Asset Framework: Beyond the Basics

AVEVA Group plc
High Cross, Madingley Road
Cambridge CB3 0HB, UK
Tel +44 (0)1223 556655
Fax +44 (0)1223 556666

aveva.com

1. Table of Contents

1. Introduction

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Course Introduction

In this course, we will continue our journey in setting up our AF database. We will explore additional functionality in working with AF. At this point, before continuing in this course, it is expected you know

- how to navigate between databases and create a new one,
- create an element with and without a template,
- how to create an attribute from a variety of data references,
- how to create a table and obtain data for an attribute,
- how to create an enumeration set,
- how to organize attributes with categories,
- how to create a referenced hierarchy,
- the importance of Unit of Measures,
- a brief understanding of substitution parameters and their usage,
- and how to check-in changes to commit them to the database.

2. Working with Attributes

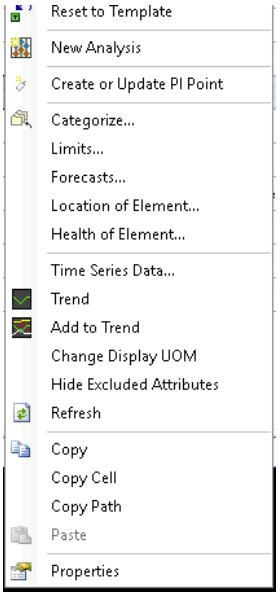
2.1. View Time Series Data – Trend Preview of Attributes

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: View Time Series Data – Trend Preview of Attributes

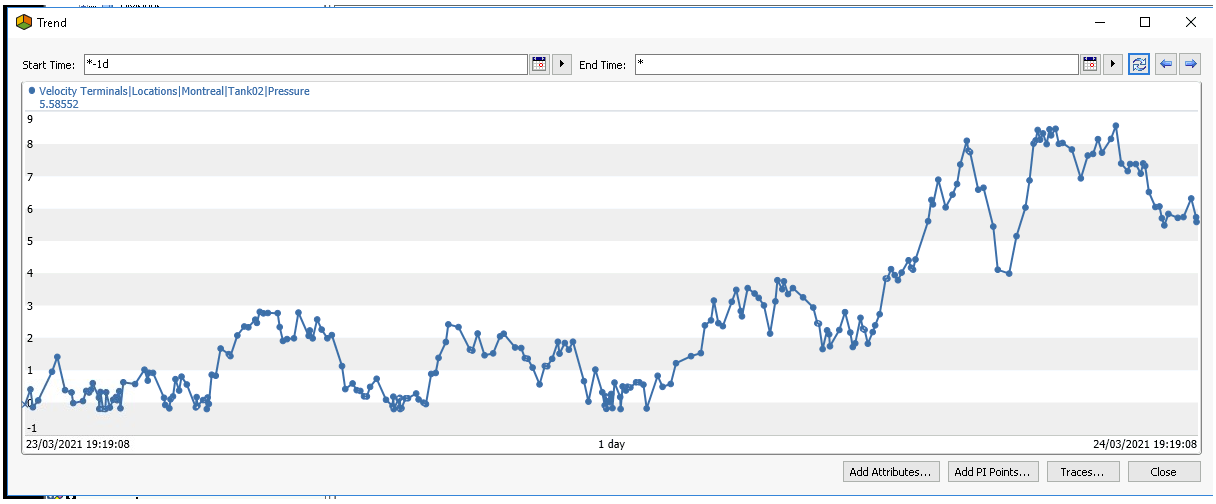
Database: **Velocity Terminals**

In this short video, we are going to view attribute data from within the template.

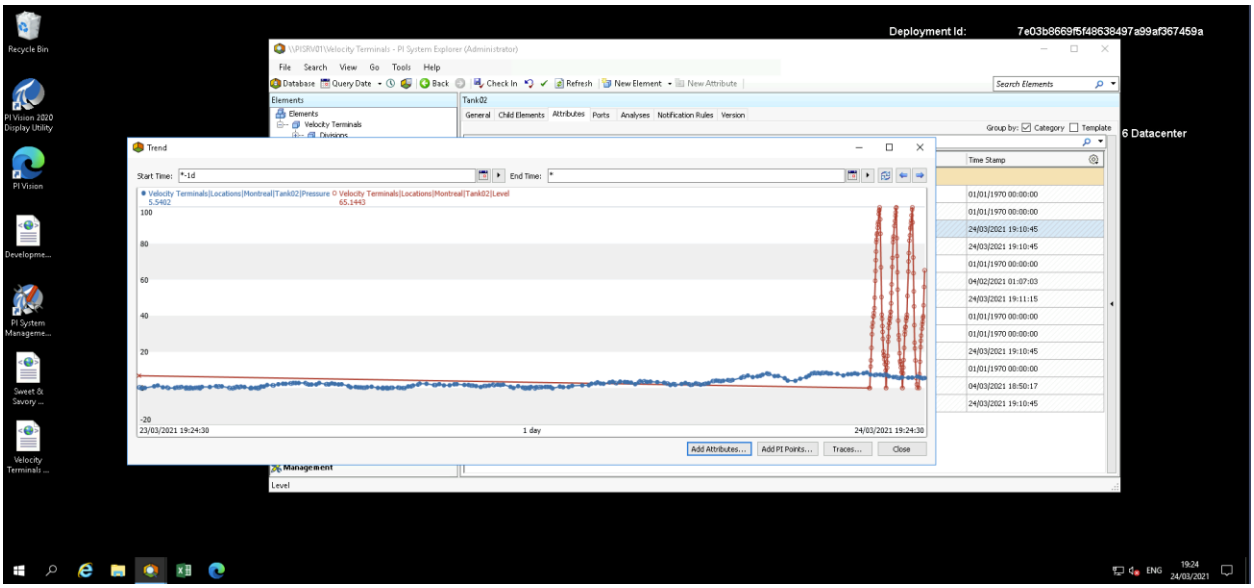
1. Be sure your database is set to Velocity Terminals.
2. Navigate to Elements in the Navigator Panel. We are going to view data from Tank02.
3. Select Elements > Velocity Terminals > Locations > Montreal > Tank02 > Attributes tab.
4. Select one of the attributes with a  (tag icon) to the left of the attribute name and does not have a value of Pt Created.
5. Right click on the attribute, in this case, we are using the Pressure attribute. Then select Trend to display the data.



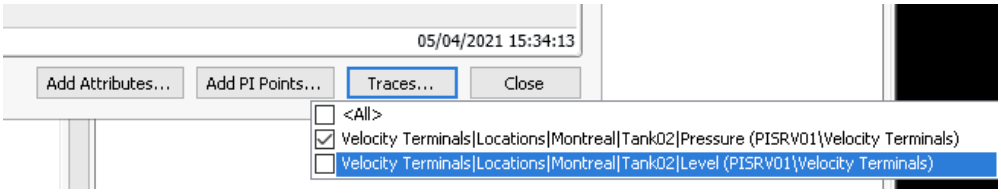
6. A trend will display. The timeframe of the trend can be modified to reflect the period of interest.



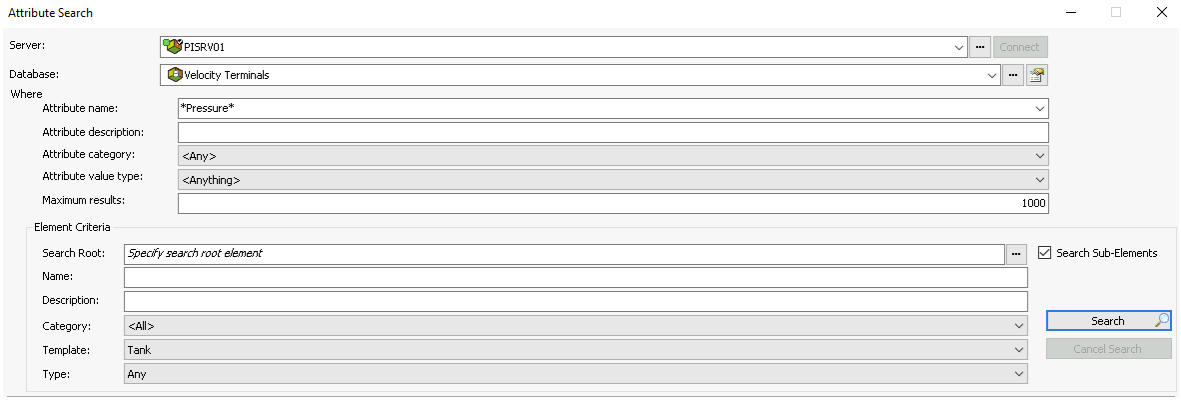
7. If a quick comparison needs to be made, additional attributes can be added to the display.
8. Additional attributes can be added through search functionality by selecting Add Attributes OR by navigating back to the viewer and select another attribute.
9. Let's start by selecting an attribute from the same element. In this case, we are selecting Level by right clicking on the attribute, then select Add to Trend.

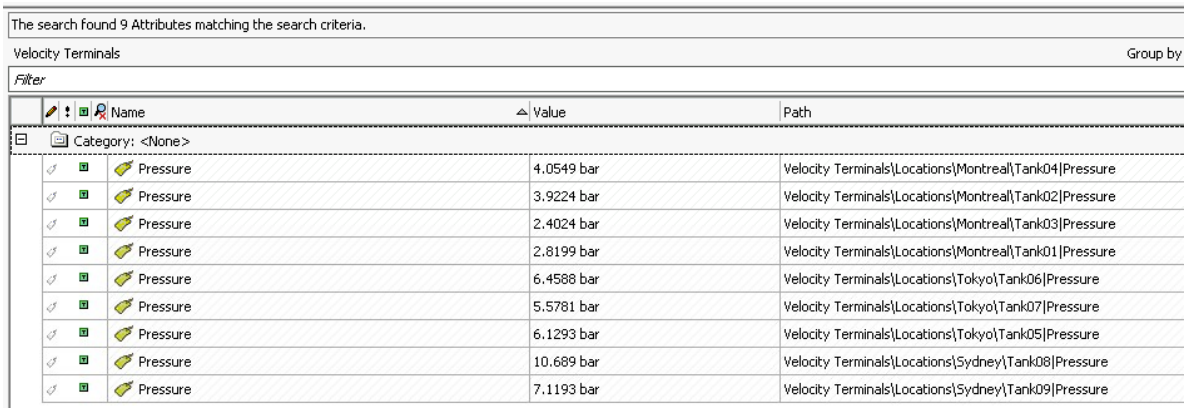


10. The Trace button at the base of the trend allows toggling attributes off and on.
11. Now, let's add an attribute from another element.
12. Start by removing the level attribute from the trend.

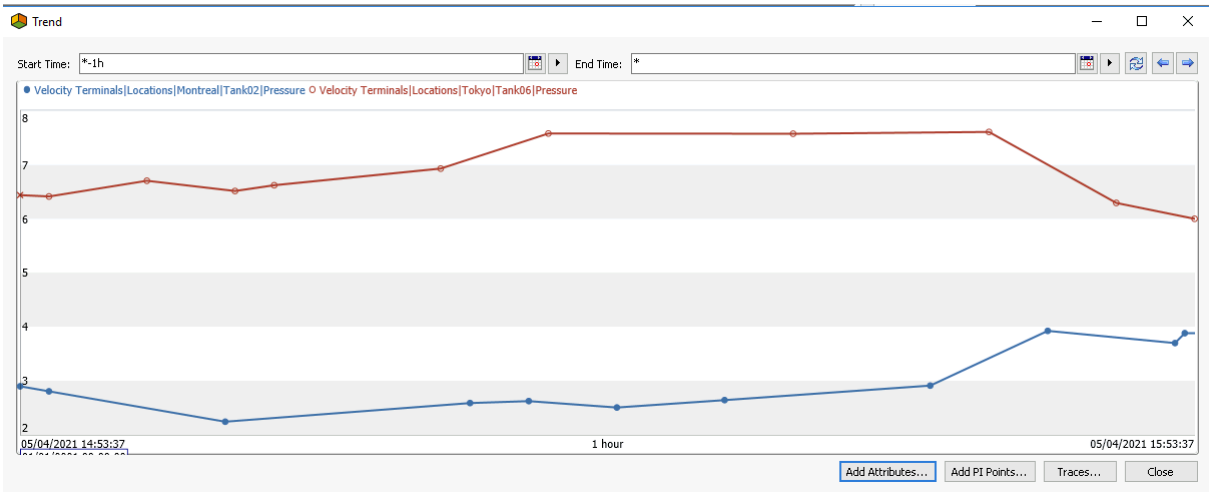


13. Select Add Attributes, an Attribute Search will display. Enter Pressure for Attribute Name, select Tank from the Template dropdown then check Search Sub-Elements. Select Search.





14. Select Tank06 (by selecting the line – verification by the line being highlighted) under the location of Tokyo followed by clicking OK
15. The two traces will appear on the trend to allow an easy comparison.



16. By clicking Close, the display will close.

2.2. Configuration of Retrieval Methods for Attribute Values

Client applications request attribute values for a specific time or for a time range. For example, in PI Vision, the display can optionally provide a time range context (a time range symbol, such as a trend, must be present on the display to enable reception of a time range). You typically configure the data reference to expect either a time or a time range. The attribute value will then be either:

- The value of the point at a specific time (functions similar to those associated with PI DataLink).

  The following table describes the available options for the **By Time** value-retrieval method in the PI Point Data Reference window.

Option	**Description**
After	Returns the first recorded value after the time requested by the client application.
At or After	Returns a recorded value at the time requested by the client application. If no value exists at the specified time, returns the next recorded value.
At or Before	Returns a recorded value at the time requested by the client application. If no value exists at the specified time, returns the previous recorded value.
Automatic	A continuous point (step attribute = 0) is treated as Interpolated, whereas a discrete point (step attribute = 1) is treated as At or Before.
Before	Returns the first recorded value before the time requested by the client application.
Exact Time	Returns a recorded value at the time requested by the client application. If no recorded value exists at that time, an error is returned.
Interpolated	Returns an interpolated value for the time requested by the client application. Discrete points (step attribute = 1) carry the previous value forward.

Not Supported	Used in time range calculations only. If the client application sends a time instead of a time range, PI AF returns an error message as the attribute value.
Time Range	Used in time range calculations only. Creates a default time range to use if the client application sends a time instead of a time range. If you choose this option, you must type a PI relative time expression in the Relative Time field. See Create default time ranges for element attributes for details.
Time Range Override	Used in time range calculations only. Specifies a time range that always overrides the time range supplied by the client application.

- The result of a calculation on the point's values over a time range. For example, the attribute value could be the average of the point values over an hour.

  The table below describes the available options for the **By Time Range** value-retrieval method in the PI Point Data Reference window.

Option	**Description**
Average	Returns the average value over the time range.
Count	Returns the event count over the time range, when Calculation Basis is set to Event Weighted. Returns the sum of event time duration over the time range, when Calculation Basis is set to any of the time weighted options.
Delta	Returns the difference in value from the end of the time range to the start of the time range.
End Time	Returns the value at the end of the time range.
Maximum	Returns the maximum value over the time range.

	Note: The timestamp value displays the time that the maximum value occurred.
Minimum	Returns the minimum value over the time range. Note: The timestamp value displays the time that the minimum value occurred.
Population Standard Deviation	Returns the population standard deviation over the time range.
Range	Returns the range of values over the time range (Maximum-Minimum)
Standard Deviation	Returns the standard deviation over the time range.
Start Time	Returns the value at the start time of the time range.
Total	Returns a totalization over the time range.

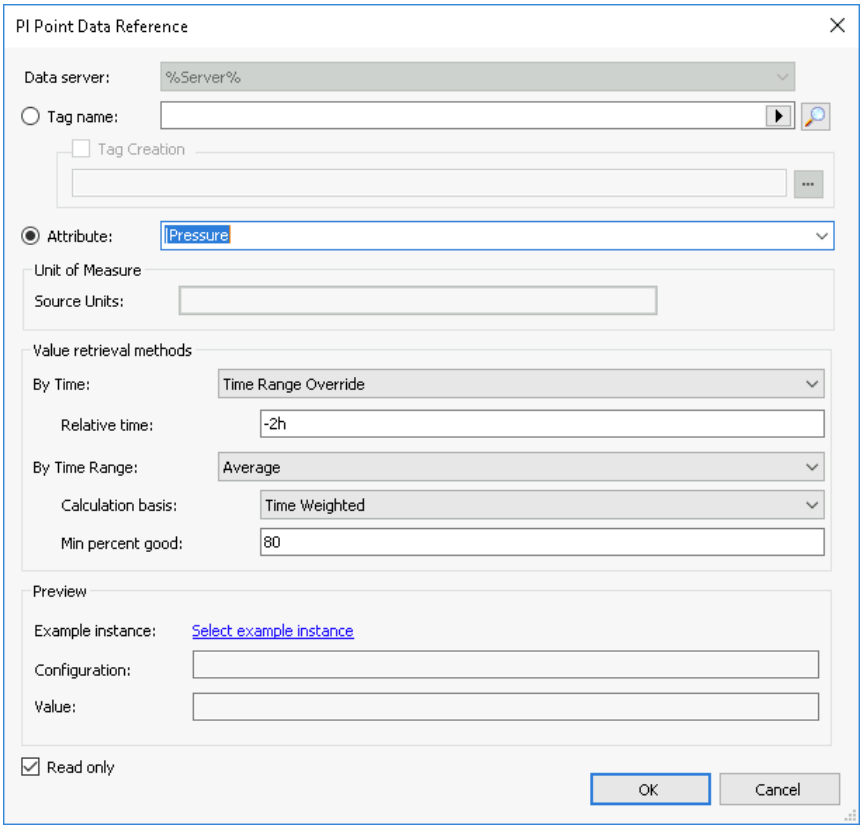
See also *Configuration of retrieval methods for attribute values* in the OSIsoft Documentation.

2.3. Create a 2HR Average using Retrieval Options

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Create a 2HR Average using Retrieval Options

Database: **Velocity Terminals**

1. We are still using the **Velocity Terminals** database.
2. Navigate to Library > Templates > Element Templates > Tank
3. Navigate to the Pressure Attribute.
4. In the previous lesson, Value retrieval modes were reviewed. At this time, we will work with some variations of retrieval modes.
5. Pressure is critical in our process, so we want to track the two-hour average pressure.
6. From the viewer screen, right click on Pressure and select New Child Attribute.
7. Rename to 2HRAvgPressure to represent the 2-hour average of the pressure attribute.
8. Open the Settings screen.
9. Select Pressure as the attribute.
10. This child attribute is based on the Pressure attribute. Note the substitution parameter usage when selecting Pressure (denoting the parent – child relationship: |Pressure).
11. Under Value retrieval methods> By Time: select Time Range Override, since we are wanting an average over a period of time.
12. In the Relative time box, enter -2h to indicate a period of two hours.
13. In the By Time Range drop down box, select Average.
14. In the Calculation basis: drop down box, event and time weighted options exist. In this instance, we are selecting Time Weighted.
15. The Min percent good box allows you to eliminate/filter out bad value associated with the Pressure attribute. By default, the field is set to 80%.

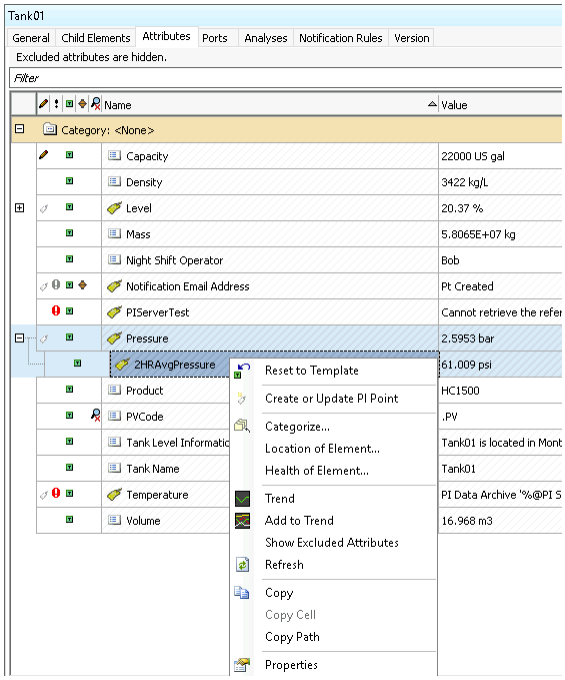


16. Prior to accepting (check in) the results, verify the 2-hour average pressure correlates with the pressure that we are experiencing.

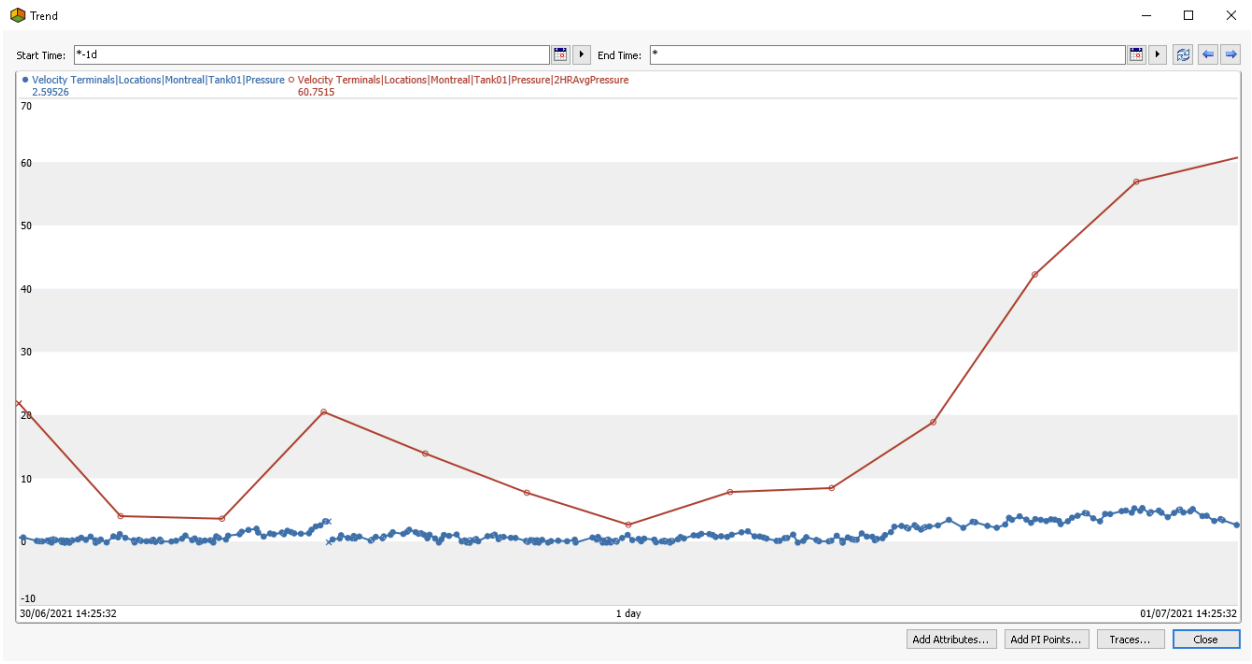
17. Navigate to one of the Tank assets.

18. Select Elements in the Navigator Pane → Velocity Terminals → Locations → Montreal → Tank01

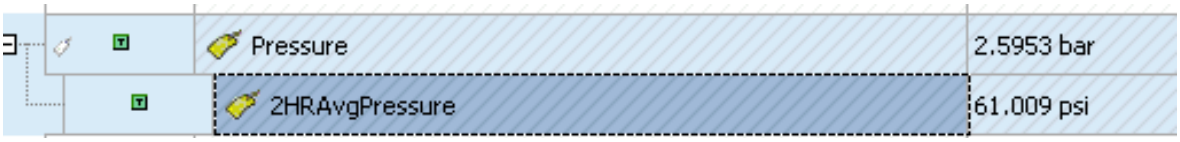
19. Select Pressure and the child attribute 2HRAvgPressure, then select Trend.



20. Upon reviewing the trend, the 2HRAvgPressure should be in line with the Pressure value. In this trend, it appears the 2HRAvgPressure is much higher than anticipated. Close the trend and review the attributes.



21. It looks like we did not verify the UOMs for the two attributes were of the same unit. Navigate back to the library and update the 2HRAvgPressure attribute to bar.

