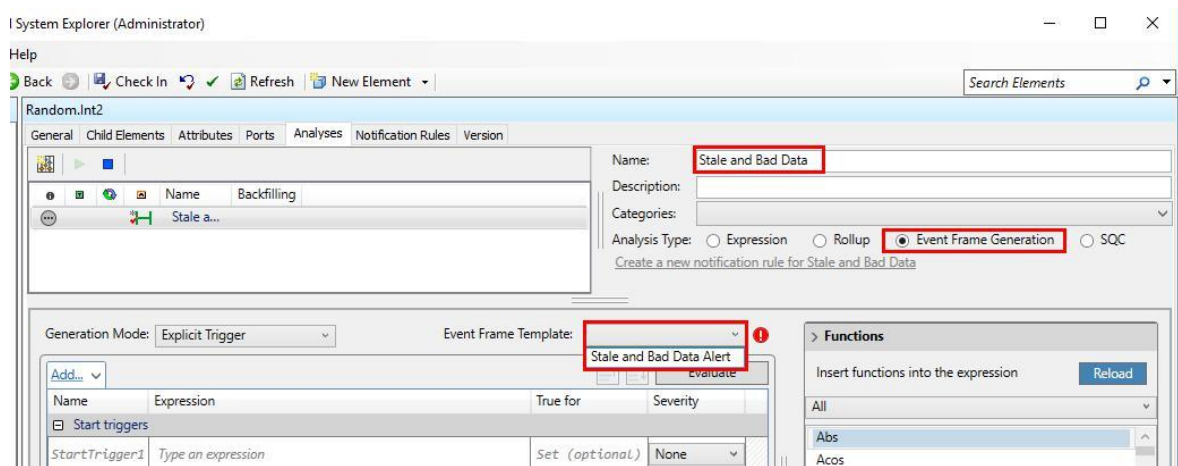
Next we'll configure the logic that defines Stale and Bad data. Go back to Elements and select the Random.Int2 Element.



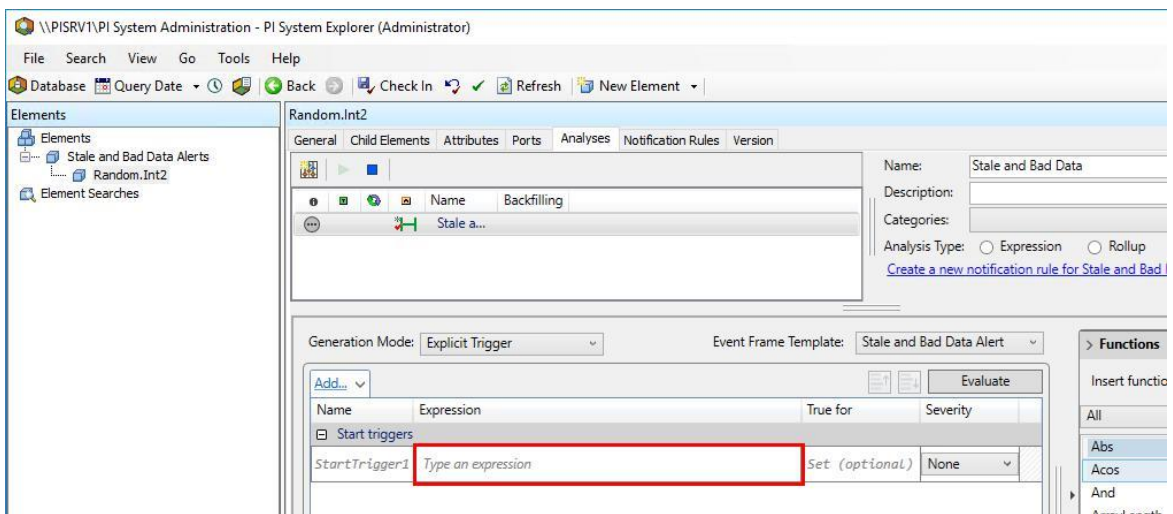
Click the Analyses tab and Create a new analysis.