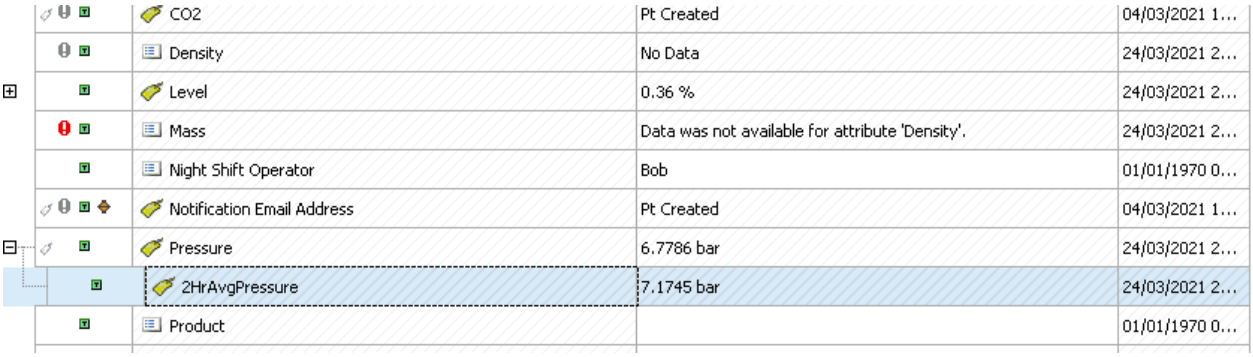


22. Check-In to save the changes to the database.

23. Refresh the screen.

24. Verify new child attribute is added to the attributes for the tanks.

25. Navigate to Elements > Velocity Terminal > Locations > Sydney > Tank08.



26. Trend the Pressure and 2HRAvgPressure to verify the two are as expected.

2.4. Syntax for Relative Paths

Before we look into the details of the syntax used for relative paths, let us briefly summarize the difference between relative and absolute path.

Especially for the configuration of attribute templates we need to consider where to use relative or absolute references. When do we need which kind of reference?

- Relative references are used when we are looking for something *relative* to the current element. Example: We need an attribute from the respective parent element of each element.
- Absolute references are used when the attribute should be coming from the *same* element for *each* element from a template. Example: We need an attribute from an element from another branch of the hierarchy. All elements from the template require the attribute from that same element.

With the Attribute Search we create absolute references, for example: \\System\Database\Element|Attribute. Those can also be changed manually to relative references if needed following the syntax rules shown below. The single period (.) can be used to replace parts of the path with the current object, for example modifying the absolute path above like this \\.\Database\Element|Attribute makes it server relative.

- Use a double period (..) to indicate the parent object. The following example references Attribute1 of SiblingElement that is in the current database:

```
..\SiblingElement|Attribute1
```

  For referencing an attribute of the parent element itself, the following syntax can be used:

```
..\|Attribute
```

- The single period (.) represents the current object. You can use it to create a relative path from the current object. For example to reference an attribute of a child element:

```
.\SubElement|Attribute
```

- When the current object is a PI AF attribute, a single period followed by a backslash (.\) represents the owning PI AF element. For example:

```
.\|Attribute1
```

- A single period followed by a vertical bar (.|) references a child PI AF attribute, for example:

```
.|Attribute1|Attribute2
```

- When the current object is a PI AF element, a relative path is created from the database, for example:

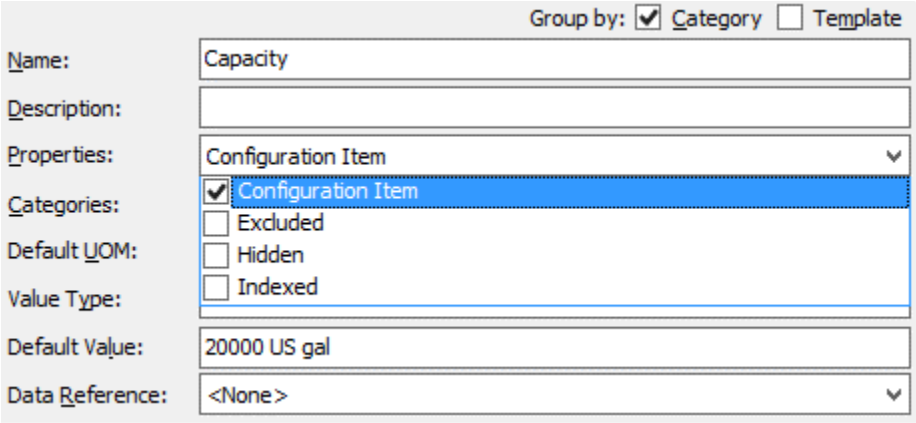
```
\Element1\Element2|Attribute1
```

See also *Indirect PI point references* in the OSIsoft Documentation.

2.5. AF Attribute Properties

When attributes are created, there are four associated properties (configuration item, excluded, hidden, indexed). You have the option to set those properties for each element attribute.

In case of an element derived from a template, the properties of the attributes cannot be changed any more except for the Exclude property:



Configuration Item	You assign the **Configuration Item** property to an attribute with a constant value that represents inherent properties of an asset (e.g. a device serial number). In PI System Explorer, configuration attributes are marked with a pencil icon (✎). When you change the attribute value of a configuration item PI System Explorer automatically checks out the attribute. To commit the change, you need to Check In.
Indexed	**Indexed** attributes are attributes that are optimized for fast search results and fast value retrieval. You can only index attributes whose values are stored in the AF database. This means that you cannot index attributes that get their values from PI point data references or from linked table references.
Excluded	In situations where not all attributes in an element template apply, attributes that are not applicable can be excluded. **Example**: only some of the tanks have a second container coating. Set the attribute for the material of the second coating to **Excluded** property for the tanks with a single coating.
Hidden	The **Hidden** property is useful if an attribute is being used to hold an intermediate result, such as a table lookup result that can then be retrieved

	by a PI point data reference, or is being used solely to populate a tag name in a substitution parameter.

2.6. Use of Attribute Properties

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Use of Attribute Properties

Database: **Velocity Terminals**

This video explores the use of attribute properties.

We will continue using the Velocity Terminals database. Right now, there are several attributes for the Tank template. In this video, we are going to assign attribute properties for specific use cases.

The first attribute property is **Configuration Item**:

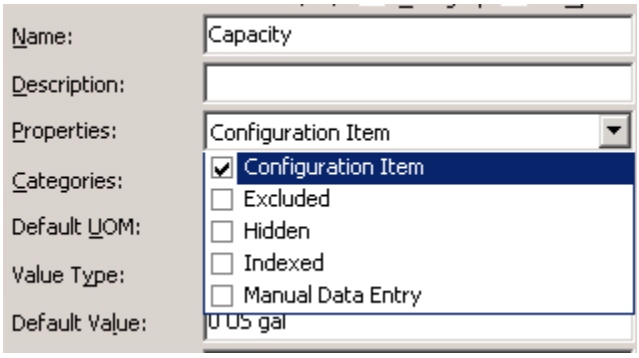
The Capacity attribute for a tank is considered as a constant that represent an inherent property for the tanks at Velocity Terminals. The value will always be the same except for a change of the equipment itself. Capacity changes should require a Check-Out and Check-In sequence in AF.

To implement this:

Open the **Velocity Terminals** database in PSE and navigate to the Tank template in the Library.

The Capacity attribute is a constant and falls under the Configuration Item property.

Select the Capacity attribute, then from the configuration pane, select Configuration Item under Properties.



Check-in the change.

Now, verify when a change is made to the Capacity attribute, the element is denoted with a change and requires a check-in.

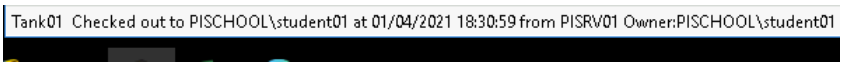
Navigate to Elements in the Navigator Pane, then traverse the hierarchy to Tank01 (located under Locations > Montreal).

Switch to elements and select *Tank01*. Change the *Capacity* from 20,000 to 22,000.

Notice, that the Tank01 element has been checked out. It is marked accordingly in the Asset tree 

.

If you select the element PI System Explorer status line displays details of the checked out element:



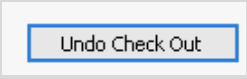
At this point, the change can be accepted (Check In) or rejected (undo Check out).

This was one of the requirements given to us from the Plant Management to prevent unnecessary changes.

We will accept the changes, to accept the change made to the Capacity attribute a check in is required.

If the changes need to be rejected:

Click on Undo Checkout button  in the toolbar to revert to the initial setting.

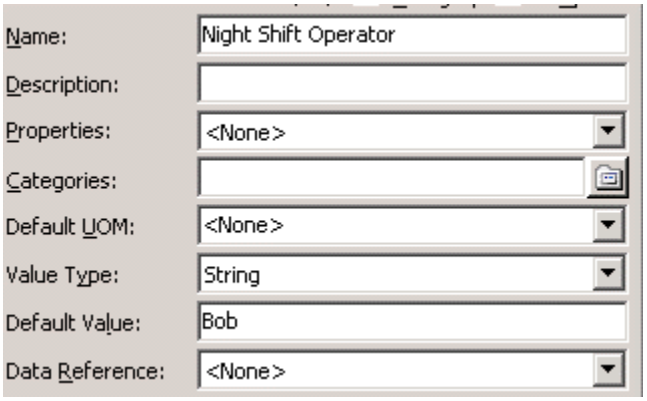
Click Undo Check Out 

The next property covered will be **Excluded**:

Only the first tank in every location of Velocity Terminals (Montreal, Sydney, Tokyo) has a dedicated night shift operator. Add attributes with the operator names, which should only exist for tanks Tank01, Tank05 and Tank08.

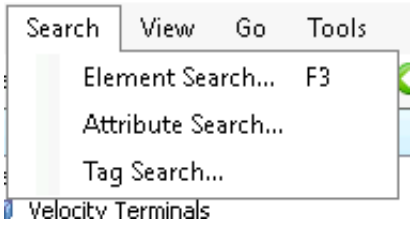
In the Tank template, add an attribute named Night Shift Operator and make it a string.

Give the attribute a Default Value of *Bob*. This will add the Night Shift Operator to all tanks using the Tank template.

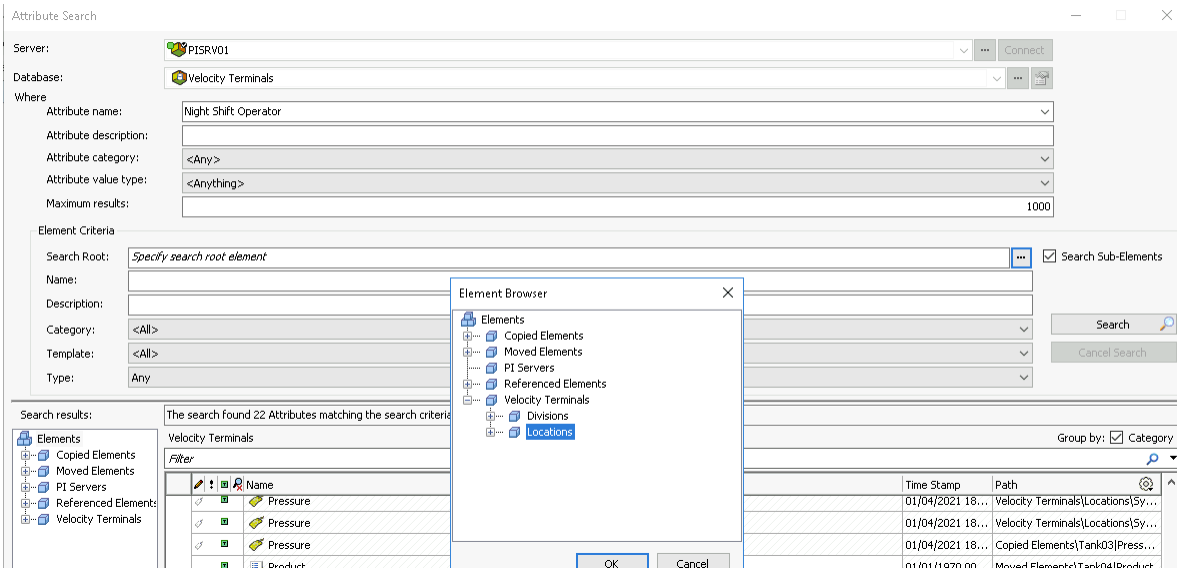


Check-In changes.

Navigate to Elements in the Navigator Pane, then perform an attribute search for the *Night Shift Operator* attributes under Locations.



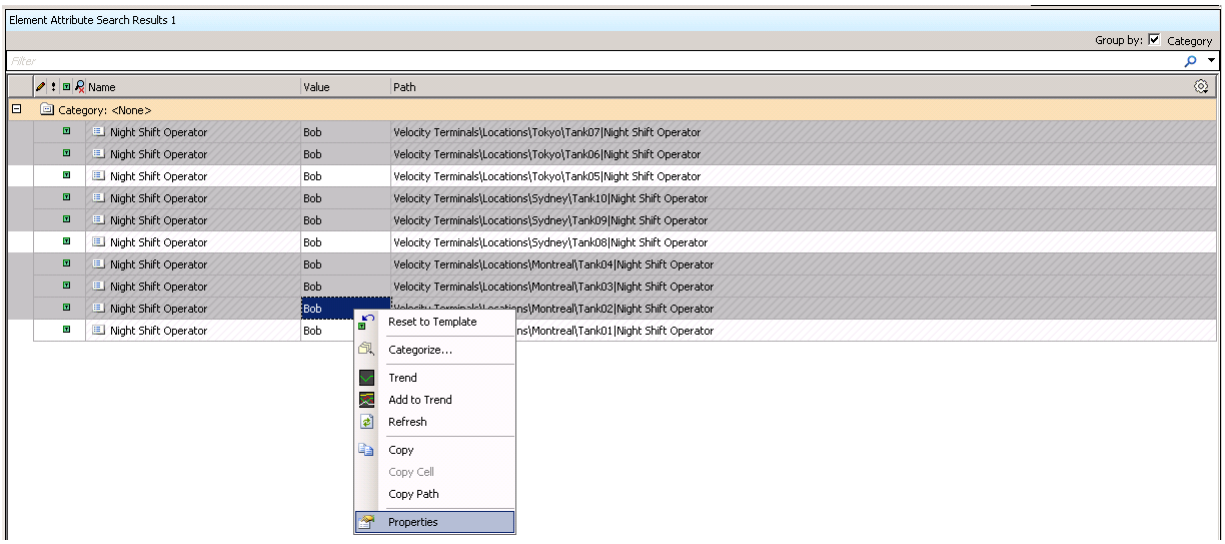
Enter Night Shift Operator to the attribute name, then select Locations from within the attribute search, select the ellipse (…) to the right of the Search Root.



Press Search, then OK.

Select all the tanks to return them from the search.

Only 3 of the tanks have a Night Shift Operator.

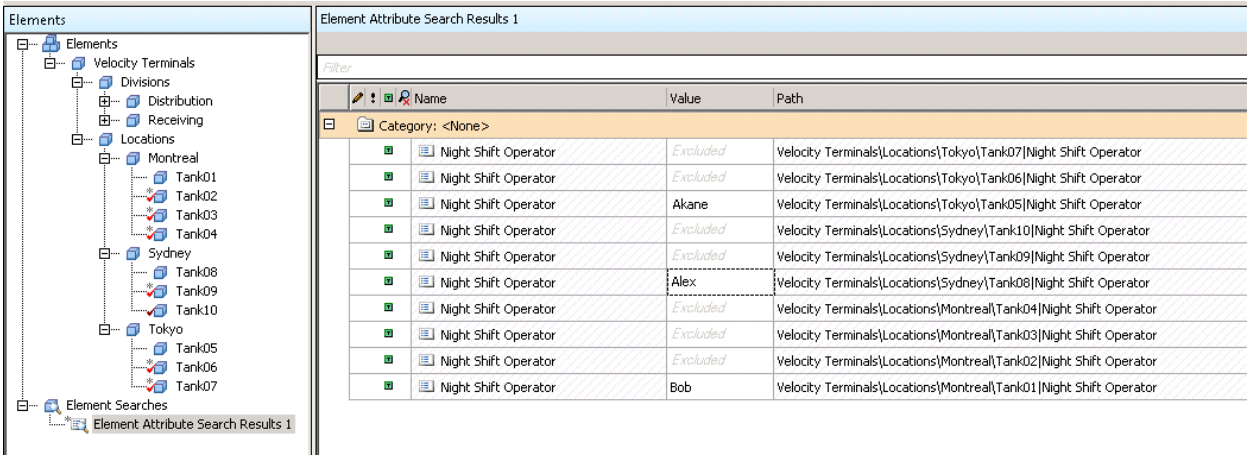


Start working through the tanks individually (or multiple rows by selecting them with the control or shift key) and update the Night Shift Operator for the tanks without a night operator (Tank02, Tank03, Tank04, Tank06, Tank07, Tank09, Tank10).

Select Properties and set **Excluded**. Click on OK.

The change applies to all the selected attributes.

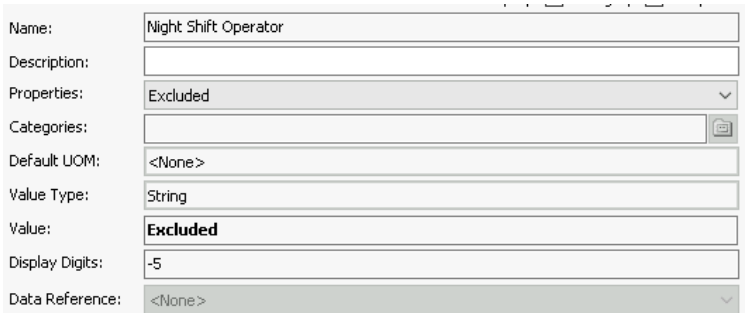
Now, change the names for the first tanks in Tokyo (Akane) and Sydney (Alex):



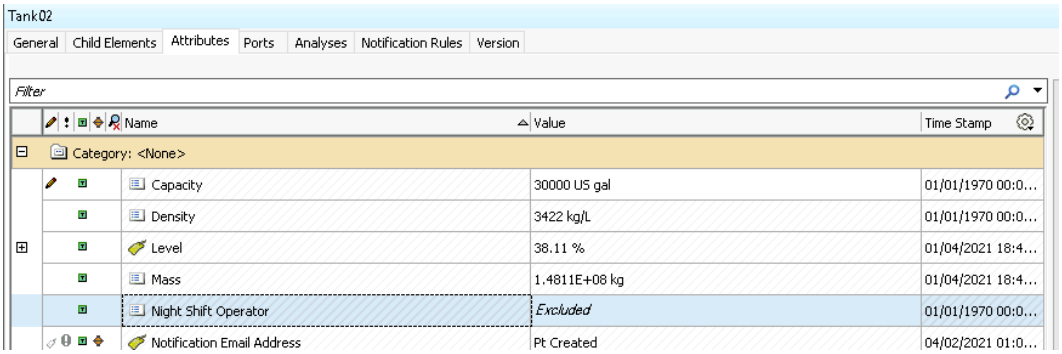
Click on Check-In. Then click on Refresh. Now the search returns only the non-excluded attributes for Tank01, Tank05 and Tank08.

Select the tank elements individually and observe what is displayed for the *Night Shift Operator.*

For all but Tank01, Tank05 and Tank08, Excluded should be in the Property field:

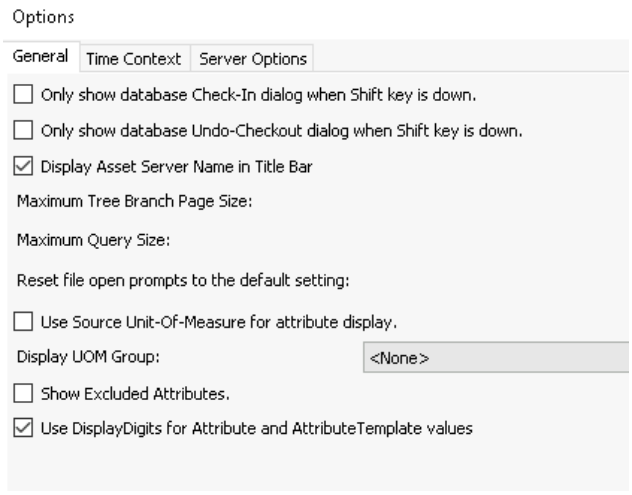


Current attributes for Tank02 lists Night Shift Operator:



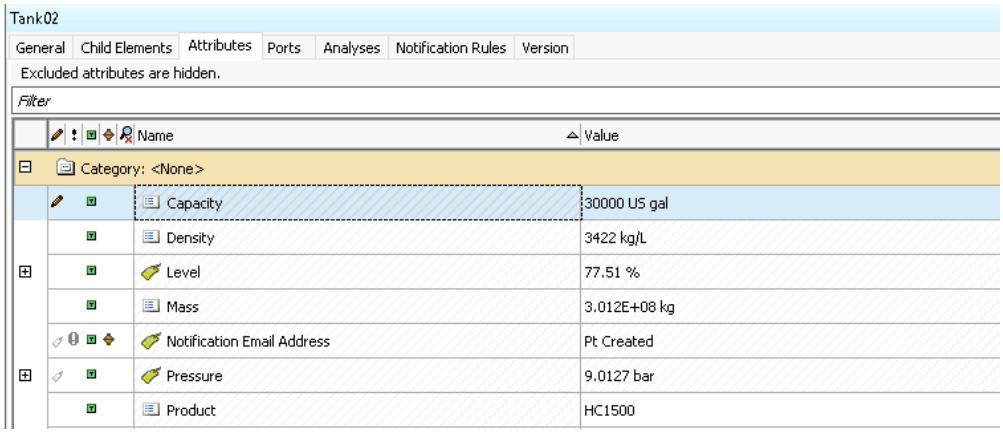
To eliminate it from the individual tanks where excluded, it is necessary to update the display properties of the elements.

Under Tools > Options select the *General* tab and uncheck the Show Excluded Attributes option.



Click OK. Select the tanks individually again and observe what is displayed now for the *Night Shift Operator.*

Tank02 after updating the display option:

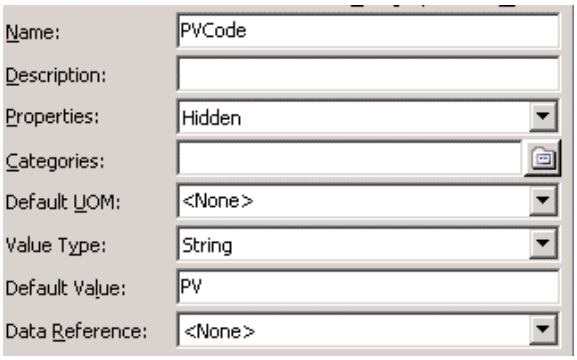


Hidden:

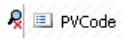
The tag name convention for process values at Velocity Terminals is the abbreviation ".PV" at the end of the tag name. The abbreviation should be defined in an AF attribute, but the attribute should not appear in PI Visualization Tools or in PI System Explorer Searches.

The PVCode is set-up to help during the creation of PI Point creation or look-up.

As this attribute is an intermediate it is unnecessary to any end users, let's mark it as a Hidden attribute.



Check-In the new attribute.

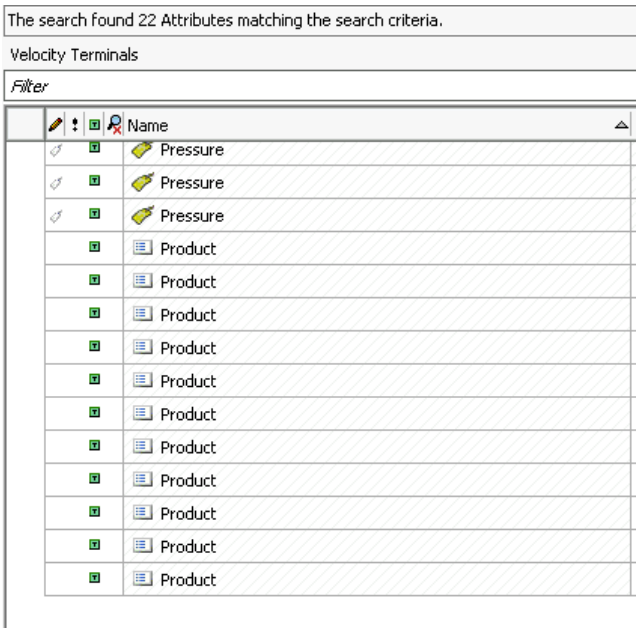
Notice that the *PVCode* attribute is marked as hidden 

.

Navigate to Elements in the Navigator Pane.

Perform an attribute search (search > Attribute Search) for all attributes using the Tank template that start with P*. PVCode is not returned.


Attribute Search
Server:
PISRV01
Connect
Database:
Velocity Terminals
Where
Attribute name:
P*
Attribute description:
Attribute category:
<Any>
Attribute value type:
<Anything>
Maximum results:
1000
Element Criteria
Search Root:
Specify search root element
Search Sub-Elements
Name:
Description:
Category:
<All>
Search
Template:
Tank
Cancel Search
Type:
Any


The search found 22 Attributes matching the search criteria.
Velocity Terminals
Filter
Name
Pressure
Pressure
Pressure
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product

2.7. Column Visibility in Viewer Pane

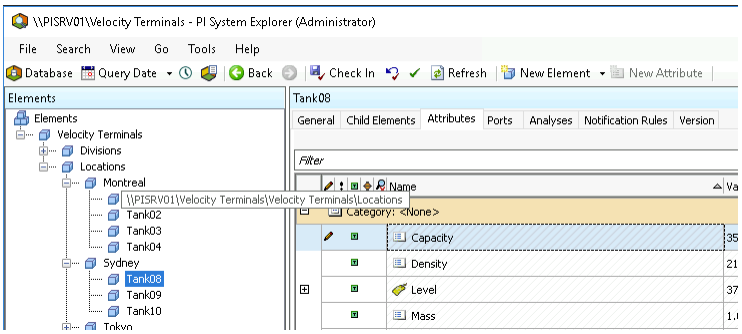
Database: **Velocity Terminals**

The fields in the viewer pane can be modified. Continue in Velocity Terminals database, navigate to Elements → Velocity Terminals → Locations → Sydney → Tank08. The column visibility can be modified to view the timestamp for the attributes.