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

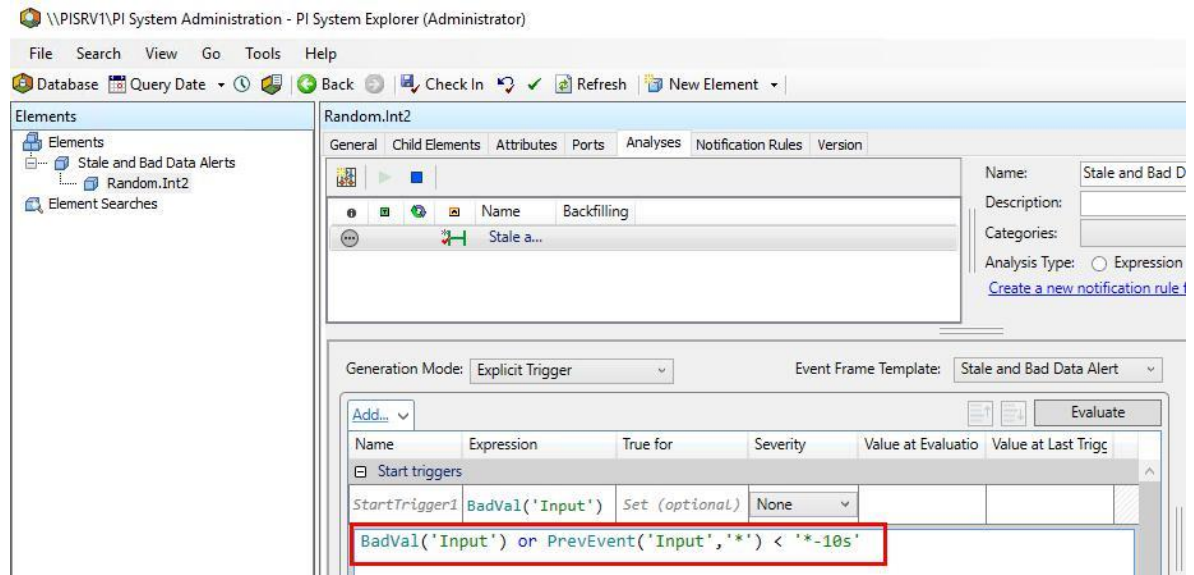


Click on the field that says Type an expression.



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'
```



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.Int2) 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

Add... Evaluate

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

Advanced Event Frame Settings...

Scheduling: ☐ Event-Triggered ☒ **Periodic**

Period: 00h 05m 00s **Configure**

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

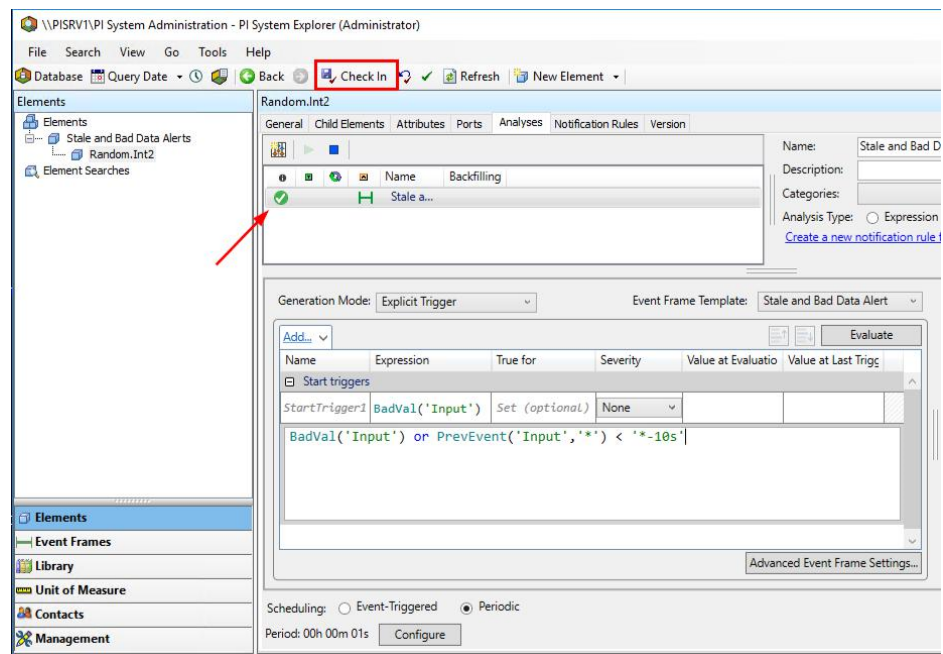
Period
 Specify the amount of time between evaluations.