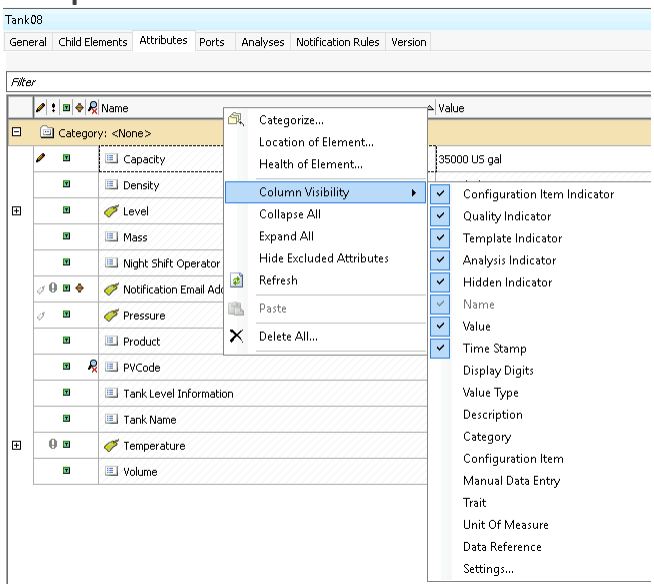
Step by step:

1. Click on a Tank08 under Sydney in the **Browser** pane

2. Click on the **Attribute** tab in the **Viewing** pane



3. Right click on the header in the **Viewing** pane and select **Column visibility>Time Stamp**



4. Now you have the **Time Stamp** added to the **Viewing** pane. This one is especially useful since the PSE does not show the current time for performance purposes.

To update the values for the attributes you need to do a **Refresh**. With the **Time Stamp** being visible you have a time context for the values.

2.8. Bulk Editing in PSE

Database: **Tank Farm**

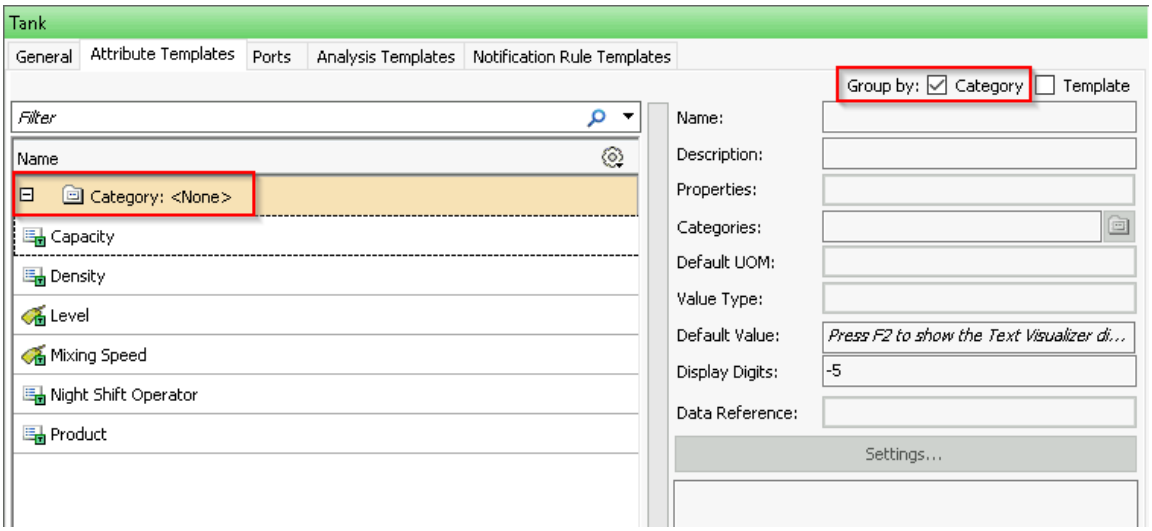
Categories and Multi-select Editing

When working in PSE, it can be tedious to edit objects one-by-one; however, there is are techniques to edit multiple objects at once. We already performed a multi-select edit when we assigned categories to the templates.

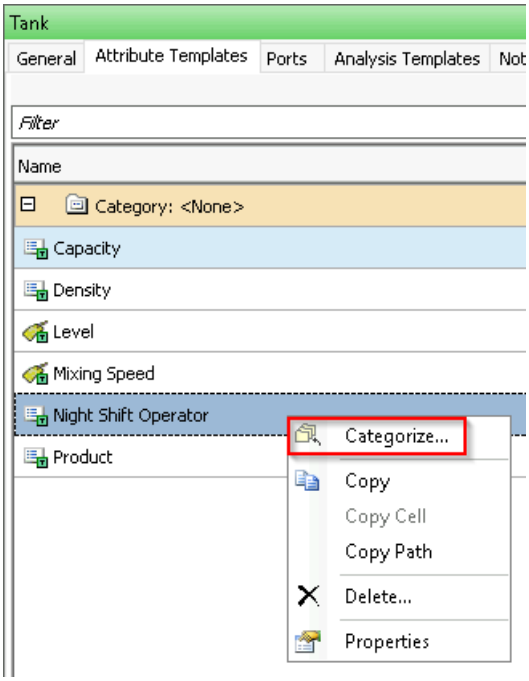
Approach: In the **Tank** template, assign categories to multiple attributes at the same time.

1. If you are not in the **Library** view press the **Ctrl+3** key combination
2. Then select the **Tank** Template under **Element Templates**. Click on the **Attribute Templates** tab.
3. Select the **Group by: Category** check box. So far we don't have any attribute categories and all attributes are grouped under <None>.

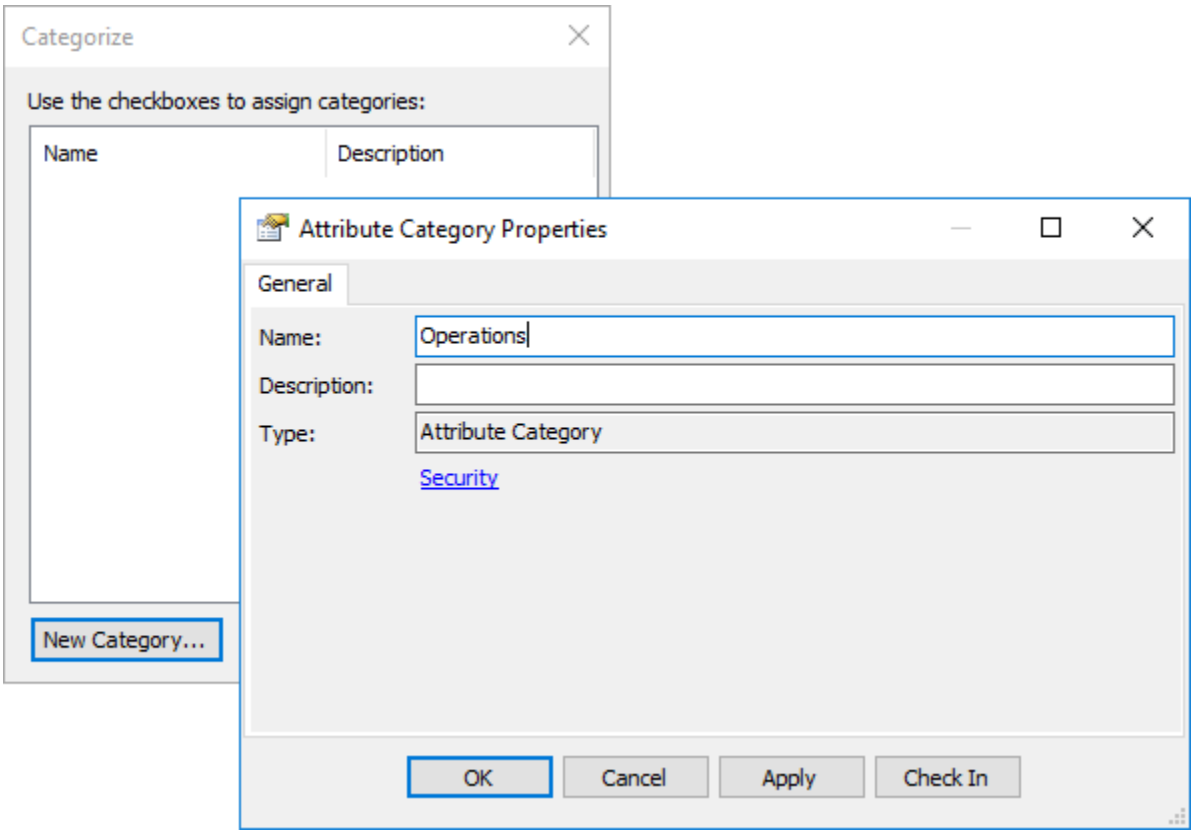
(**Note:** an attribute can belong to multiple Categories.)



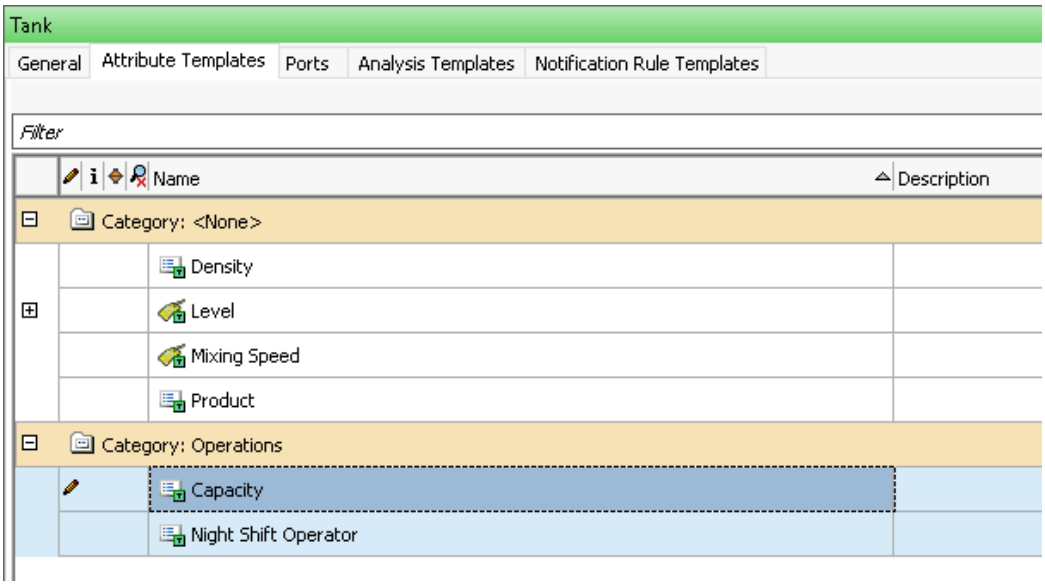
4. First, we want to categorize the attributes associated with Operations. Select the **Capacity** Attribute, hold down the **Ctrl** key and select the **Night Shift Operator** Attribute so both Attributes are selected. Right click and select Categorize…



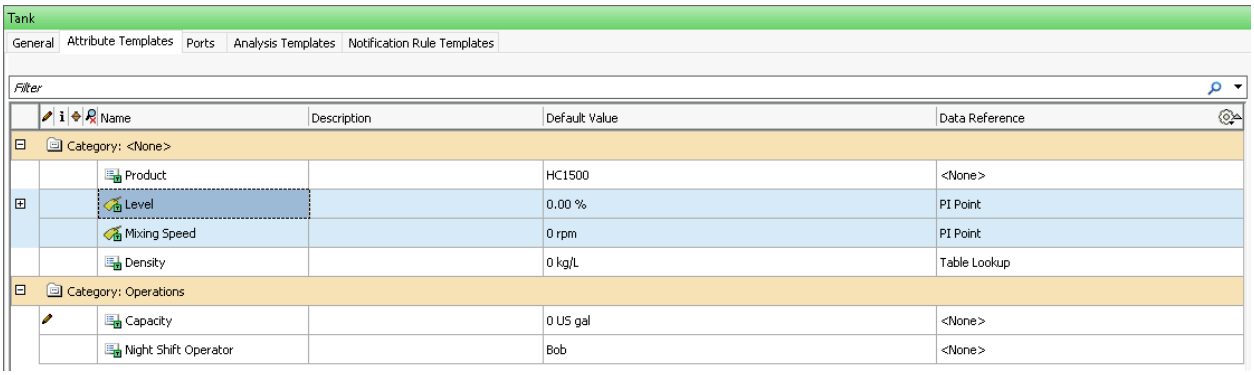
5. In the Categorize window click New Category… and type the name Operations in the Attribute Category Properties window. Click Check In and OK.



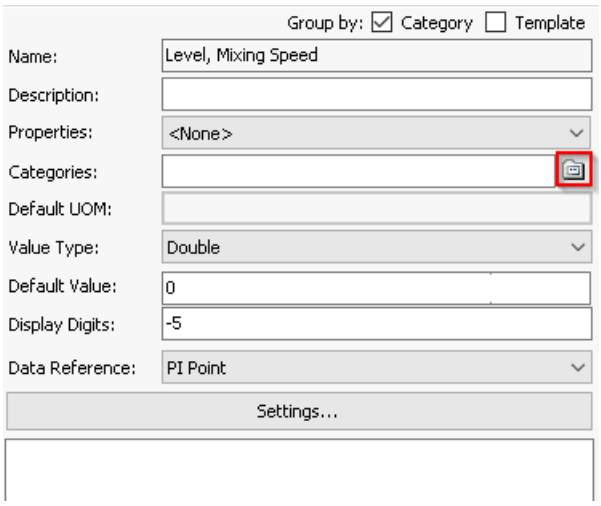
6. We can see the two attributes are now grouped under the category Operations.



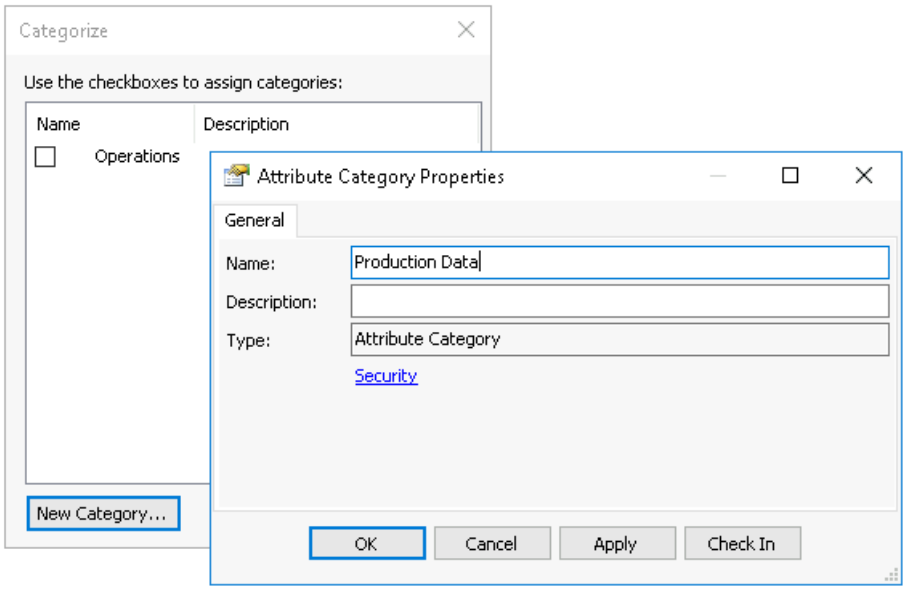
7. Of the remaining attributes we would like to group those with PI Point Data Reference into a category **Production Data**. To make an easier multi-selection, we sort the attributes by Data Reference by clicking on the header of the Data Reference column (use the gear icon to display the Data Reference column if it is not shown). Now we can select all PI Point DR attributes by clicking on the first one on the list (Level), holding down the Shift key and clicking on the last one on the list (Mixing Speed). This is particularly useful when there is a longer list of attributes to categorize.



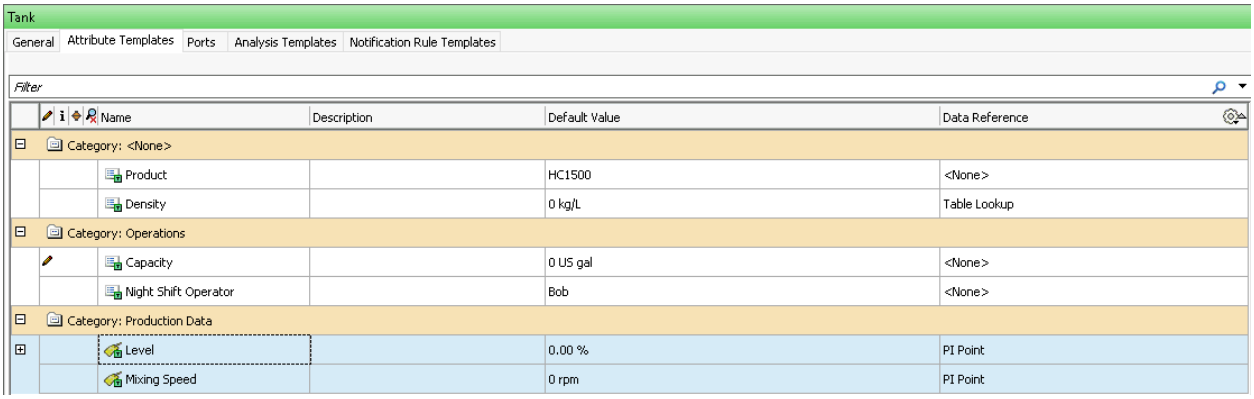
8. Let us now apply a slightly different method. Select the attributes **Level** and **Mixing Speed**. This time we want to use the Configuration Panel to apply the configuration. Notice that the name field shows both selected attributes. Only those fields which are configured identically are active (Caution: the Settings… button is actually active. Clicking it resets the existing configuration.) and can be configured for all the selected attributes at once. We can click the folder symbol to the right of the Categories field to choose a category for both attributes.



9. The Categorize window opens up again. Now it shows the available category. We create another new category and name it Production Data.



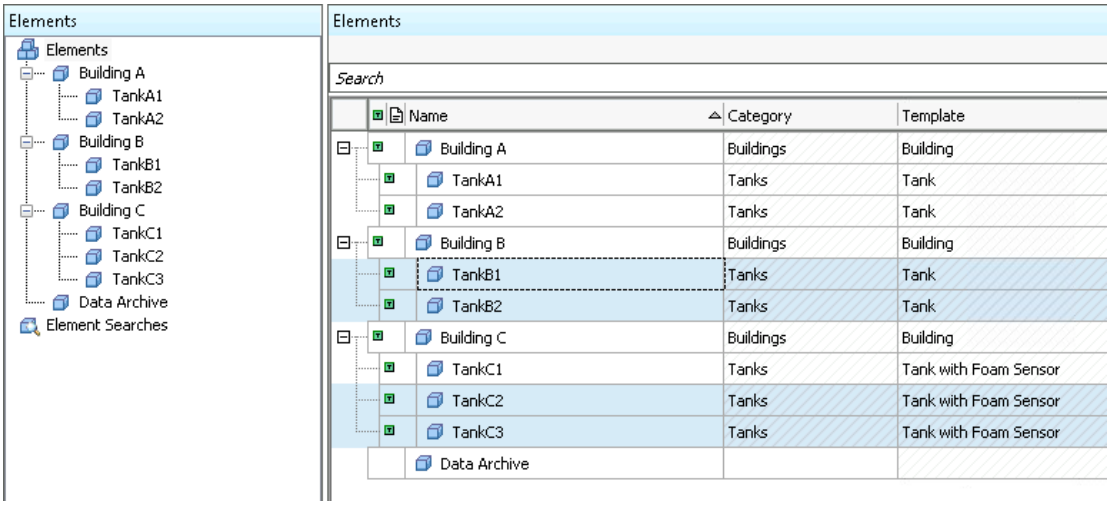
10. Click Check In and OK and again OK.



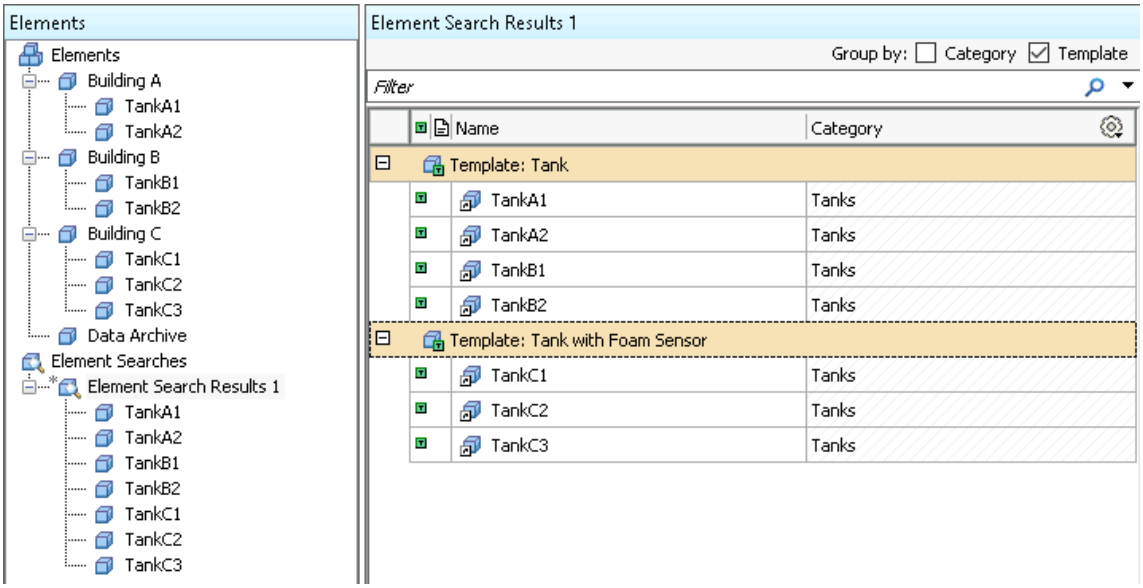
Some Tips for Bulk Editing of Elements

The Browser pane does not allow multi-selection, so we need to display the objects we want to select in the Viewer pane where this is possible. Here are some examples how this can be achieved:

- The whole element tree can be displayed in the Viewer by selecting Elements. This can be useful for selecting elements which are not child elements of the same parent at the same time.



- An Element Search can be performed to selectively list elements. For example we can search for all elements using the Tank template. In the screenshot below the search results have been additionally grouped by template.



3. Substitution Parameters Revisited

3.1. Advanced Substitution Parameters

Not only can substitution parameters resolve to the name of an AF object like an attribute, but also to the value of an attribute.

Imagine the Velocity Terminals example again. Imagine the tag names are always built the same way following this tag naming convention:

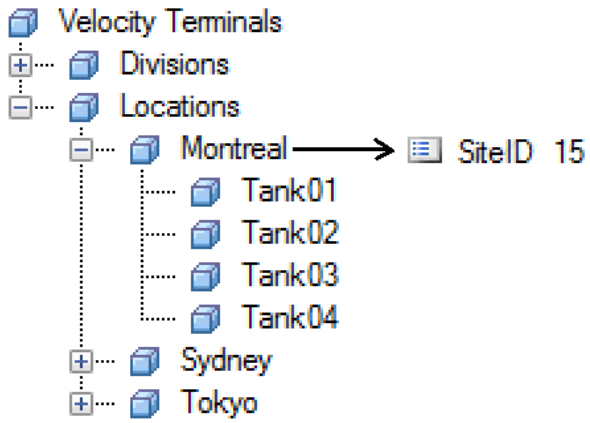
15Tank01**TS**.PV

LocationCode*EquipmentName****MeasurementType****.DataType*

Where

LocationCode	=	Code for the location (ex.: Montreal is 15)
Equipment Name	=	Equipment name (ex.: TANK01)
MeasurementType	=	Measurement type (ex.: TS for temperature sensor)
Data Type	=	Type of data (ex.: PV for process value)

This could lead us to determine that the Tank01's temperature would have its tag named ***15TANK01TS.PV***. With that in mind, substitution parameters make it possible to configure the temperature attribute at the template level in such a way that any newly created tank in the AF hierarchy would automatically see its temperature mapped to the correct Data Archive tag. The idea here is to have all the required information somewhere in the structure so the attribute can reconstruct the tag name depending on where it is in the hierarchy. A solution to the above tag naming convention would simply be to add an ID attribute under the city name element.