 00 h 00 m 01 s

☐ Specify Offset

Example evaluation times
 1/24/2019 12:00:00 AM
 1/24/2019 12:00:01 AM
 1/24/2019 12:00:02 AM

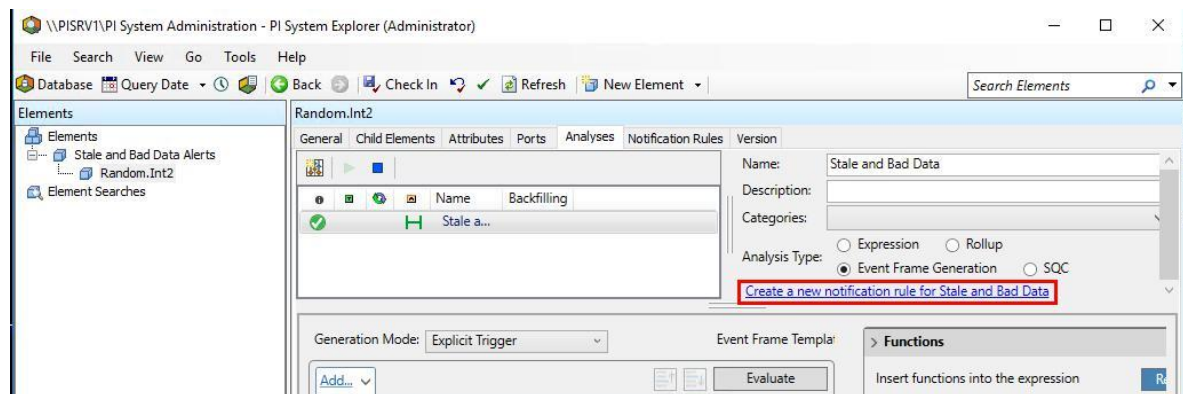
OK Cancel

Check In, the Analysis should start after a few seconds.

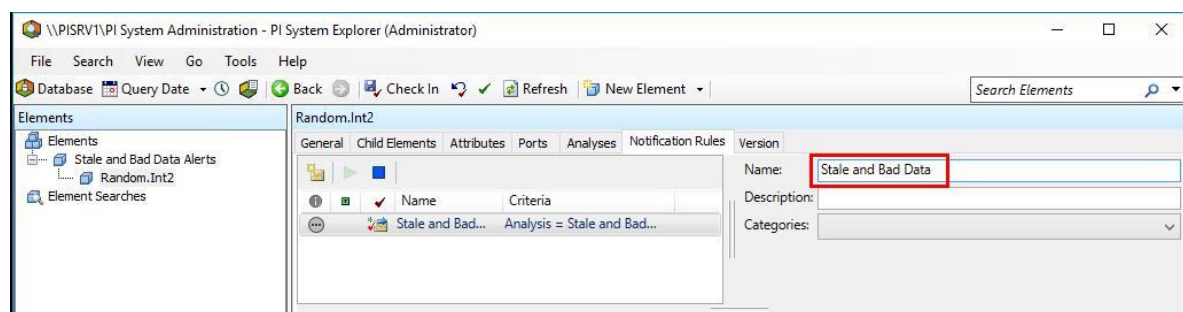


Now to attach a Notification to the Event Frame.

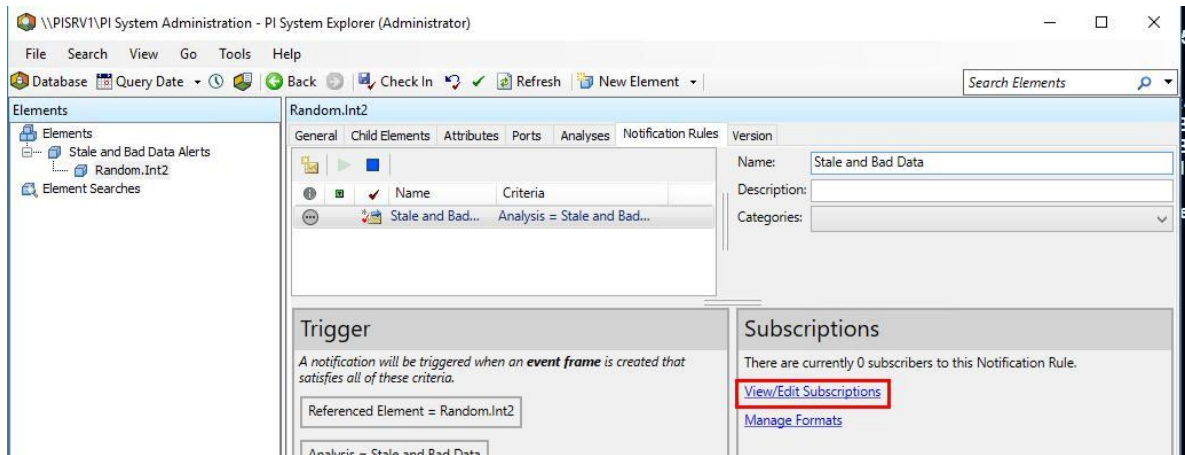
Click 'Create a new notification rule for Stale and Bad Data'.



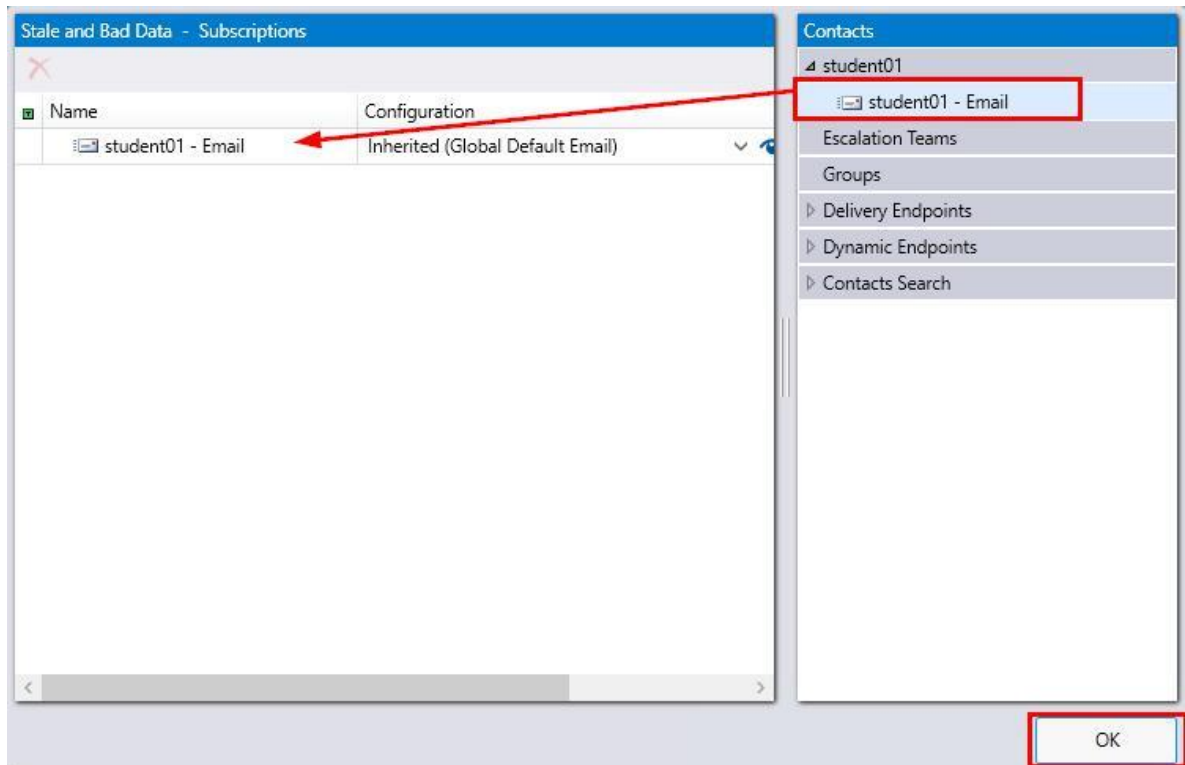
Name it Stale and Bad Data.



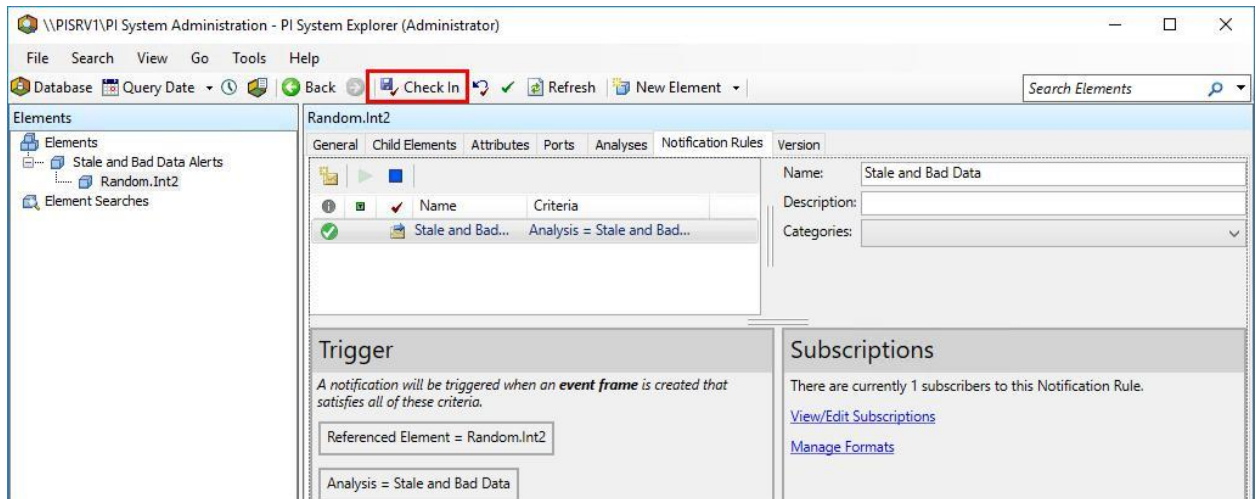
Click View/Edit Subscriptions to add a subscriber.