Then, the level attribute could be built using the following configuration string:

\\%Server%\%@..\|SiteID%%Element%TS.PV

Reading a Substitution Parameter

The above *%@..\|SiteID%* substitution parameter can be read as: "Give me the value of the parent-element's SiteID attribute." In order to better understand the configuration string above, refer to the following table:

Symbol	Definition	Examples
%[…]%	Consider the expression as a substitution parameter.	**%**Element**%**, **%**Attribute**%**
.	Navigate a level down.	%@.\<ChildElement>|<Attribute>%
..	Navigate a level up.	%**..\..**\Element%, %**..**|Attribute%
\	References an element.	%..****Element%
|	References an attribute.	%..**|**Attribute%, %@**|**<Attribute>%
@	References the value of the object instead of its name.	%@..\..|<Attribute>%

General syntax for Substitution Parameters

- Strings enclosed between % and % will substitute the name of the entity. So %Element% returns the name of the Element containing this Attribute, %Attribute% returns the name of the Attribute.
- | symbol indicates that the following string is an Attribute. If the Attribute is at the same level as the current Attribute, you do not need the | symbol.
- With the @ symbol the value of the Attribute is returned. For example, %@.|PITagName% means replace with the value of the child Attribute called PITagName.
- Using ..\ we can move upwards in the hierarchy. For example, %..\Element% means replace with the name of the parent Element of the current Element. Using this syntax repeatedly we can move up multiple levels, %..\..\Element% would return the grandparent (the parent of the parent) of the current Element.

For more information see also *List of PI AF Substitution Parameters* and *Substitution Parameters in Data References* in the OSIsoft Documentation.

3.2. Using Substitution Parameters in a Template Attribute

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Using Substitution Parameters in a Template Attribute

Database: **Velocity Terminals**

Exercise Objectives

- Assemble substitution parameters to automatically map Data Archive tags to AF attributes.

Problem Description

Velocity Terminals will buy other facilities in the next few months where multiple tanks are present. They are going to implement the same tag naming convention at those new locations and will want to save time when adding those extra assets to the actual AF hierarchy.

They want to use the substitution parameters to speed up the creation of additional assets in AF. To demonstrate that this is feasible, you are asked to add a new temperature attribute template to the Tank's template so that it uses substitution parameters to automatically find the correct PI tag.

To assist in simplifying tag creation, you plan to set-up a table as a look-up reference for sites.

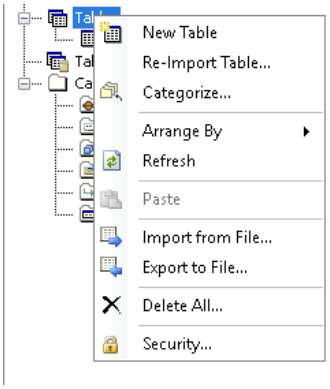
Site	SiteID
Montreal	15
Sydney	23
Tokyo	44

Approach

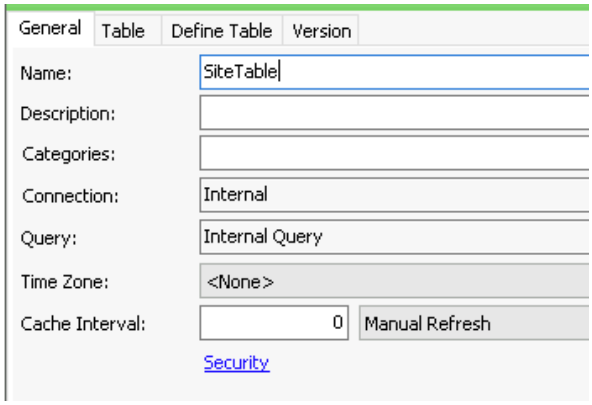
When we created the hierarchy, we hard-coded the Locations in the elements. Now, that we are expanding, it might be easier to create a table reference for the locations.

In the C:\OSIsoft folder, there is a spreadsheet named SiteTable. The spreadsheet will be imported into a new table called SiteTable.

Create a table named **SiteTable** in Library > Tables.

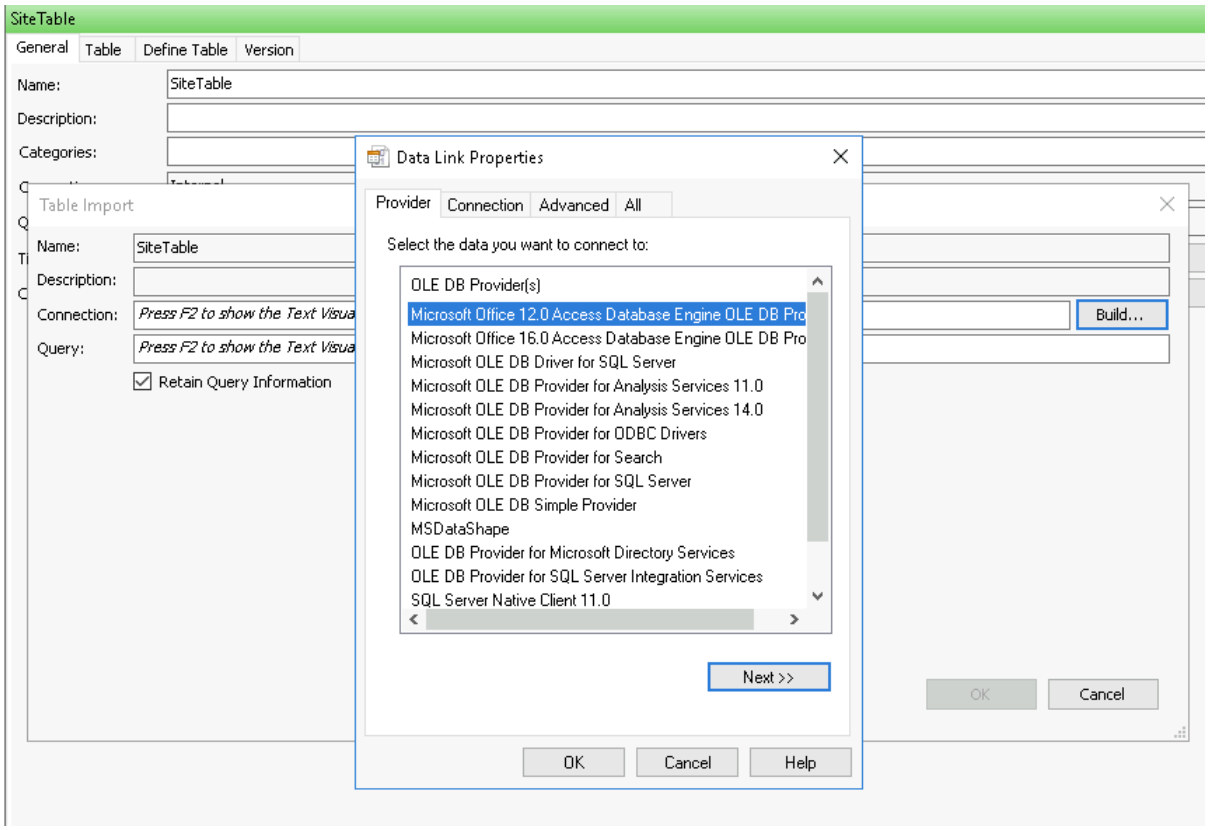


At this time, we will not define the table, we are going to rely on the spreadsheet to create and populate the table.



Select **Import** to build the connection to the spreadsheet.

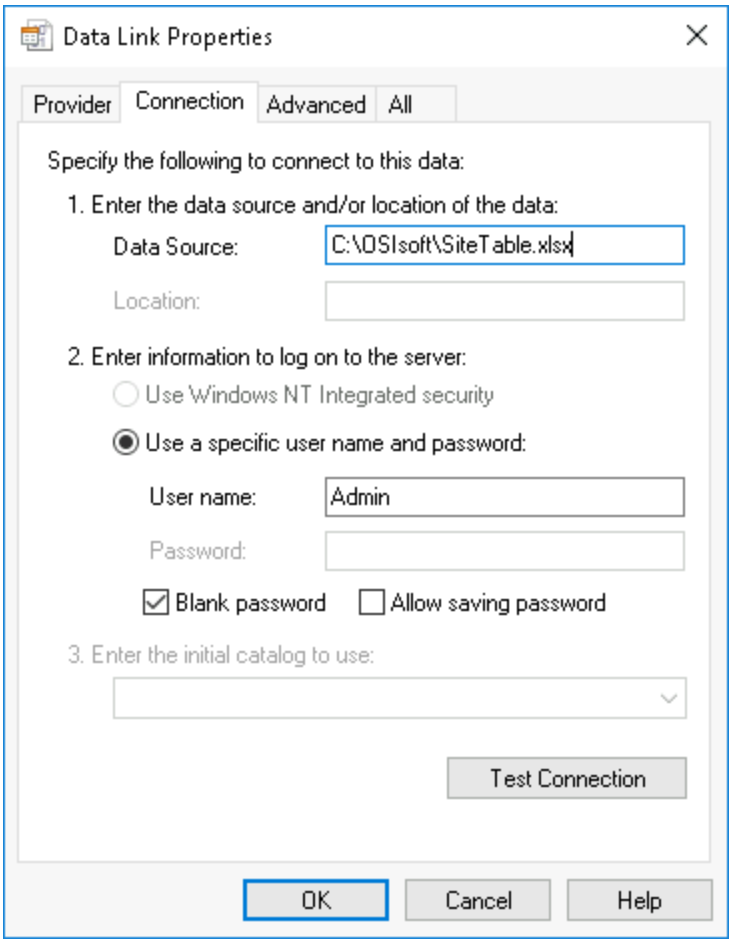
Select Build.



On the **Provider** tab, select the provider according to the version of Microsoft Office that you are using. In our case select **Microsoft Office 12.0 Access Database Engine OLE DB Provider** (appropriate for Office 2007 and higher). Click Next.

Note: For Office 97-2003: select Microsoft Jet 4.0 OLE DB Provider.

On the **Connection** tab, specify the following and click OK.



- Data Source

  The location and file name of the database or workbook (in our case **C:\OSIsoft\SiteTable.xlsx**). If you are linking, the path to the file must be relative to the PI AF server. Note that linking is not recommended for Excel spreadsheets.

- User Name

  Login credentials of a user that has been granted read access to the database or workbook. We are using the **Admin** account here.

Note: To store the password with the connection information, select the Allow Saving Password check box. The password is stored as plain text (not encrypted).

On the **Advanced** tab, in the Access permissions list, select **Share Deny None**.

On the **All** tab, select the **Extended Properties** value and click Edit Value.

The Edit Property Value window opens.

Enter the property value according to the version of Microsoft Excel that you are using, and then click OK. In our case (for Excel 2007 and higher) we enter **Excel 12.0**.

Note: For Excel 97-2003 use Excel 8.0.

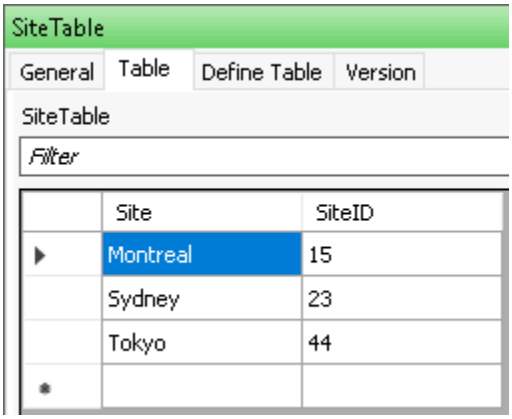
To verify that the spreadsheet is accessible, return to the Connection tab and click **Test Connection**.

If the settings are valid, a Test connection succeeded message displays.

Click OK to dismiss the window and return to PI System Explorer.

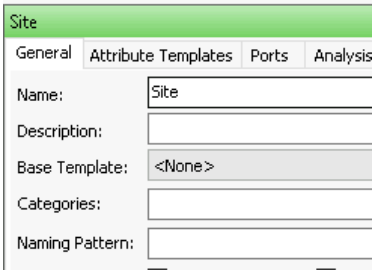
To define the data to be returned from the spreadsheet, enter an SQL query in the Query field. In this case we are using SELECT * FROM [Sheet1$]. To dismiss the window, click OK.

To review the resulting data, examine the Table tab. If the query is specified correctly, the tab contains a table displaying the results.

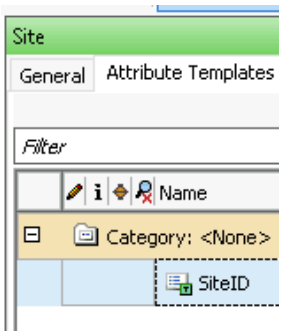


To save your changes, right-click the table node (or access from the top menu) and choose Check In.

Create a new template – called Site.



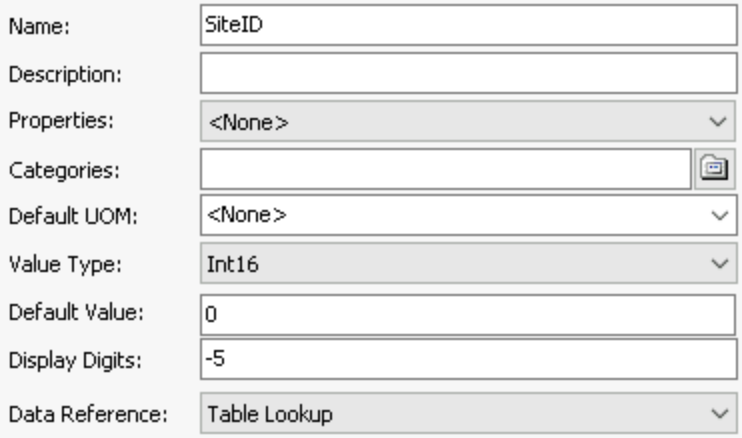
Create an attribute named SiteID in the Site template.



Earlier, we created a table named SiteTable that we will use to identify each Site. In order to populate the SiteID attribute.

Set value Type to Int16

Set the Data Reference to Table Lookup

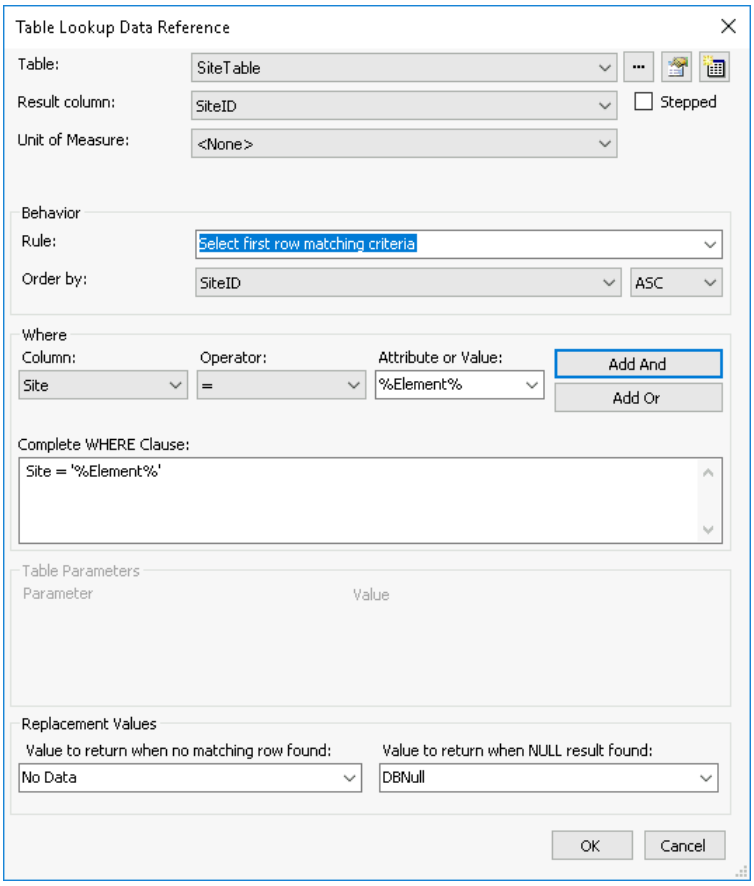


Open Settings to instruct the lookup how to identify the SiteID associated with the location (%Element%).

The statement passed to the database:

```
SELECT SiteID FROM SiteTable WHERE Site = '%Element%'
```

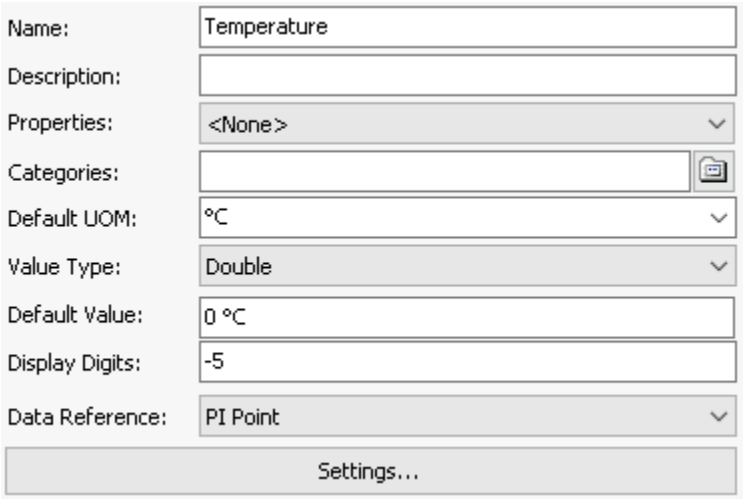
The set-up within the Table Lookup Data Reference screen:



Check in the Site template.

Now, the temperature attribute can be completed. Navigate back to the Tank template.

Navigate to the Attribute tab and add a new attribute, change the name to Temperature. Set the default UOM to °C and Value Type to Double.



Use substitution parameters so that the tag names are automatically built according to the expected tag naming convention.

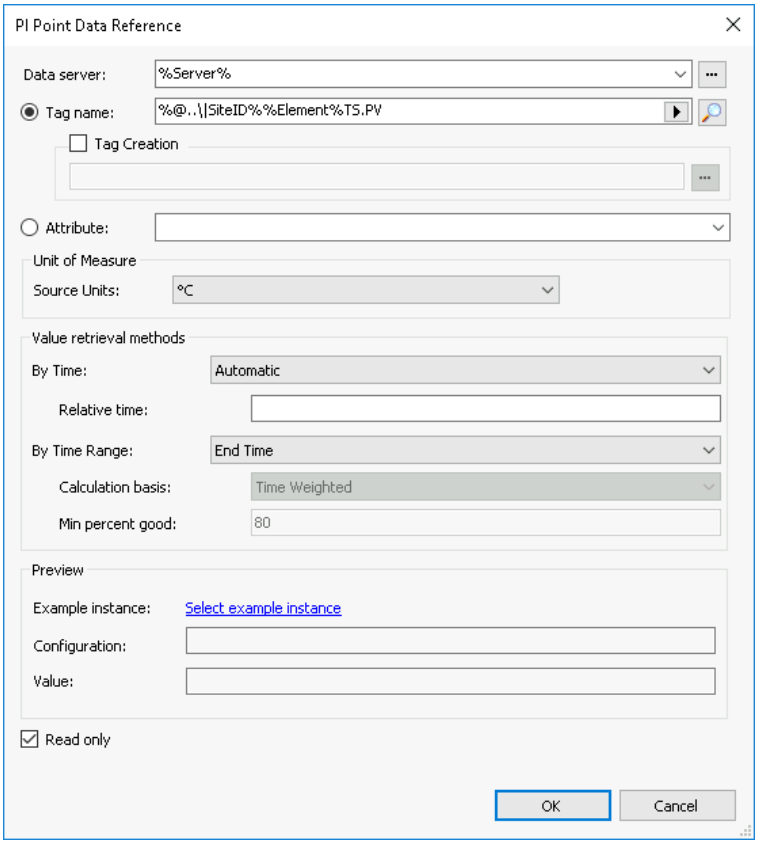
Two options exist to create the PI Point syntax.

Option 1: Directly from the PI Point Data Reference.

%@..\|SiteID%%Element%TS.PV

%@..\|SiteID%: This portion of the syntax above obtains the value from the Primary Parent template SiteID attribute.

%Element% represents the Element name itself.



Option 2: As a Child Attribute of Temperature to contain the string

String Builder Settings: '..\|SiteID';"%Element%";TS.PV;

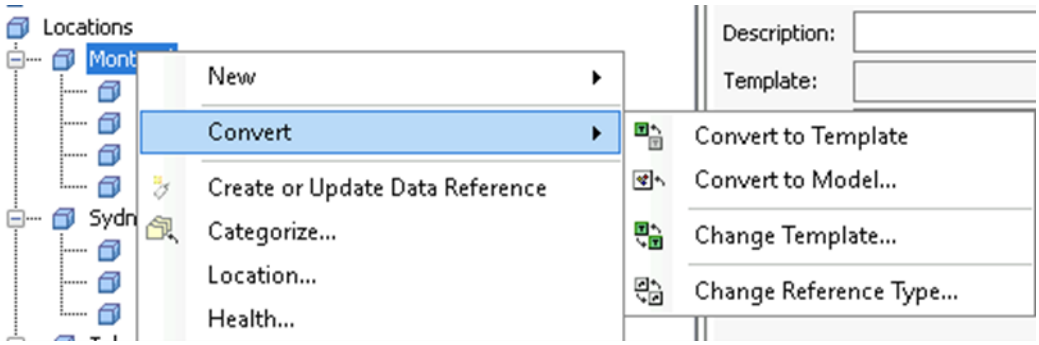
Note: There is a difference between the substitution parameters used in PI Point Data References and the syntax used by String Builder: %@..\|<attribute>% works for PI point references but not for the string builder reference, it is instead '..\|<attribute name>'.

For more information refer to "*Configuration of data references*" in OSIsoft Documentation.

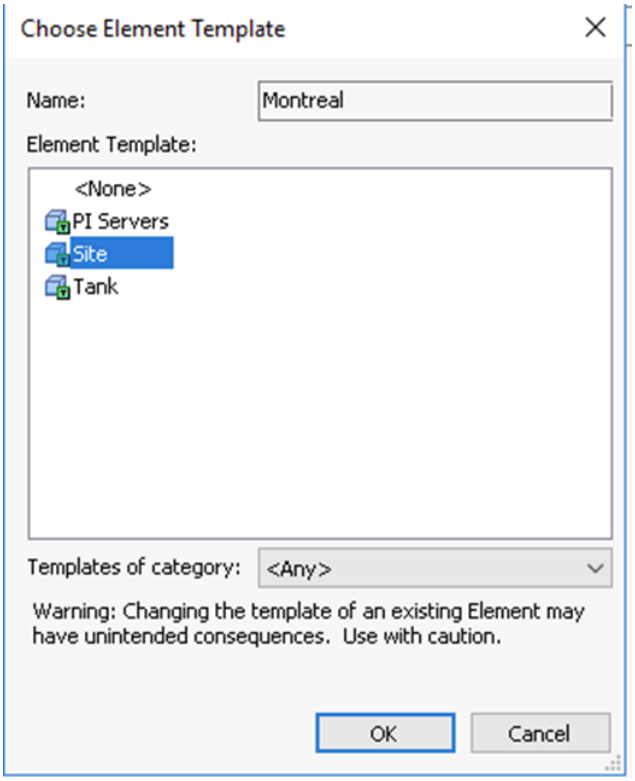
Check in the changes.

One more step is required to complete the validation of the syntax. The site locations need to be associated with the newly defined Site template. Navigate to Elements> Velocity Terminals > Locations > Montreal

Right click to display the content menu, then select Change Template.

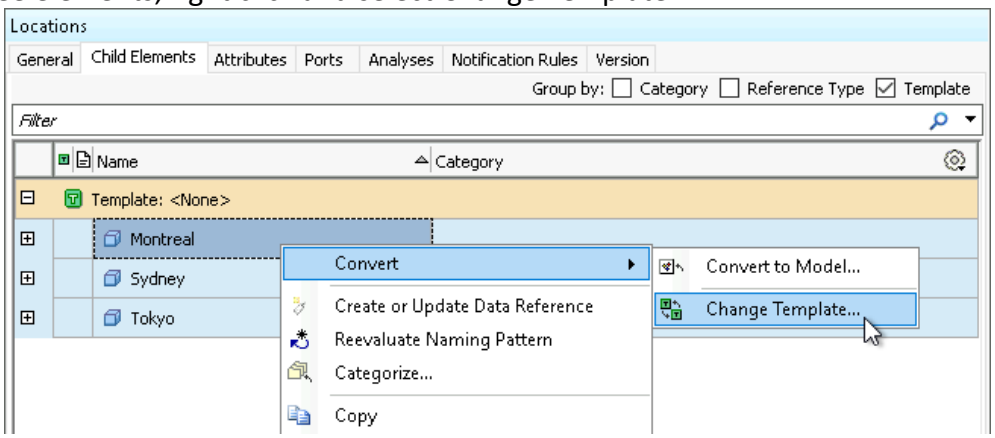


From the Choose Element Template display, select Site. Repeat this process for Sydney and Tokyo.

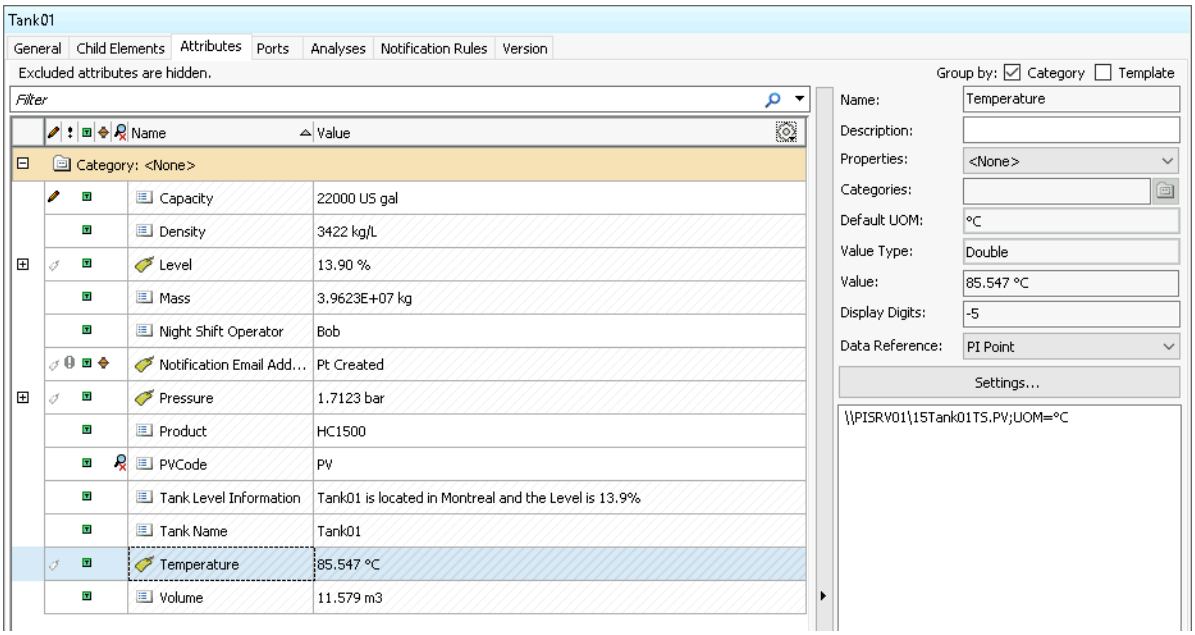


Check in the changes.

Tip: Alternatively, you can navigate to Locations and open the Child Elements tab. Select all three elements, right click and select Change Template.



Validate the Temperature attribute is displaying values by browsing the tanks. You will notice there are values for the Temperature and the PI Point resolved to the Site number + Element Name + TS.PV.

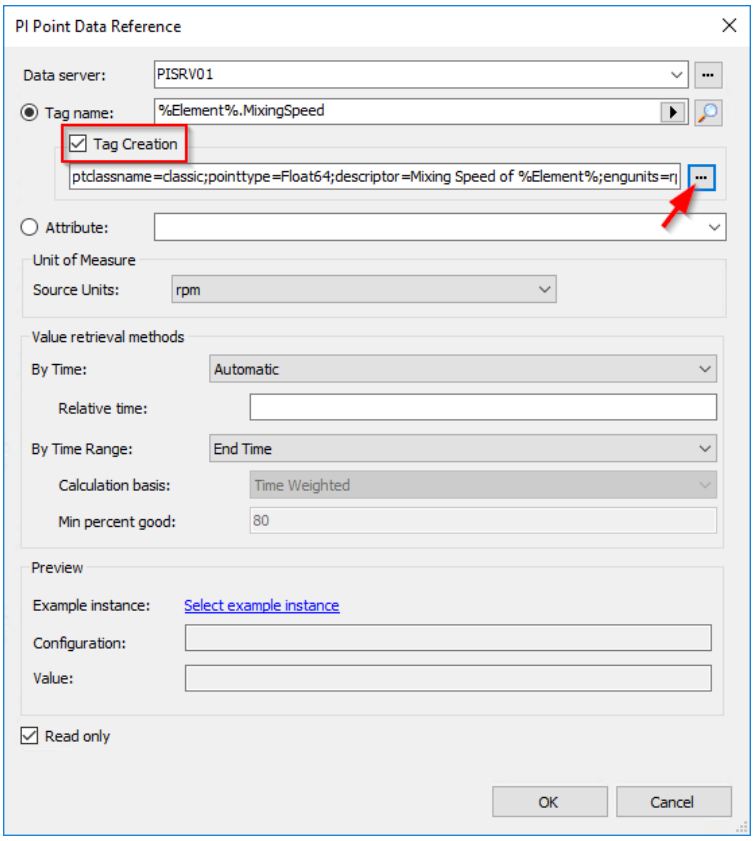

Tank01
General
Child Elements
Attributes
Ports
Analyses
Notification Rules
Version
Excluded attributes are hidden.
Group by: Category Template
Filter
Name
Value
Category: <None>
Capacity 22000 US gal
Density 3422 kg/L
Level 13.90 %
Mass 3.9623E+07 kg
Night Shift Operator Bob
Notification Email Add... Pt Created
Pressure 1.7123 bar
Product HC1500
PVCode PV
Tank Level Information Tank01 is located in Montreal and the Level is 13.9%
Tank Name Tank01
Temperature 85.547 °C
Volume 11.579 m3
Name: Temperature
Description:
Properties: <None>
Categories:
Default UOM: °C
Value Type: Double
Value: 85.547 °C
Display Digits: -5
Data Reference: PI Point
Settings...
\\PISRV01\15Tank01TS.PV;UOM=°C

3.3. Setting Point Attributes for Tag Creation (optional)

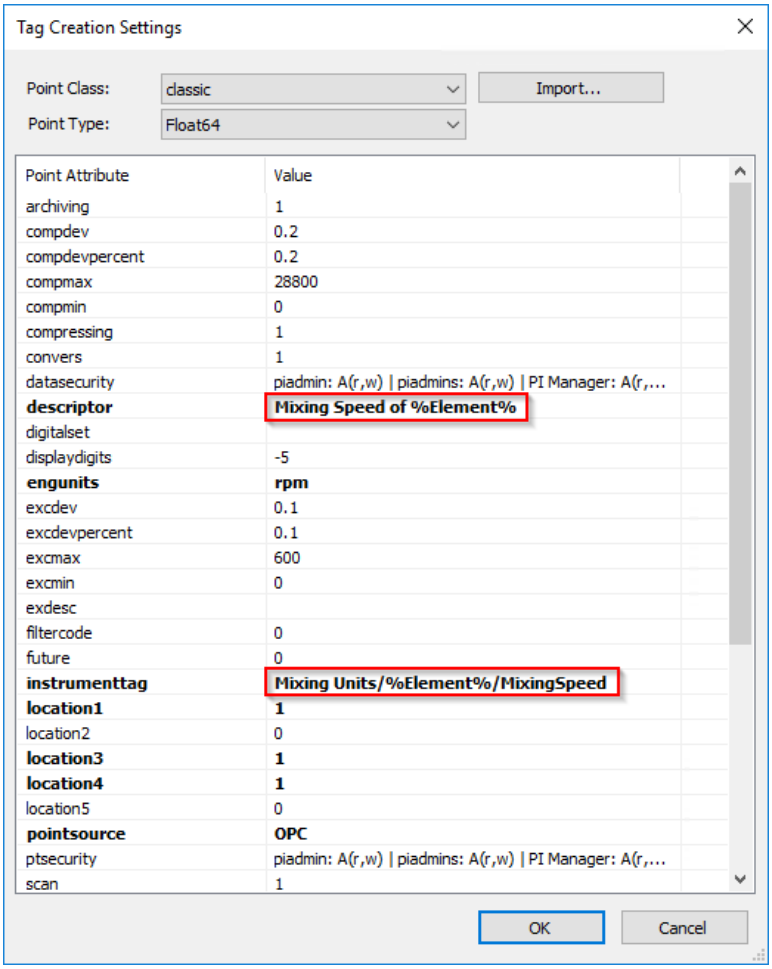
Database: **Tank Farm**

PI Tags can also be created via AF. One way is going via the PI Analysis Service which is creating the tags for writing the calculation results, but another is independent of the data source and is done via template-based attributes with PI Point DR type. Since this is only possible for attributes based on a template, substitution parameters are crucial for defining the tag names and attributes. In the following we will go through an example of setting up tags for an OPC interface via PI Point DR attributes.

Before going through the steps of setting up an attribute for tag creation, let us first look at the attribute **Mixing Speed** which was built in a similar way. In the PI Point Data Reference Settings, note that Tag Creation is ticked. The Point Attribute configuration which is partly visible in the field below can be configured by clicking button with the three dots.



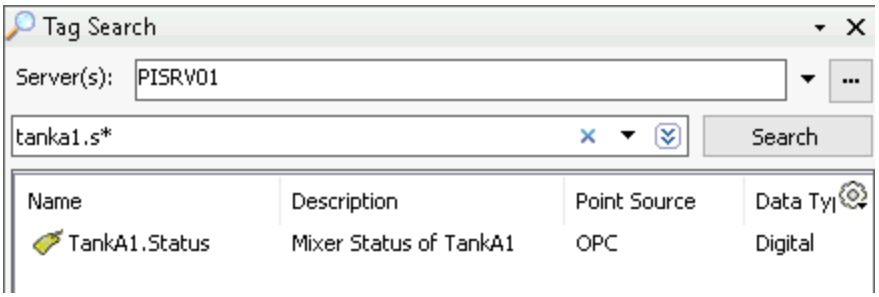
In the Tag Creation Settings window, note the use of substitution parameters for the descriptor and instrumenttag (in our case this is the address on the OPC Server).



We will now go through an example of setting up a new attribute for tag creation. In our database we want to include the **status** of the mixing units which are part of the tanks. The PI System Administrator has already set up the tag for TankA1 which will serve as a template for the configuration. We can find the tag by the name TankA1.Status.

Note that in a real-life scenario, you should have some expertise on tag creation and the interface in question. If you do not have this kind of expertise, speak to your local PI System Administrator to get assistance. In our example, we use tags from an OPC interface and use a configuration in which it is obvious which parts need to change from tank to tank.

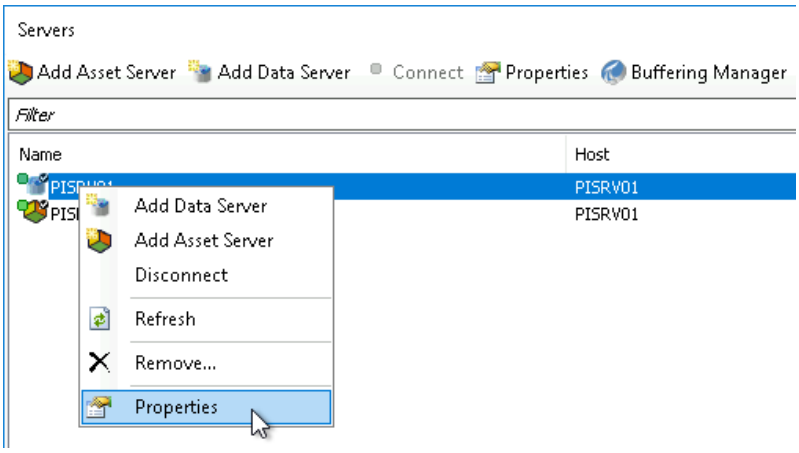
1. We open the Attribute Templates tab of the Tank template in the Library. We can again make use of the Palette to quickly build the new attributes. Open the Tag Search Palette from the menu with View > Palette > Tag Search.
2. In the Tag Search enter TankA1.S* to find the pre-built tag.



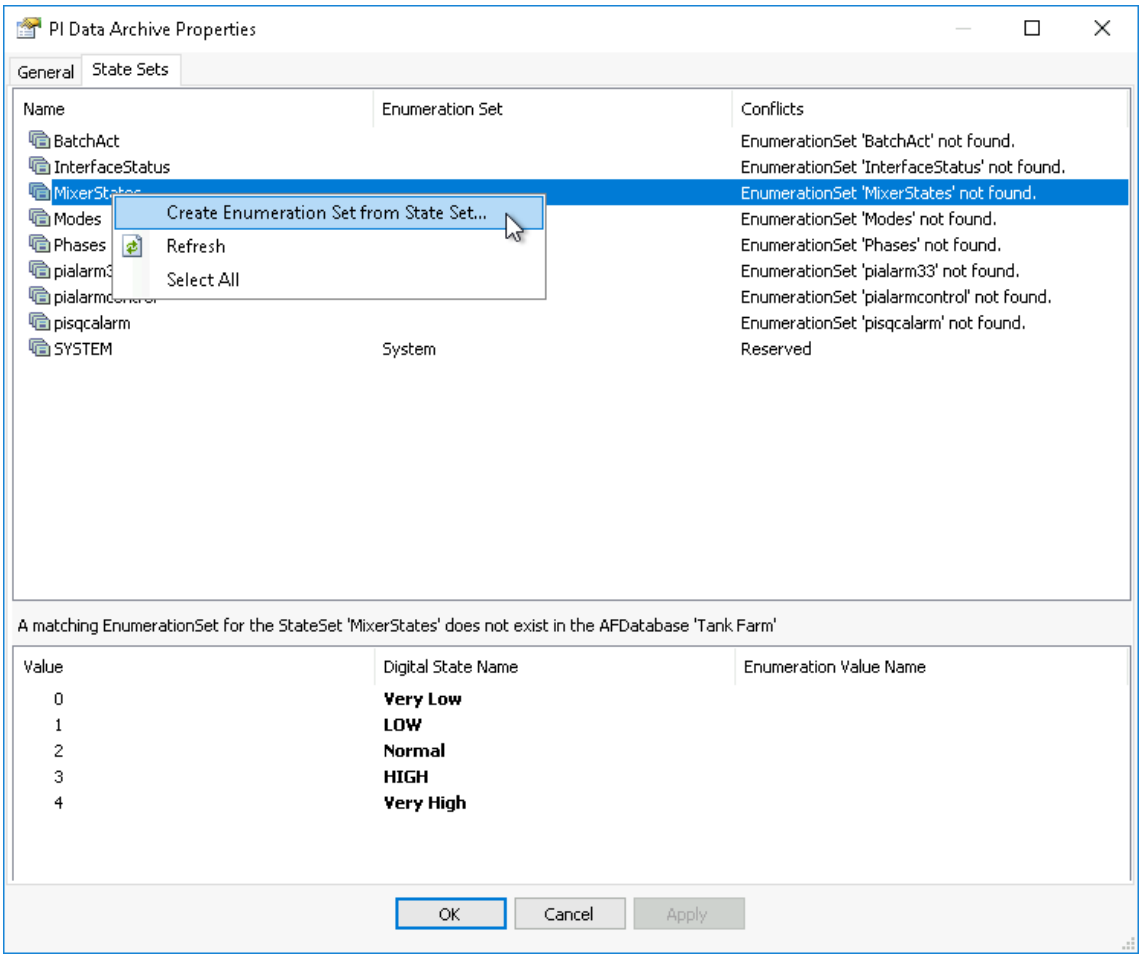
3. Drag and drop the tag to the Attribute templates area. The attribute is created with the name of the tag. We will apply some changes to the configuration.

4. Rename the attribute to “Mixer Status”.

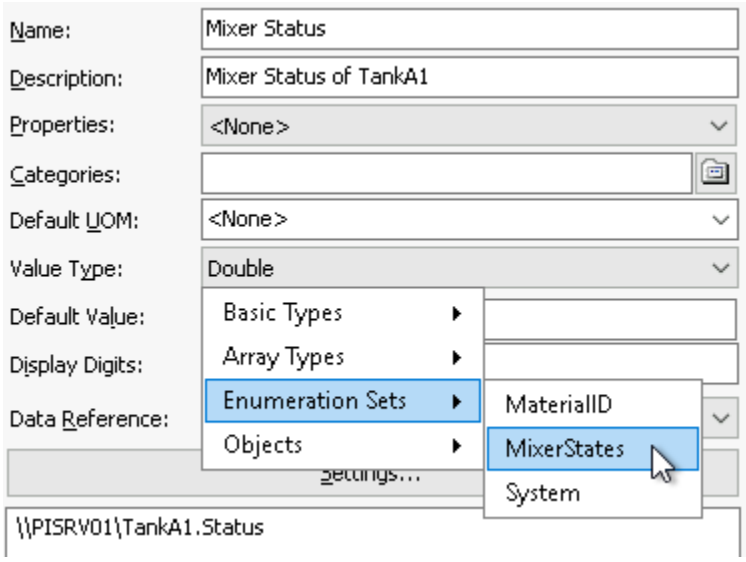
5. Looking at the configuration of the Status attribute we notice that the tag is digital and we still need to create an Enumeration Set to match the Digital State Set. Later in the course we will discuss Digital State Sets and Enumeration Sets in more detail. In the menu we open File > Connections, right click on the Data Server and select Properties.



6. In the PI Data Archive Properties window we navigate to the State Sets tab. Here we right click on the set MixerStates and select Create Enumeration Set from State Set… and confirm with Yes to create a matching Enumeration Set.



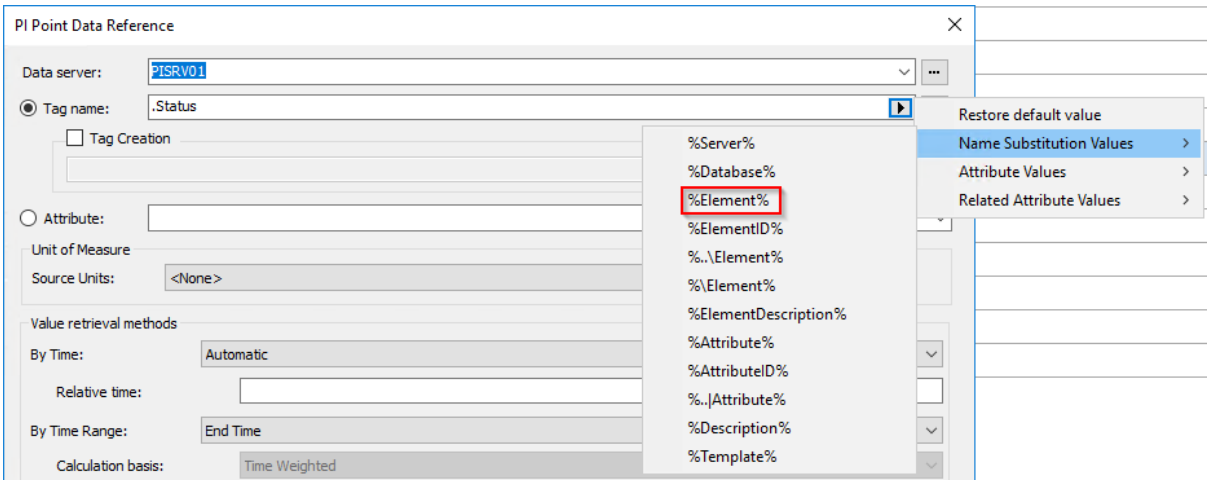
7. Now we can assign the Enumeration Set MixerStates as the Value Type for the attribute template Mixer Status.



8. We do not need the description (which came from the PI tag) in the attributes since the attribute names are descriptive enough, so we will delete those.

9. Now that the basic configuration is set, we can configure the template further to create the tags for the other tanks. For this we click Settings… on the Mixer Status attribute to open the PI Point Data Reference window.

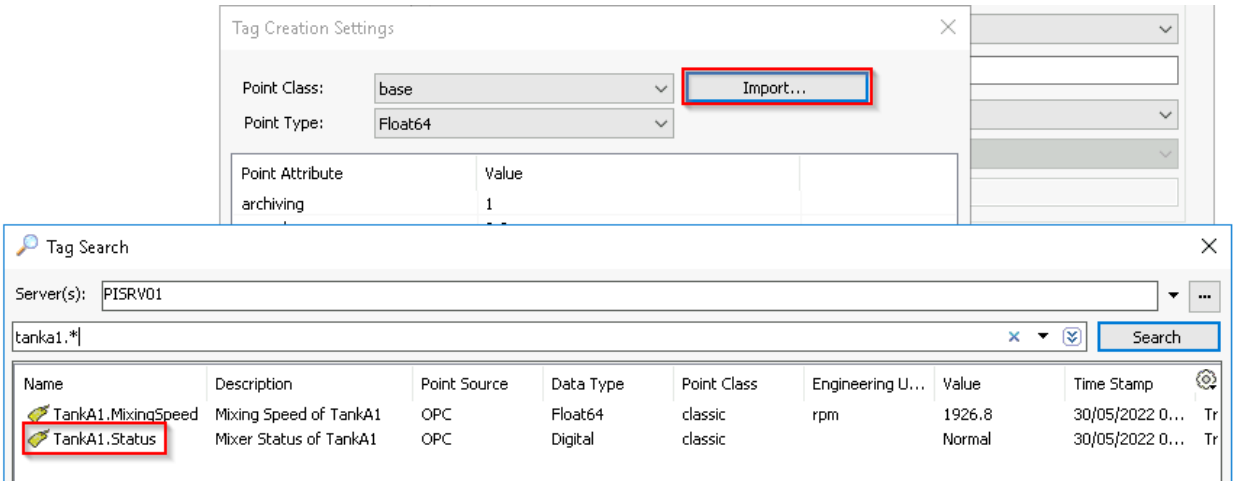
10. First we change the Tag name to create tags according to the naming convention <tank name>.Status. To do this we replace TankA1 with %Element% either by entering the substitution parameter manually or through selection from the Name Substitution Values list as shown below.



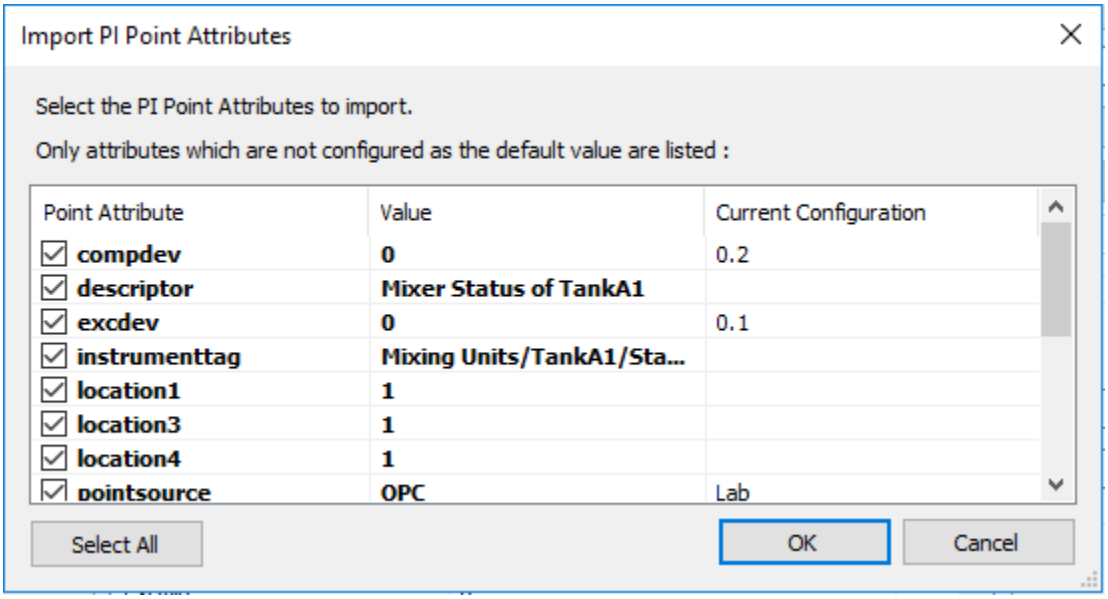
In the end, we should have %Element%.Status in the Tag name field.

11. Next, we tick the box for Tag Creation and click the ellipsis button on the right to open the Tag Creation Settings window.

12. In the Tag Creation Settings window we can manually make the required changes, but instead of starting from scratch it is easier to use an existing tag as a starting point. For doing so, we use the Import… button and search for the tag TankA1.Status which we used earlier to build the attribute.