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

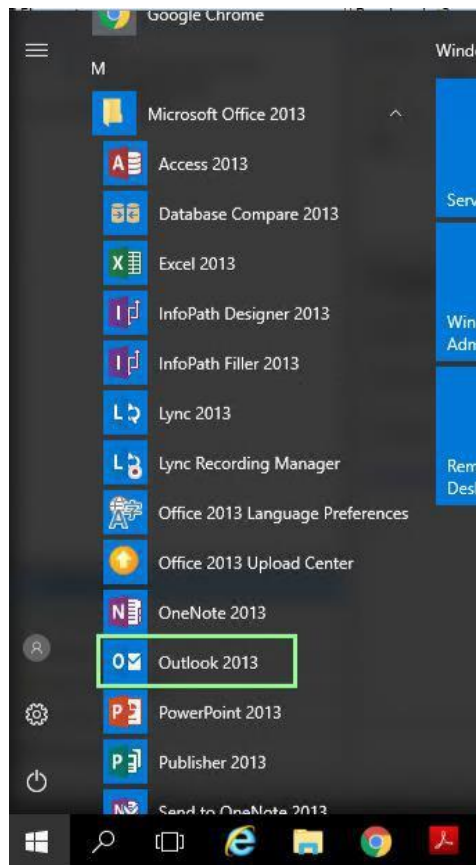


Check in, the Notification Rule should start after a few seconds.

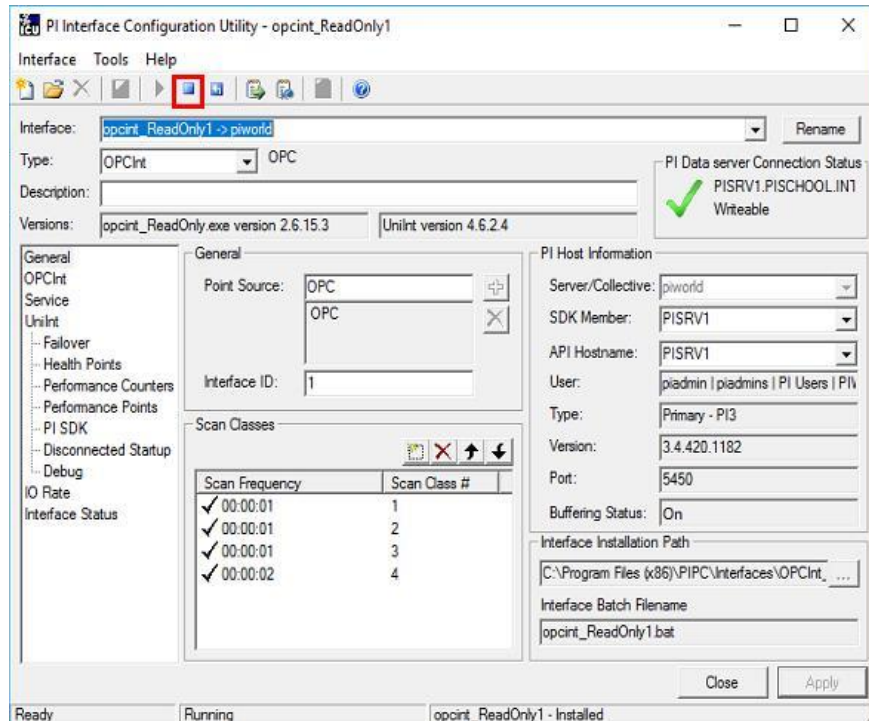


Now to test it. In a production environment you'd use a less intrusive method, but we're just going to stop the OPC Interface on PIINT01, then wait a little more than 10 seconds and see if we get an email.

First open Outlook on PICLIENT01.

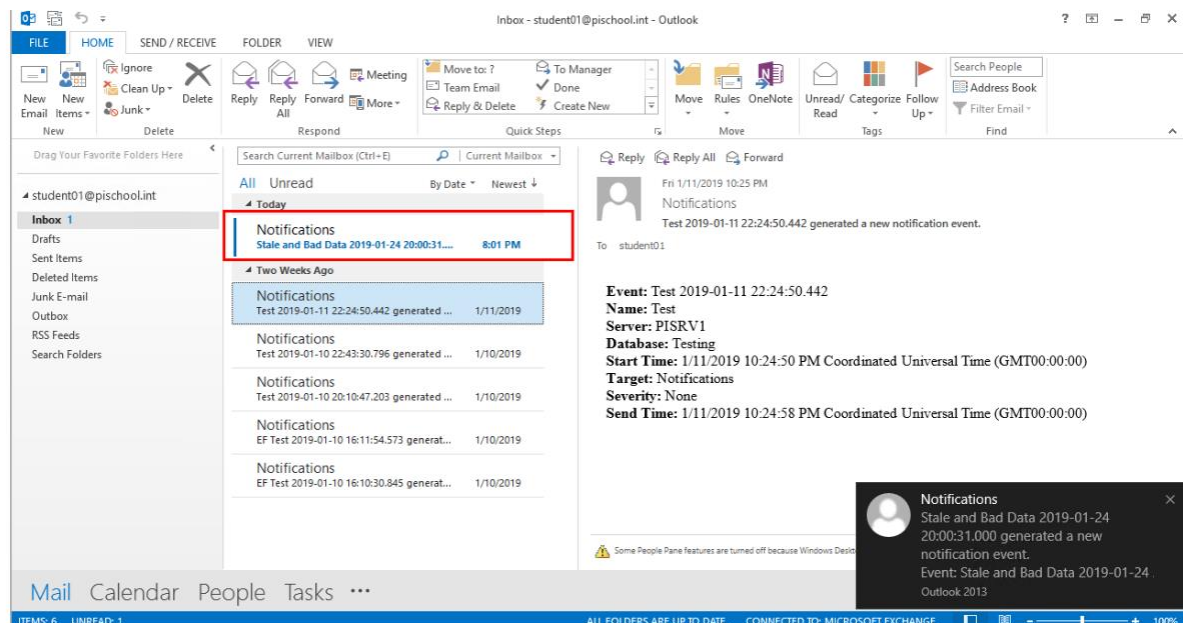


Connect to PIINT01, open PI ICU, select opcint_ReadOnly1, and stop the interface service.

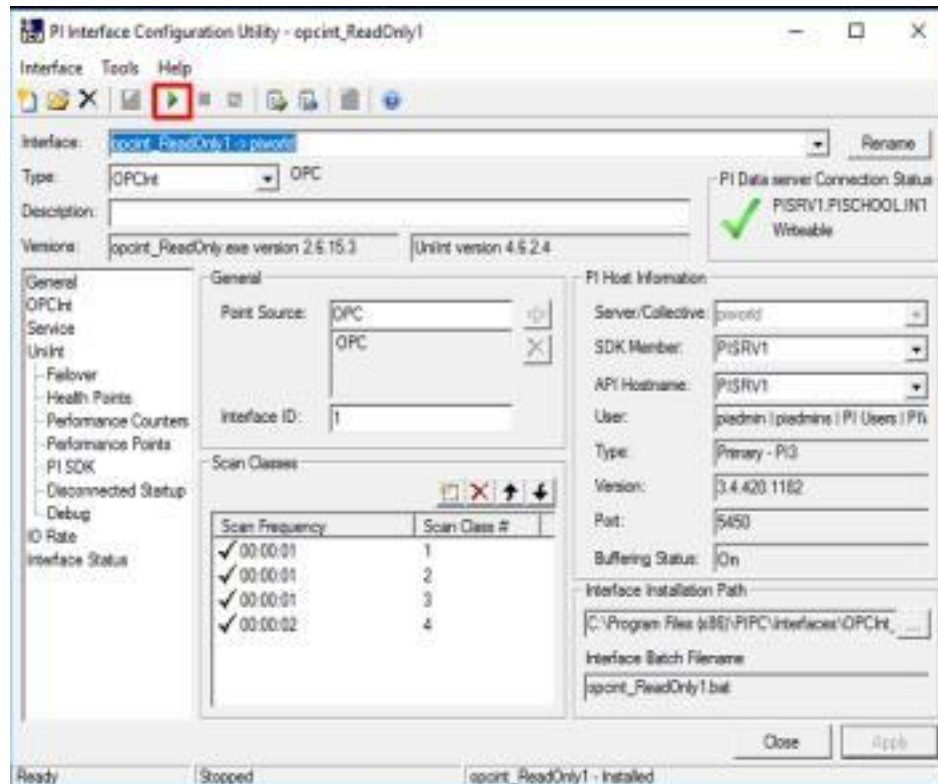


The Notification should trigger after 10 seconds, but it will take a few more seconds to propagate through the Exchange server.