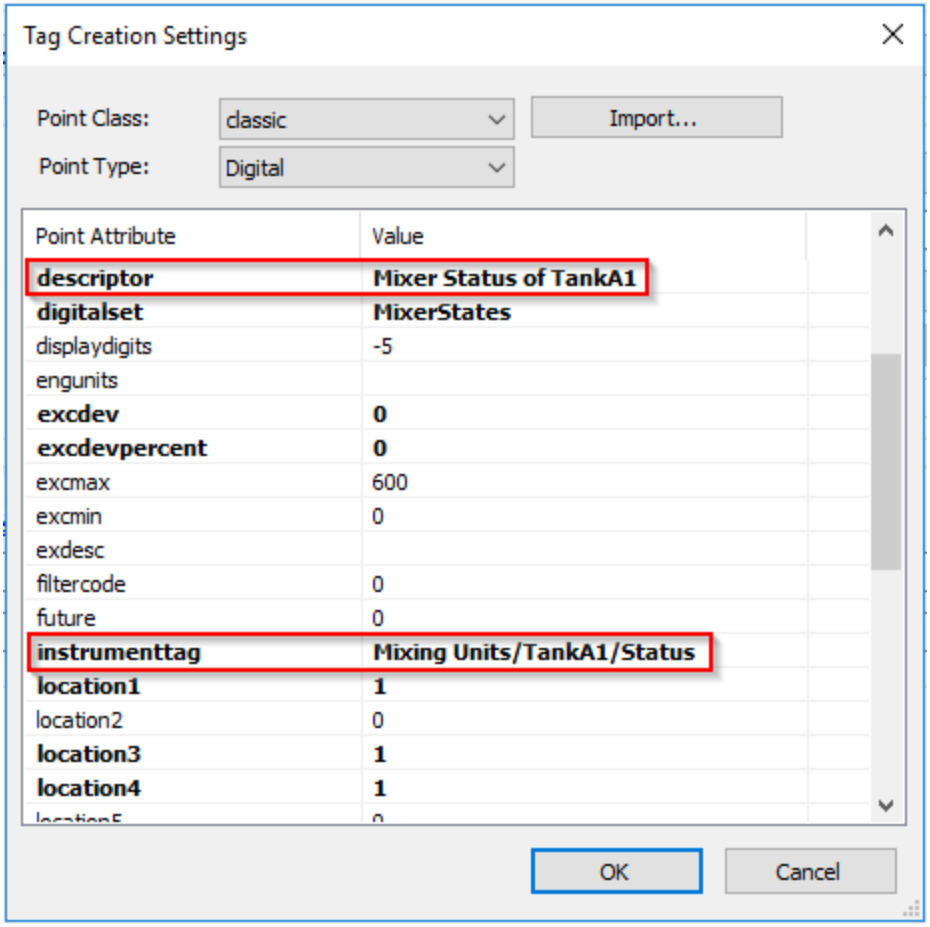


13. Next, a window opens to show all the differences to the default setting and we can decide for each Point Attribute if we want to import the settings from the example tag or stick with the default (or what is presented as the current configuration).

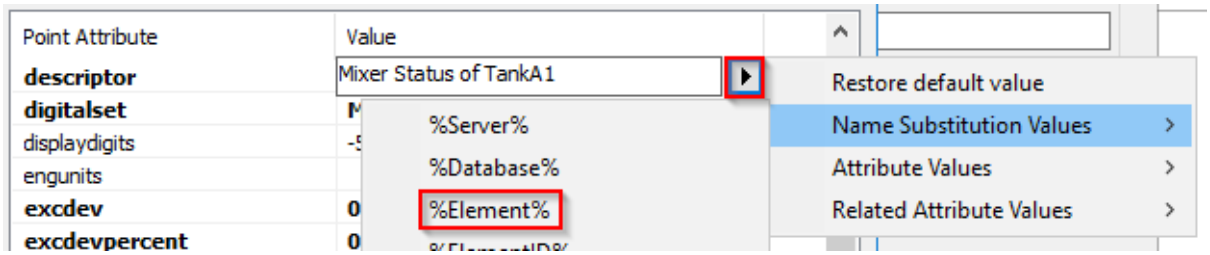


In this case, we will click OK to take all settings from the imported tag. Note that we still need to make changes to the configuration to make it universal for all tanks which we will do in the next step.

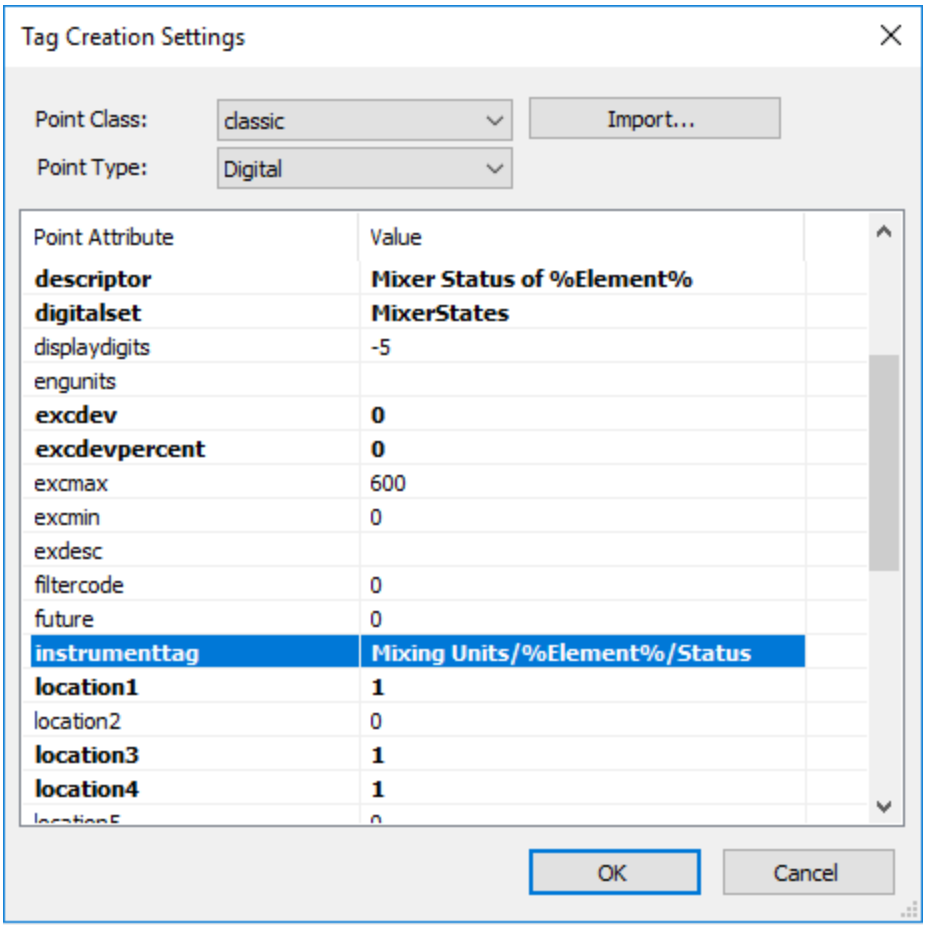
14. The Tag Creation Settings window now shows all Point Attributes which are differing from the default in bold. We still need to make changes where the configuration needs to be different between the tanks. This is the descriptor and the instrumenttag (this corresponds to the address on the OPC server).



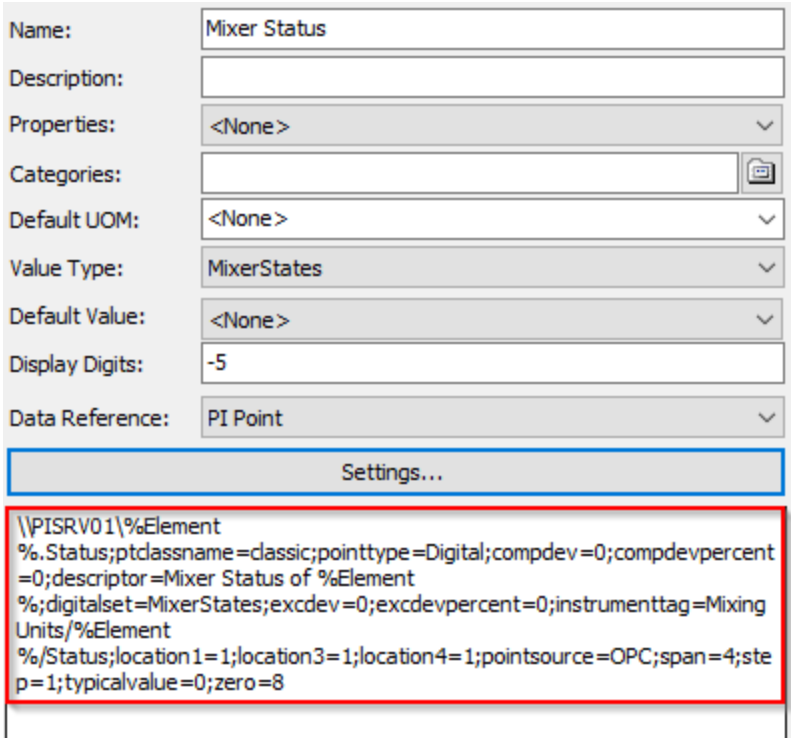
15. To make changes we click on the field and can again either type in the substitution string use the arrow for assistance.



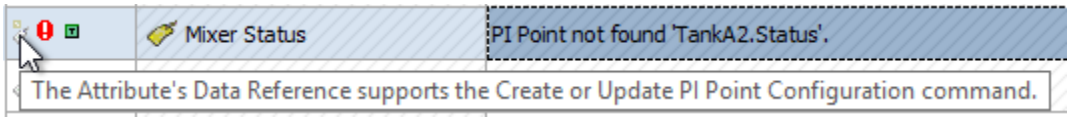
In this case we replace TankA1 with %Element% and do the same for the instrumenttag. The result should look like below.



16. We can close out of the windows with OK and OK. All changes to the default are displayed in the configuration string of the attribute.

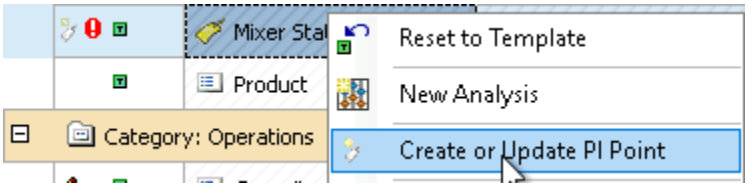


17. When we navigate to the Elements tab and look at a different tank than TankA1, say TankA2, we see the following.



The PI Point was not found since it was not created yet. The little white tag symbol on the left indicates that we can apply the Create or Update PI Point command (see "*Attribute indicators for updates of PI point data references*" in the OSIsoft Documentation for details).

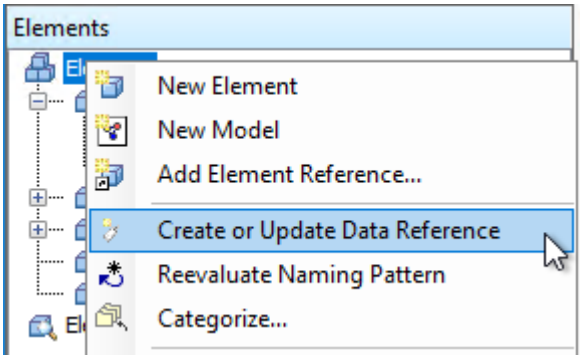
18. We can right click and apply Create or Update PI Point to create the tags.



Note that the attributes will probably show Pt Created when the tags got created since it can take a bit until the interface is aware of the new tag and the first values are written. How long this takes will depend on the interface setting how often to check for updates in the tag list (typically every 2 minutes) and how frequent the tag is getting new values. In our case it should not take longer than 2 minutes as the tags are updated frequently.

We do not need to perform this for each attribute individually as the Create or Update Data Reference option is also available from the element tree. It

can be performed from any element and would be applied to this element and all its descendants. In our case it is easiest to perform this on the highest level directly.



If this seems complicated, remember that this is just moving administration work to AF which might otherwise have been done within PI System Management Tools or PI Builder. The task of building the tags has to be done somewhere. The advantage of including it in the template is that whenever we build a new element from the template the tags will be created automatically (as long as we built the instrumenttag in the correct way to apply to all elements).

For full information refer to "*Configure creation of PI points*" and "*Edit PI point properties*" in the OSIsoft Documentation.

4. Tables

4.1. Parameterized Queries

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Parameterized Queries

Database: **Sweet & Savory Corps**

In this video, we explore data retrieval from a table.

Tables in the AF database provide contextual information through the Table Lookup data reference. Tables can provide information about the equipment or process, store tag names or other configuration information.

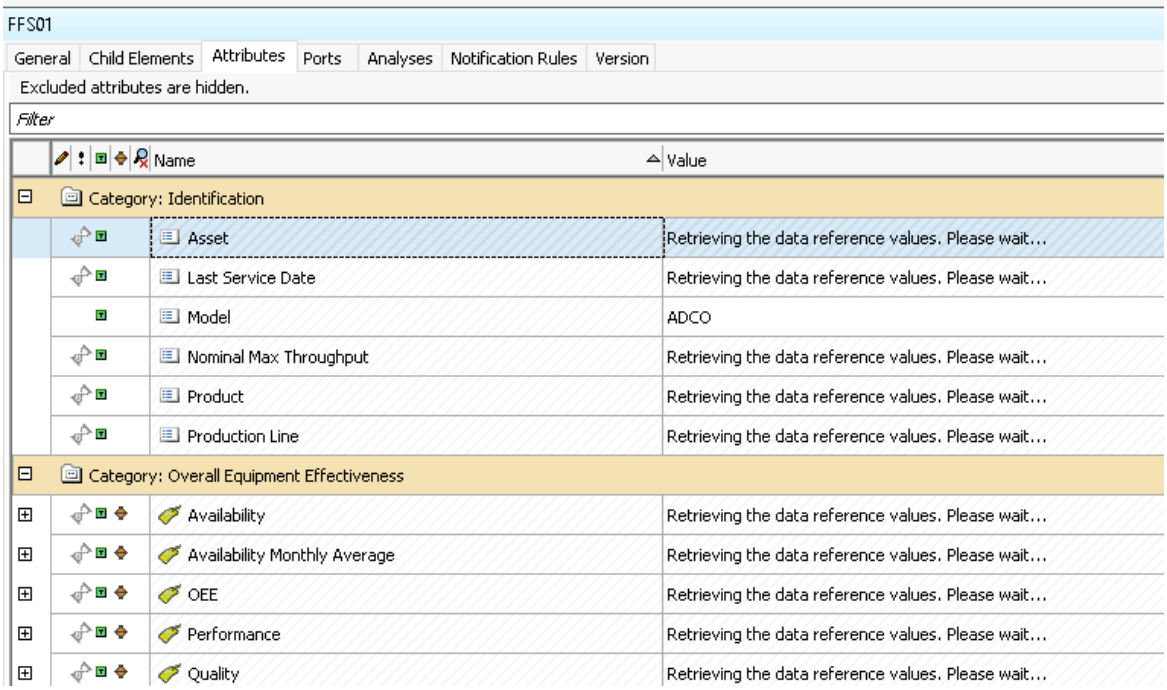
Tables can be created internally, imported from an external relational or tabular data source, or linked dynamically to an external relational data source. In this way, Tables can expose information in maintenance, production planning, or equipment databases for use by PI client applications.

Switch to the Sweet and Savory database.

Navigate to Elements tab → Sandbox → Packaging → FFS01 and click on the Attributes tab. It may take a few minutes for the data to update.

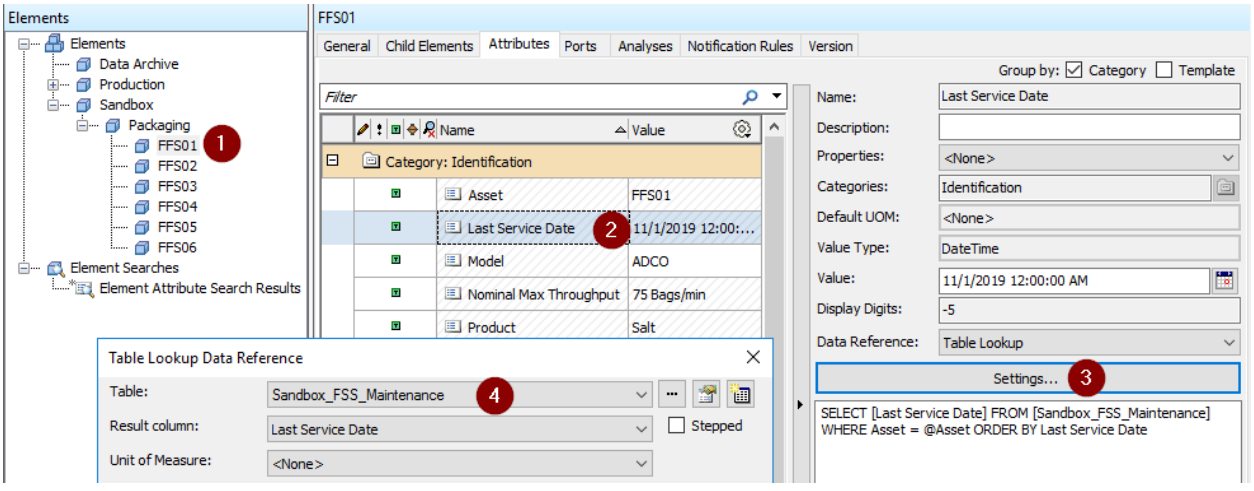
When you opened PSE and navigated to the Sandbox_FFS_Maintenance table in the Sweet & Savory Corps database, you may notice it takes a long time to initially open the table.

Why is the response so slow?



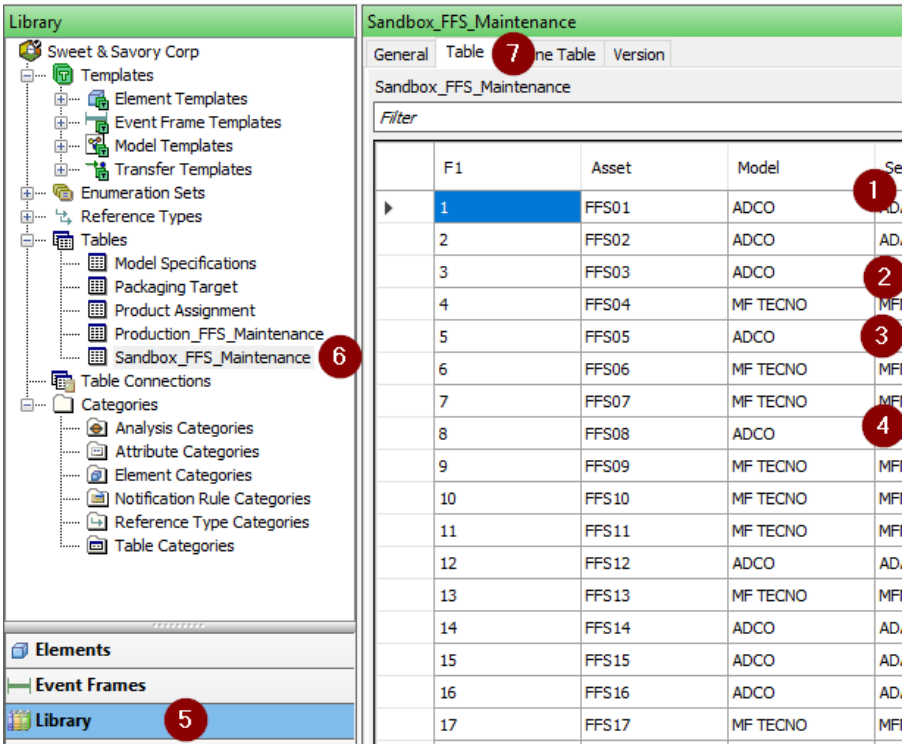
Based on the set-up of the Sandbox_FFS_Maintenance table, we suspect the slowness is caused by a single heavy hitter, the attribute **Last Service Date**. This attribute is a Table Lookup, which means that it… looks up a value in a table, using a SQL-like query of type:

SELECT [Last Service Date] FROM [**Sandbox_FFS_Maintenance**] WHERE Asset = @Asset ORDER BY Last Service Date



Let's look at the table we use to populate the Last Service Date attribute:

1. Move to the Library tab and open the Tables dropdown list. (5 in image below).
2. Select table (6 in image below)
3. Move to the Table tab. (7 in image below)



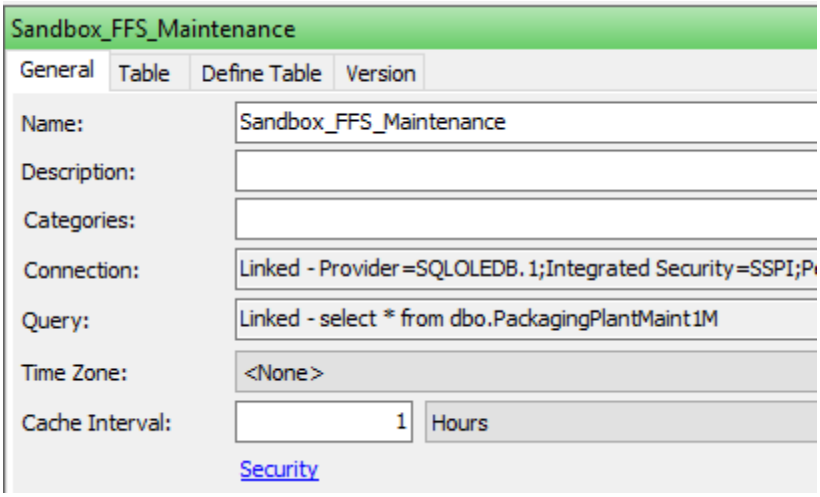
How many rows are present in this table? *[hint: far too many!]*

Do we need all these rows? *[hint: no, we only need the FFS Units that appear in our database!]*

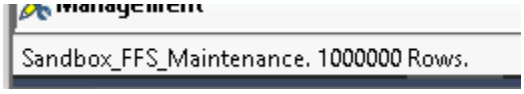
The table Sandbox_FFS_Maintenance is stored in a central SQL Server at the corporate level and contains asset metadata from all the production sites around the world (around 1M FFS units… probably unrealistic, but good for our discussion!). For our goals, we only need to retrieve the data for our pilot site. This has already been done correctly in the Production environment, so we just need to understand how it works and replicate it in the sandbox environment.

4. Still on table Sandbox_FFS_Maintenance, move to the General tab.
5. Click on the "**Link…**" button. You'll see a Connection String and a query to an external SQL Server. The current query is retrieving all rows and columns from the external SQL table:

   select * from dbo.PackagingPlantMaint1M

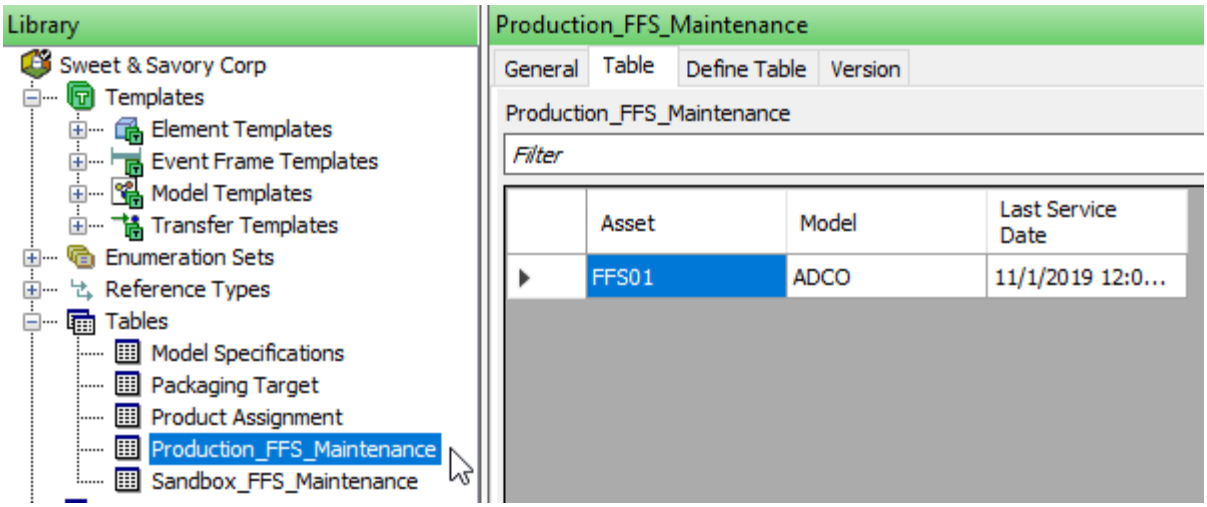


It may take a few minutes to retrieve all the rows:

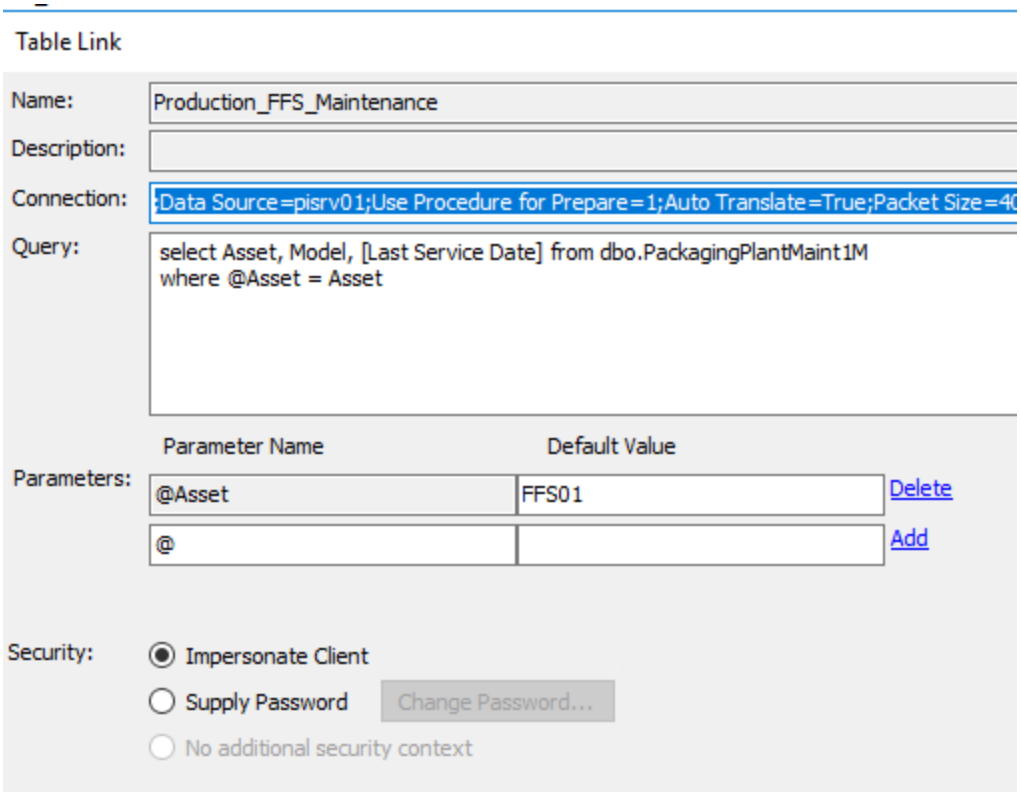


6. Once complete, the number of rows will display.
7. Select table **Production_FFS_Maintenance**.
8. Move to the Table tab, how many rows do you see?