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



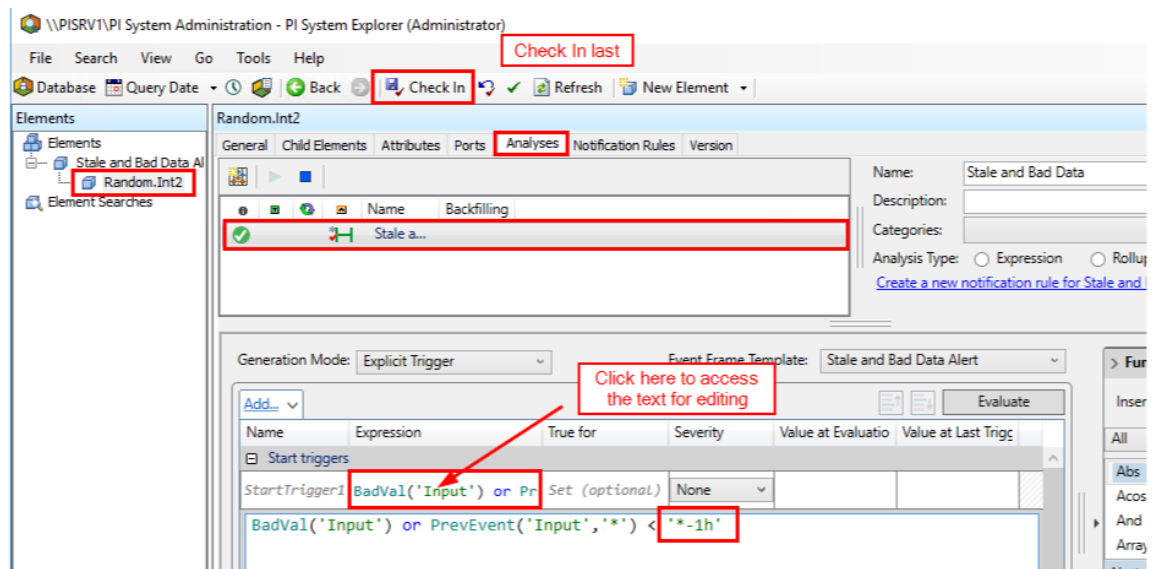
Now let's start the OPC Interface from PIINT01 to get things back to where they were.



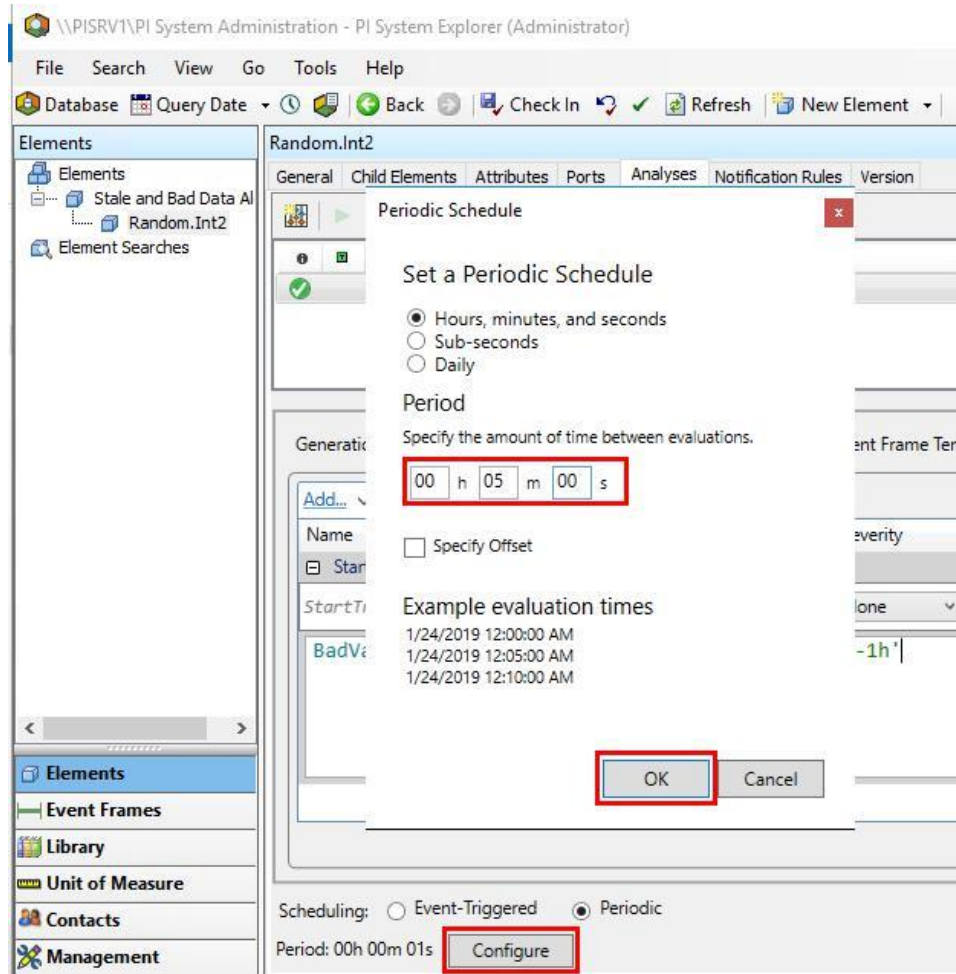
Let's change the triggering logic so that a Notification is only sent when data is stale for an hour.

Go back to PI System Explorer on PICLIENT01. With the Random.Int2 Element selected go back to the Analyses tab.

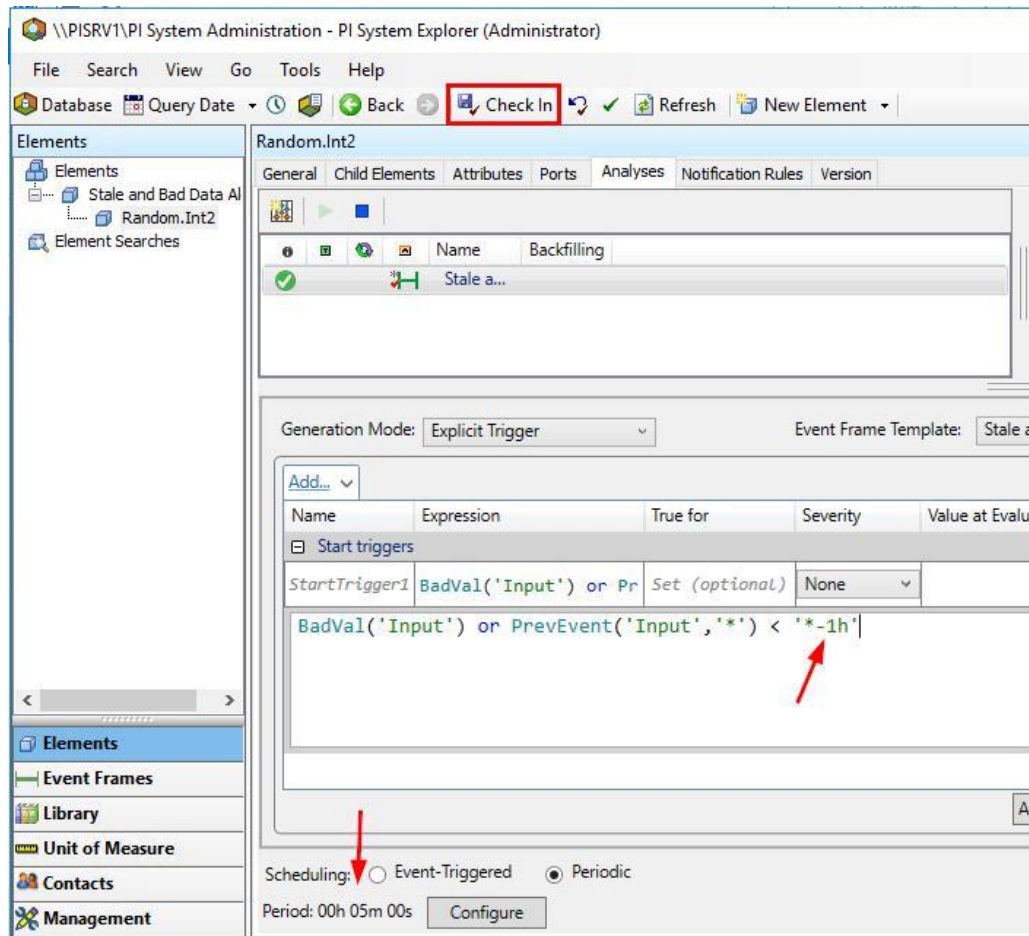
Change 10s to 1h and Check in.



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.



Check in one last time after confirming the edits are reflected.



Congratulations! That's all for today.





Save the Date!

OSIsoft PI World Users Conference in Gothenburg, Sweden. September 16-19, 2019.

Register your interest now to receive updates and notification early bird registration opening.

https://pages.osisoft.com/UC-EMEA-Q3-19-PIWorldGBG-RegisterYourInterest_RegisterYourInterest-LP.html?_ga=2.20661553.86037572.1539782043-591736536.1533567354