NB: The Table tab shows only one row corresponding to the default value of the @Asset parameter (FFS01), but the query will work fine for any other AF element.



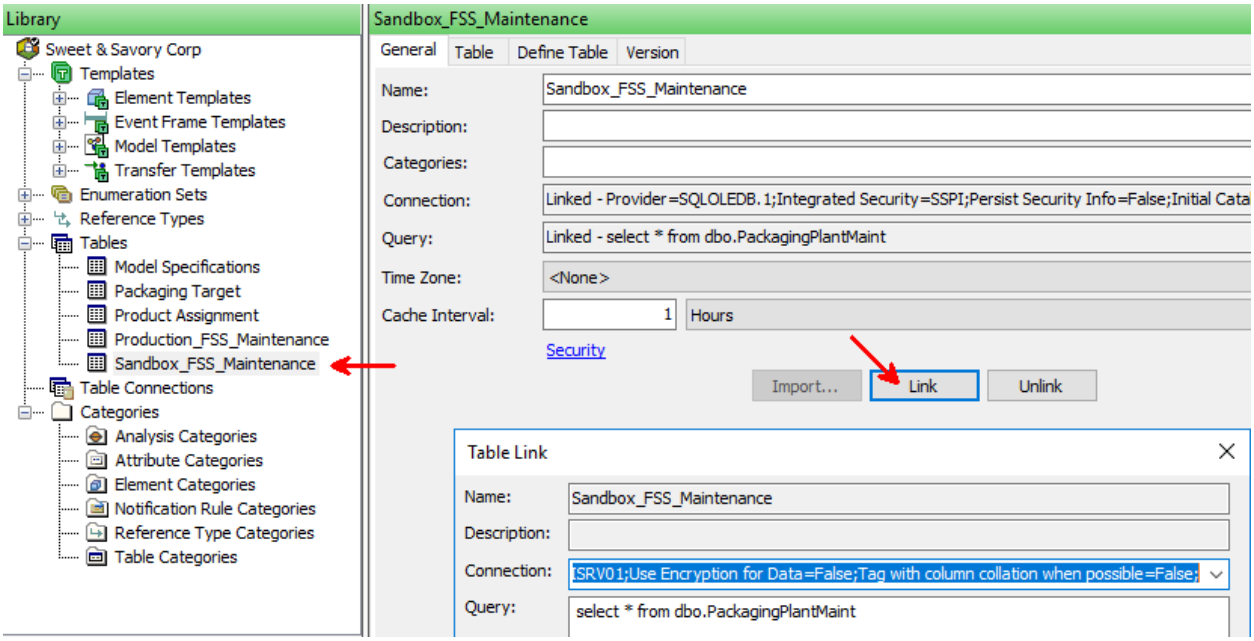
9. Go to General --> **"Link…"** and check how the query is performed. You'll notice that the SQL query contains a parameter (*where @Asset=Asset*). You can hit Cancel to close the window.



Using parameters in a linked table query is useful to limit the number of rows returned from a very large external table. You can add conditions and parameters to return more targeted results, such as all rows that include a device or manufacturer ID number, specific for each table lookup data reference. (*Documentation*).

Now go back to the Sandbox table and edit the query so that only needed information is returned:

10. Open table Sandbox_FFS_Maintenance → General tab.
11. Select the **Link…**



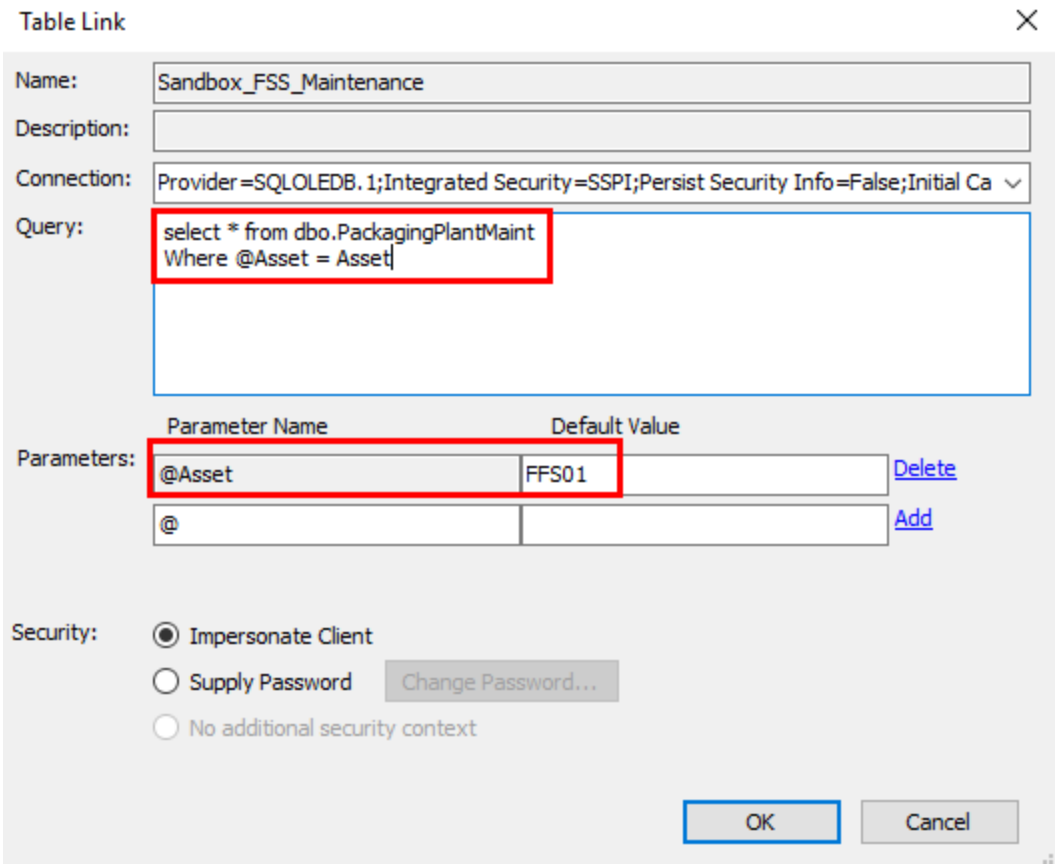
12. Edit the query as follows (feel free to copy and paste from the .PDF file of this workbook):

Select * from dbo.PackagingPlantMaint

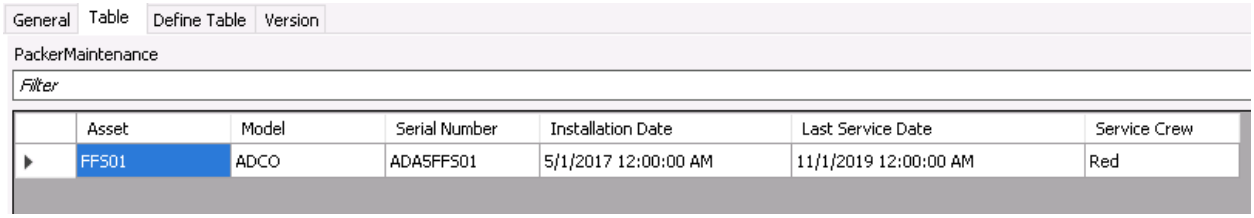
Where @Asset = Asset

13. In the Parameters Section, add the parameter @Asset with a default value of FFS01. Press Add (located to the right of the Default value box) to include it in the query statement at run time.

Parameter Name	**Default Value**
@Asset	FFS01



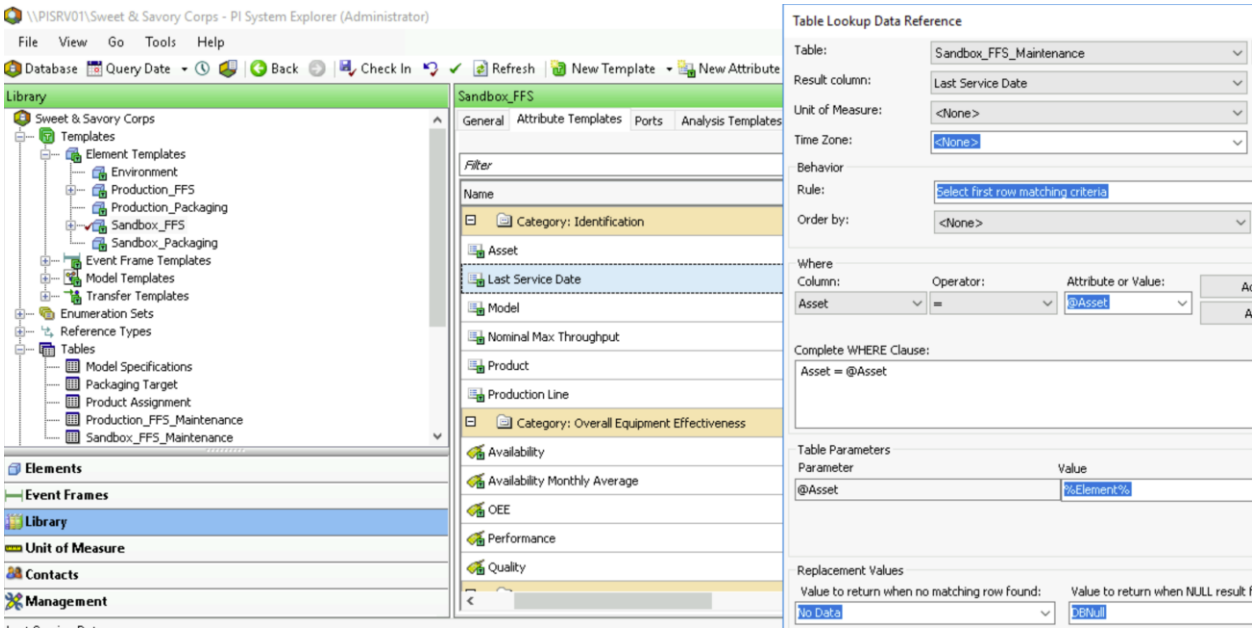
14. Check In and Click on Refresh and open the Table tab: how many rows are present now?



Only one row is returned, and it corresponds to the Default value of the SQL query.

Note: as stated before, the parameter value in the configuration above is just an example parameter value (a default value) and it doesn't affect the query when evaluated for a different AF element.

15. Move to the Templates → Element Templates → **Sandbox_FFS** and select the Attribute Templates tab.
16. Select the **Last Service Date** and click on Settings...
17. Add the Table Parameter as in the screenshot below (Parameter: @Asset; Value: **%Element%**).



In this way, every time a value is requested by a Table Lookup attribute, the parametrized query will be passed to the external SQL server and only the requested value will be returned.

18. Important: Click on 

to save your changes.

19. In order to verify whether the caching performance has improved, close PI System Explorer. Then reopen PI System Explorer and move between AF elements under the Sandbox branch (make sure you have the Attributes tab open). Do you notice any improvements in terms of data retrieval speed? [*hint: yes! :D*]

Note: Don't forget the **square brackets** around Last Service Date. The space characters in the column name Last Service Date would break the SQL query unless square brackets are used.

Still hungry for knowledge on AF Tables? For more information on parametrized queries, please visit *Data references from outside the PI System* and *Parameters for linked table queries*.

Last but not least, this PI Square post illustrates good and bad uses of Table Lookup, Formula and Analysis data references: *Asset Analytics Best Practices - Part 3: Input Attributes* (5).

✓ Best Practice	Don't bring data you don't need into the PI System. In the case of table lookups, you can use parametrized queries to only fetch data that will be used. A rule-of-thumb is to avoid importing tables with over 10 000 rows.

4.2. Introduction to Building SQL Queries for Targeted Data Retrieval (optional)

To retrieve data from relational databases we need to build queries which specify which data is needed. While those queries can be kept simple to retrieve a complete table it is often times better to make a more specific request to limit the amount of data and therefor keep the processing load small.

SQL stands for Structured Query Language. SQL is an American National Standards Institute (ANSI) definition for the language used to communicate with relational database systems. It is used by virtually all relational databases in the world today. (Even the PI Data Archive has a SQL Subsystem that can act as a translator to make it "look" like a relational database). SQL Commands are often called "**SQL Statements**." They can be executed interactively or as stored procedures.

While SQL is a standard and every relational database you encounter will understand it there are slight differences as most databases have unique extensions and/or syntaxes that are unique to those systems.

To give a simple example, when passing dates into Access you use pound signs (#) for surrounding dates. On the other hand, in SQL Server you need to use apostrophes (').

Access: [...] WHERE dtColumn >= #2001-11-05#

SQL Server: [...] WHERE dtColumn >= '20011105'

A common SQL syntax starting command is **SELECT** which is used to query the database. The data retrieved from the statement is based on the criteria specified in the SELECT statement. Following the **SELECT** command identifies the columns to be selected from the tables(s).
The **FROM** command identifies the first (or perhaps only) table being queried.
The **WHERE** command contains criteria to filter the data being retrieved.

The conditional operators include:

- equal (=)
- greater than (>)
- less than (<)
- greater than or equal (>=)
- less than or equal (<=)
- not equal to (<>)
- LIKE (which is a pattern matching operator)

Note: If the conditional clause is set to compare to text, the text value is encased in single quotes ('text').

ORDER BY clause

To guarantee the order of rows returned by a query, the ORDER BY clause can be used. Ascending order is assumed by default if not specified otherwise. To be explicit, ASC for ascending can be placed behind the column to sort by, or if sorting in descending order is needed an explicit DESC needs to be set.

If the column which is used for sorting does not have unique entries, additional columns can be used. After each column the sorting order can be specified with ASC or DESC.

AND and OR statements

- **AND** indicates both statements must be TRUE for the row to be returned when the query is executed.
- **OR** returns data rows if either condition is met

The **LIKE** operator is used to search for a specific pattern in a column. In conjunction with the LIKE operator a **wildcard of %** is used for comparison. The % can represent a single character or multiple characters. Another wildcard is the underscore (_) which can be used to represent a single character.

To work with column/table names which have special characters, such as a space, use square brackets.

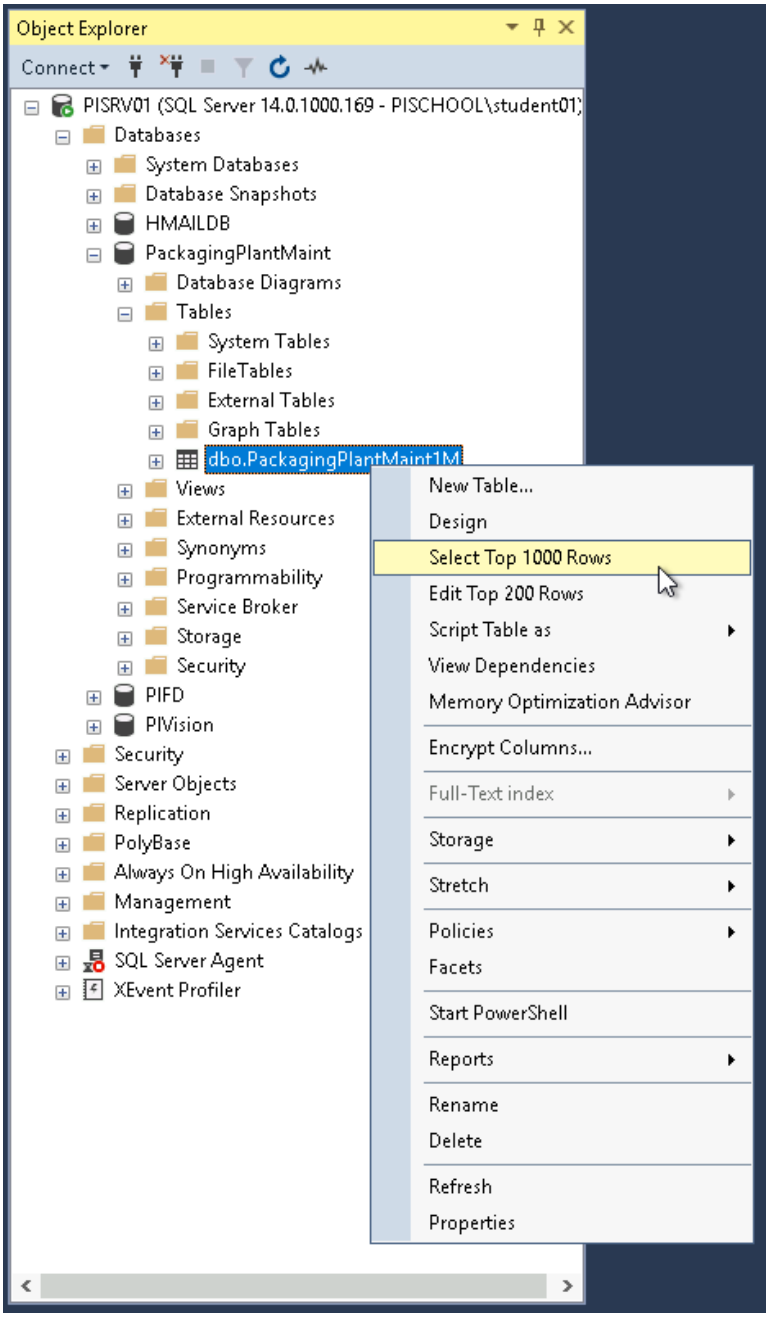
Any name may be wrapped in square brackets, so when in doubt as to what constitutes a special character, wrap the name in square brackets.

Examples:

In the last section we have seen where SQL queries can be used in linking and importing tables in AF. For more interactive testing we will use the **Microsoft SQL Server Management Studio** in the examples below. The same queries could be used in PSE.

When we open Microsoft SQL Server Management Studio we find the table dbo.PackagingPlantMaint1M in the database PackagingPlantMaint. To get us started

we can run a predefined query by selecting “Select Top 1000 Rows” from the menu we get on right mouse click.

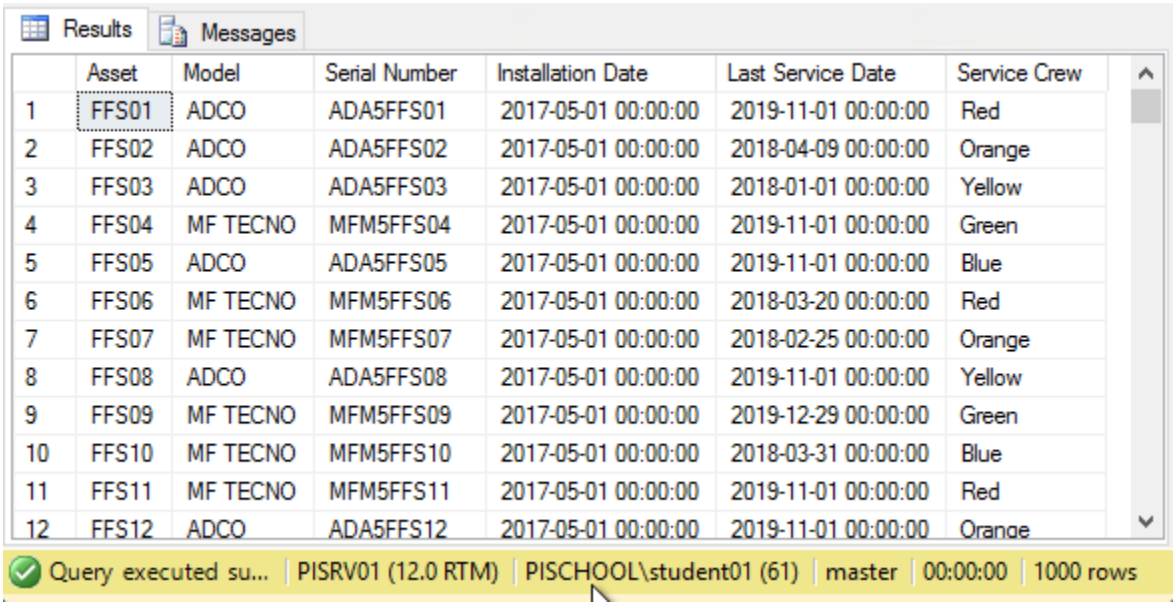


This produces the following query:

```
SELECT TOP (1000) [Asset]
      ,[Model]
```

```
      ,[Serial Number]
      ,[Installation Date]
      ,[Last Service Date]
      ,[Service Crew]
FROM [PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]
```

The resulting table looks like this:



Results | Messages

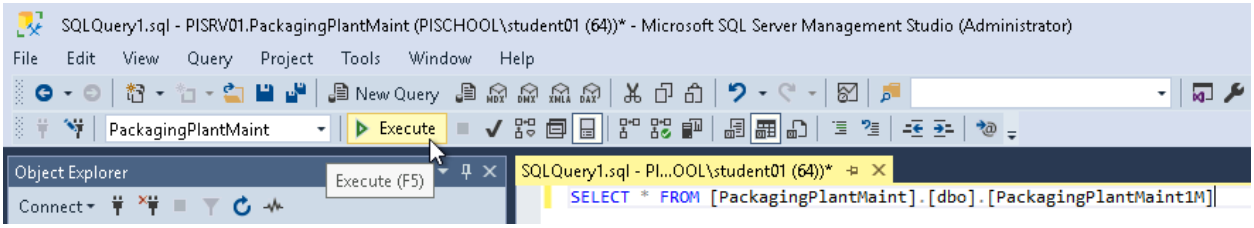
	Asset	Model	Serial Number	Installation Date	Last Service Date	Service Crew
1	FFS01	ADCO	ADA5FFS01	2017-05-01 00:00:00	2019-11-01 00:00:00	Red
2	FFS02	ADCO	ADA5FFS02	2017-05-01 00:00:00	2018-04-09 00:00:00	Orange
3	FFS03	ADCO	ADA5FFS03	2017-05-01 00:00:00	2018-01-01 00:00:00	Yellow
4	FFS04	MF TECNO	MFM5FFS04	2017-05-01 00:00:00	2019-11-01 00:00:00	Green
5	FFS05	ADCO	ADA5FFS05	2017-05-01 00:00:00	2019-11-01 00:00:00	Blue
6	FFS06	MF TECNO	MFM5FFS06	2017-05-01 00:00:00	2018-03-20 00:00:00	Red
7	FFS07	MF TECNO	MFM5FFS07	2017-05-01 00:00:00	2018-02-25 00:00:00	Orange
8	FFS08	ADCO	ADA5FFS08	2017-05-01 00:00:00	2019-11-01 00:00:00	Yellow
9	FFS09	MF TECNO	MFM5FFS09	2017-05-01 00:00:00	2019-12-29 00:00:00	Green
10	FFS10	MF TECNO	MFM5FFS10	2017-05-01 00:00:00	2018-03-31 00:00:00	Blue
11	FFS11	MF TECNO	MFM5FFS11	2017-05-01 00:00:00	2019-11-01 00:00:00	Red
12	FFS12	ADCO	ADA5FFS12	2017-05-01 00:00:00	2019-11-01 00:00:00	Orange

Query executed su... | PISRV01 (12.0 RTM) | PISCHOOL\student01 (61) | master | 00:00:00 | 1000 rows

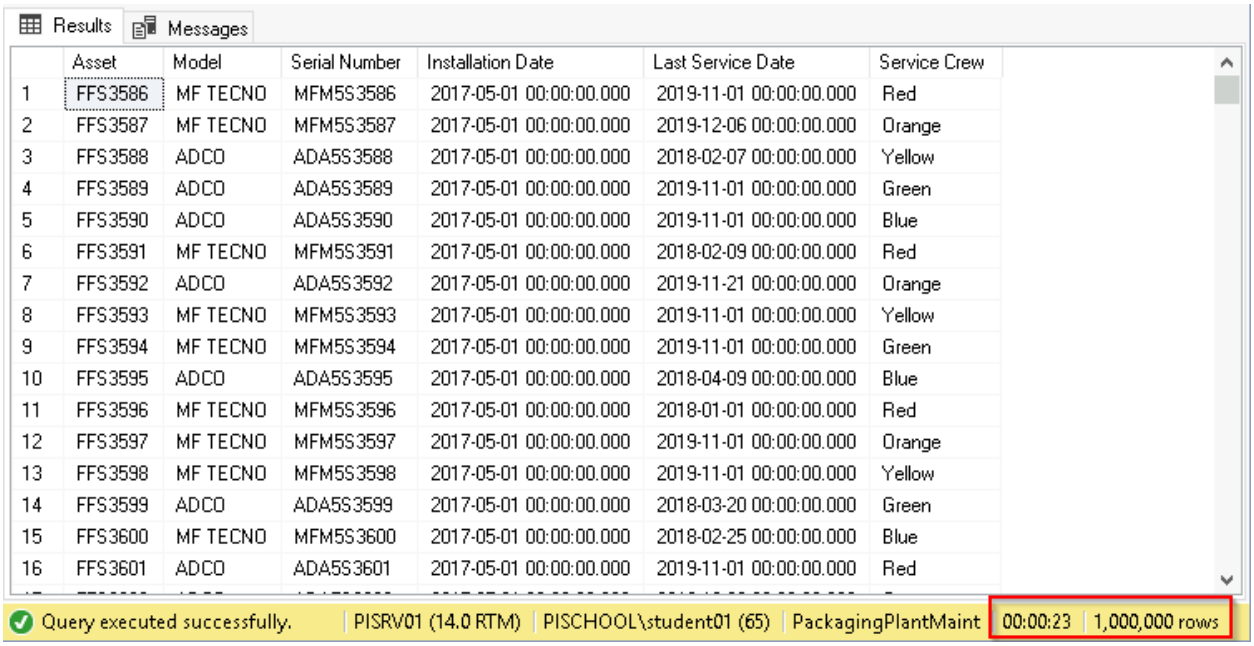
The easiest query however would be to just select everything from this table:

```
SELECT * FROM [PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]
```

To test, we can modify the query and execute.



The result will show the entire table and displayed below the Results window we can see how many rows of data we have.



Results | Messages

	Asset	Model	Serial Number	Installation Date	Last Service Date	Service Crew
1	FFS3586	MF TECNO	MFM5S3586	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Red
2	FFS3587	MF TECNO	MFM5S3587	2017-05-01 00:00:00.000	2019-12-06 00:00:00.000	Orange
3	FFS3588	ADCO	ADA5S3588	2017-05-01 00:00:00.000	2018-02-07 00:00:00.000	Yellow
4	FFS3589	ADCO	ADA5S3589	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Green
5	FFS3590	ADCO	ADA5S3590	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Blue
6	FFS3591	MF TECNO	MFM5S3591	2017-05-01 00:00:00.000	2018-02-09 00:00:00.000	Red
7	FFS3592	ADCO	ADA5S3592	2017-05-01 00:00:00.000	2019-11-21 00:00:00.000	Orange
8	FFS3593	MF TECNO	MFM5S3593	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Yellow
9	FFS3594	MF TECNO	MFM5S3594	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Green
10	FFS3595	ADCO	ADA5S3595	2017-05-01 00:00:00.000	2018-04-09 00:00:00.000	Blue
11	FFS3596	MF TECNO	MFM5S3596	2017-05-01 00:00:00.000	2018-01-01 00:00:00.000	Red
12	FFS3597	MF TECNO	MFM5S3597	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Orange
13	FFS3598	MF TECNO	MFM5S3598	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Yellow
14	FFS3599	ADCO	ADA5S3599	2017-05-01 00:00:00.000	2018-03-20 00:00:00.000	Green
15	FFS3600	MF TECNO	MFM5S3600	2017-05-01 00:00:00.000	2018-02-25 00:00:00.000	Blue
16	FFS3601	ADCO	ADA5S3601	2017-05-01 00:00:00.000	2019-11-01 00:00:00.000	Red

Query executed successfully. | PISRV01 (14.0 RTM) | PISCHOOL\student01 (65) | PackagingPlantMaint | 00:00:23 | 1,000,000 rows

For this table there are 1,000,000 rows. We can also see how long it took to retrieve the information from the server. We will get back to this later.

If we are interested only in a **subset of columns**, we can specify those after the SELECT. For example:

```
SELECT [Asset], [Model], [Serial Number], [Service Crew] FROM
[PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]
```

Note that while we can wrap all column names in square brackets in our example it would be only necessary for the columns Serial Number and Service Crew since they contain a space.

To limit the **number of rows** returned the WHERE clause can be used. For example, to only show the assets maintained by the “Red” Service Crew we can use the query:

```
SELECT Asset, Model, [Serial Number], [Service Crew] FROM
[PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]
    WHERE [Service Crew]='Red’
```

In some cases, it might be necessary to **filter using a wildcard**. For example, if we are looking for all assets ending with the number 6:

```
SELECT Asset, Model, [Serial Number], [Service Crew] FROM
[PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]
    WHERE Asset LIKE '%6’
```

Sometimes the **order of results** might be important. For example, if there is a date column and we want to see the most recent entries. We can sort by the column Last Service Date to find the assets with the most recent service:

```
SELECT [Asset], [Model], [Serial Number], [Installation Date], [Last Service Date], [Service Crew] FROM [PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]

ORDER BY [Last Service Date] DESC
```

If we want to see the assets which are around for the longest we can sort by Installation Date:

```
SELECT [Asset], [Model], [Serial Number], [Installation Date], [Last Service Date], [Service Crew] FROM [PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]

ORDER BY [Installation Date] ASC
```

Here, specifying ASC is optional as ascending order is assumed by default.

It is also possible to use a combination of columns if the column used has not unique entries. For example, we might want to see the assets by Service Crew, with the most recent services listed on top:

```
SELECT [Asset], [Model], [Serial Number], [Installation Date], [Last Service Date], [Service Crew] FROM [PackagingPlantMaint].[dbo].[PackagingPlantMaint1M]

ORDER BY [Service Crew] ASC, [Installation Date] DESC
```

4.3. Table Caching

The AF Table cache interval is strictly a client-side property. This also means that each AF Client using the AF Table will have its own cache and cache interval start and end time. The cache interval expires relative to the last time data in the AF Table on the AF Client was updated. Note that while the cache is "automatically" refreshed, **the cache is actually NOT updated on a schedule**; it is only refreshed if the cache interval has expired when the client calls for the AF Table data.

This implies that if your AF client has to keep a large amount of unnecessary table information in memory and / or update this table frequently, you might start seeing performance issues. Parametrized queries means that the extra, unneeded data is processed out on the external SQL side and not on the AF SDK client side (as it would have to be if you were not using parametrized queries) leading to better performance. Thus, you should always try to limit the size of your in memory AF tables as much as possible through parametrized queries.

AF Tables can be one of two types - Internal or External.

- Internal tables are stored in the PIFD database. For an Internal table, when the cache interval expires and the AF Table is refreshed, the AF SDK client will update its cache with data from the AF Server.
- External tables retrieve their data from an external data source. For an External table, when the cache interval expires and the AF Table is refreshed, the AF Client will request the AF Server to make the external query to the external data source. The AF Server will then return the results to the AF Client, which will then update its in-memory copy with these results.

Example:

You have PI System Explorer (PSE) and PI Vision (Vision) both looking at a Table lookup attribute (TL) which itself is set to get its value from an externally linked table. Say the cache interval is set to 1 hours for this table.

You first open PSE and navigate to the TL at 10:00 am. Then you open Vision to a display also looking at this same TL attribute at 10:15 am. At 10:30 am, this value (the one being looked at by TL) in your external table is updated. At 11:00 am, you refresh PSE to get the newest value of TL as reflected in your external table, and both systems match. However, even if you refresh Vision at the same time, you will still see the older value of TL. This is because each client maintains its own cache, and while PSE's cache interval has expired (1 hour) since 10:00 am, Vision's has not (45 min); thus, Vision will not update its cache, and it will still show the older value for this refresh.

Each client's cache will only update if 2 criteria are met:

1. The client requests values from the table.
2. The Cache Interval has expired for that particular client (this is because the AF server will only query the external table on the client's behalf if the client's cache interval has expired relative to the last update time of the client's in memory copy of the table).

5. Working with Templates

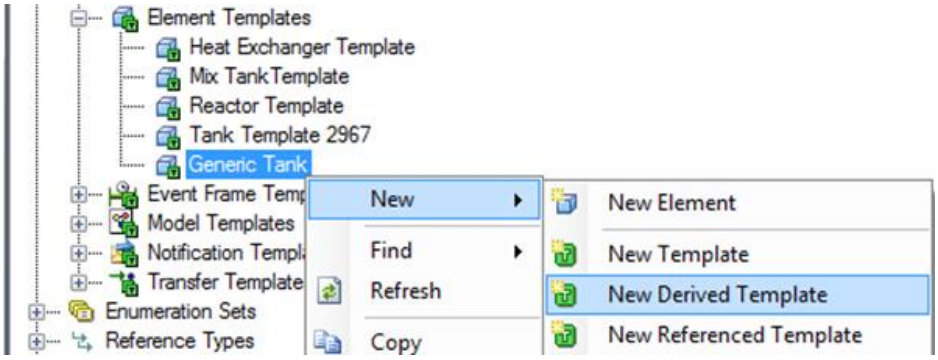
5.1. Template Inheritance and Base Template

A powerful feature of the element template is the ability to set a base template. Once a base template is created, it can be used to create a number of derived templates. When an element is created from a derived template, the element contains all attributes from both the base template and the derived template.

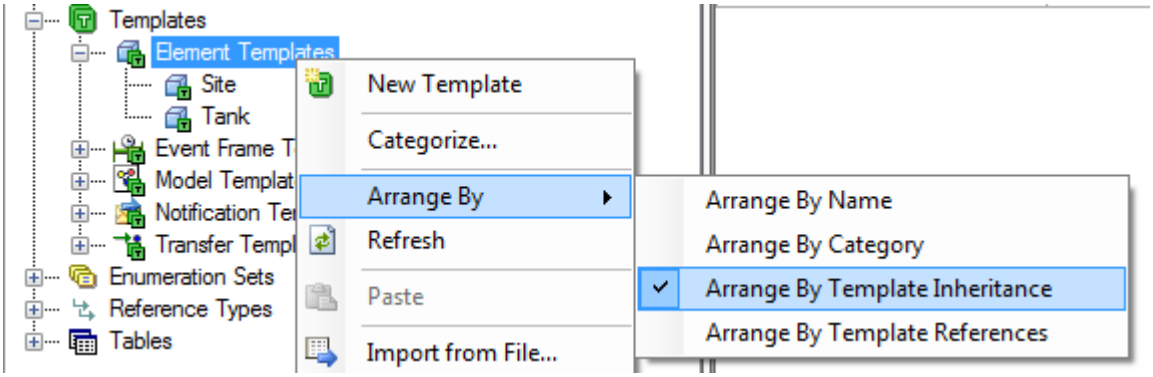
- An element created from the Tank element template has three attributes: Diameter, Height and Fill Level.

- An element created from the Mixing Tank element template has five attributes: Diameter, Height, Fill Level, Mixer Speed and Mixer Blade Size

A base template is best used when you are modeling elements that have a set of attributes in common with a few attributes that differ. For example, if you have a set of tanks, some with two valves and some with one valve, you can create an element template for the one-valve models and use that as the base template for the two-valve models. Set the base template of an element template in the *General* tab; alternatively, you can set the base template at creation time by right clicking the base template and select *New Derived Template*.



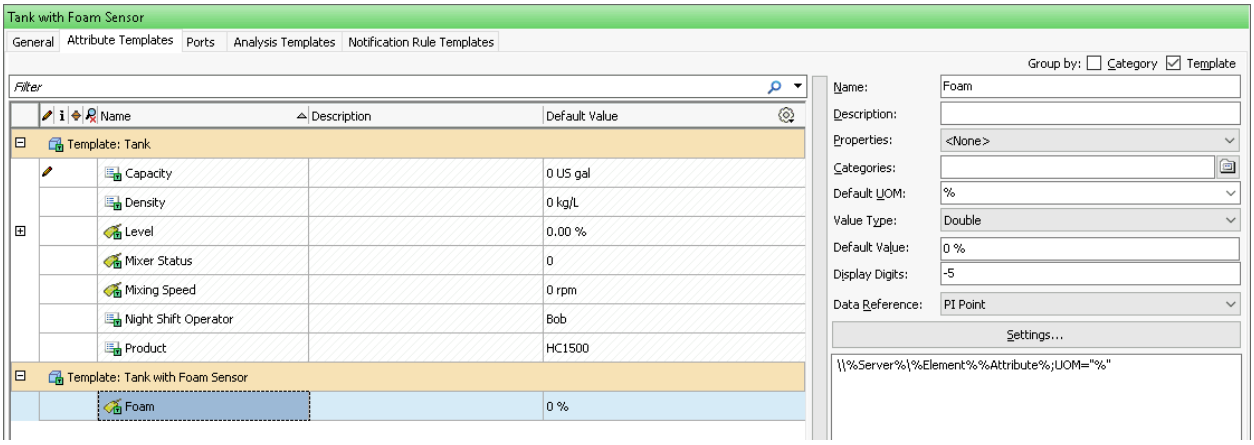
To view the template inheritance tree from the PSE Library, simply organized the templates by inheritance.



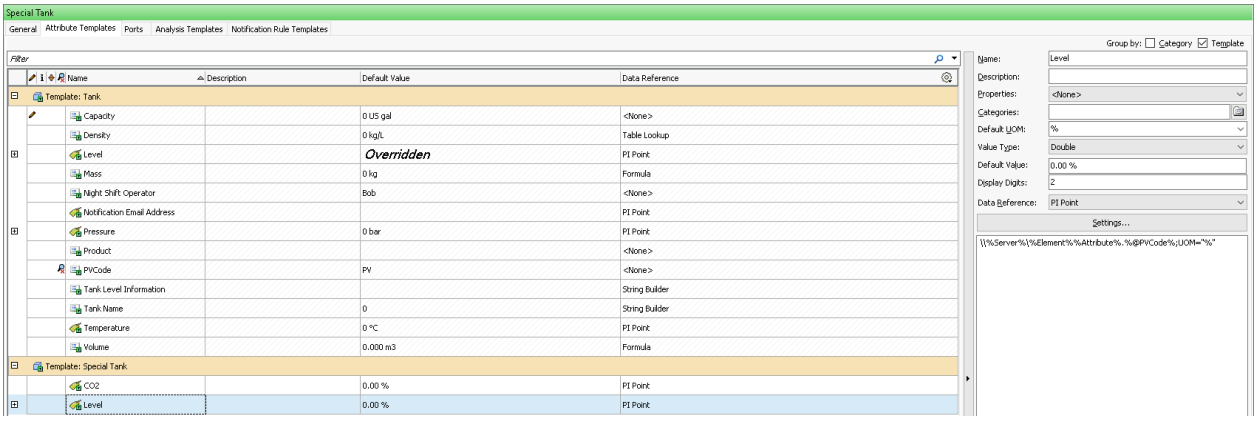
5.2. Tips for Dealing with Differences in Template-based Assets

A tip for working with derived templates:

The standard view of the Attribute Template list of derived templates is only showing the attributes which are special to the derived template and not those of the base template. We can use the Group by Template option also in the Library to see all attributes an element from this template will get.



This view is also useful when attributes from the base template have a different definition in the derived template. In the example below, which will be discussed in the next section, the Level attribute from the Tank template is marked as Overridden since the derived template Special Tank has a different definition for this attribute.



A note on Allow Extensions:

There is also the option to add attributes to individual template-based assets by allowing extensions. Using derived templates should always be the preferred method over Allow Extensions even if the exception only applies to a single asset. As soon as it is more than one, derived templates help to ensure consistency.

How to deal with assets which are lacking measurements or properties?

Looking at the opposite problem, one or just a few assets might lack a measurement most assets of the type have. Instead of creating a base template with all attributes but the one in question and a derived template with that additional attribute, we can also exclude attributes for individual assets. We have discussed how to apply the excluded property in the earlier section *AF Attribute Properties*.

5.3. Working with derived element templates

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Working with derived element templates

Database: **Velocity Terminals**

Activity Objectives (Working with Velocity Terminals)

- Create a derived template for a special tank that has an additional sensor for CO2 measurement
- Use a derived template to accommodate a different PI Point reference for the Level attribute in a special tank.

Two modifications were done on Tank10, which means that the tank deviates from the normal ones on the following details:

- There is an additional sensor for CO2 measurement. The tag name is:
  Tank10CO2.PV

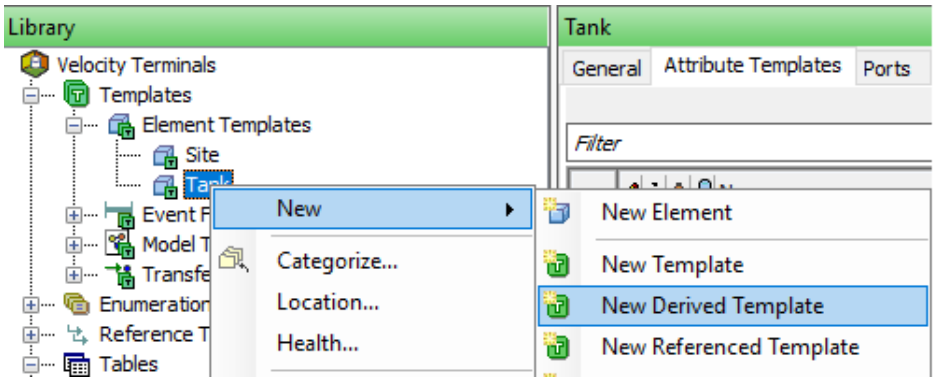
- The name for the Level sensor is different now. Instead of Tank10LI.PV the name is
  Tank10LEVEL.PV

  This means the name does not follow the normal name pattern %Element%LI.PV. Instead the name pattern has to be %Element%Level.PV (or: %Element%%Attribute%.PV).

Approach

Create an element template for a Special Tank that has an additional sensor and a different naming pattern for the Level attribute.

1. Open the Tank template in the Library of the Velocity Terminals database.
2. From the Tank template context menu and select New Derived Template



3. Rename the new element template to Special Tank.
4. On the Attributes Templates tab, add a new attribute template CO2:
5. Additional Attribute for CO2 (PI Point: **Tank10CO2.PV**)

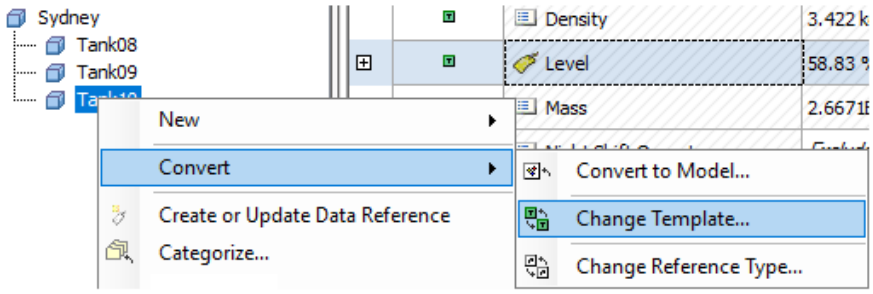
   Default UOM = percent (Ratio)
   Value Type= Double
   DisplayDigits=2
   Data Reference = PI Point

6. Click on Settings… to enter the PI Point name. Use substitution parameters to define a standard naming pattern based on the element and the attribute name.
   What do you write? %Element%%Attribute%.PV

7. Change the Source Unit from <Default> (%) to %., Select OK.

8. Check in the change, then refresh.

9. In the Navigator, switch to Elements. Select Tank10 and change the template from Tank to Special Tank.



10. Verify the Tank10 has the CO2 attribute, the value is from PI Point Tank10CO2.PV.

Override for Level

11. On the Attributes Templates tab, add a new attribute template Level (in the Special Tank template) :

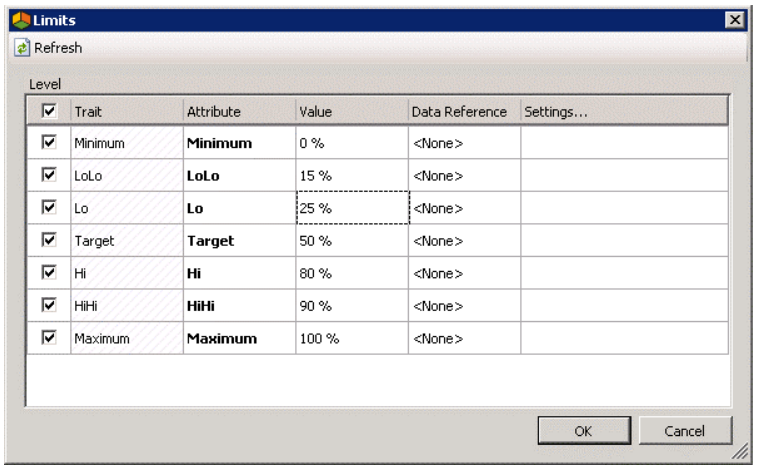
    Default UOM = percent (Ratio)
    Value Type= Double
    DisplayDigits=2
    Data Reference = PI Point

    Setting - (PI Point: **Tank10LEVEL.PV**)

12. Click on Settings… to enter the PI Point name. Use substitution parameters to define the different naming pattern based on the element and the attribute name. %Element%%Attribute%.%@PVCode%

13. Change the Source Unit from <Default> (%) to %.

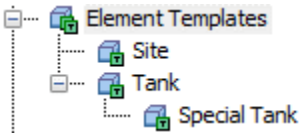
14. To define the limits, select the Level attribute and click on Limits…in the context menu. Set the limits in the same way as for the normal tank:



15. Check-In your changes and refresh.
16. In the Navigator, switch to Elements. Select Tank10 and verify the PI Point for the Level attribute of Tank10 is Tank10Level.PV

Show element templates based on inheritance

17. Open the Element Templates in the Library of the Velocity Terminals. From the context menu, select Arrange By > Arrange By Template Inheritance. Result: Special Tank is shown below Tank.



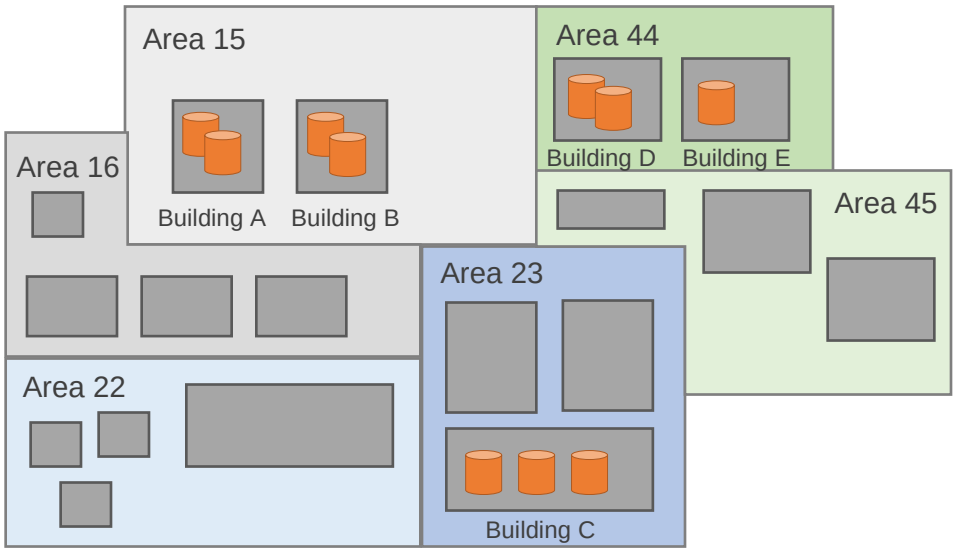
5.4. Working with Referenced Templates

Database: **Tank Farm**

When starting out with AF some people get confused with the hierarchical structure of derived templates and hierarchical structure of elements. Those are different concepts since Base Templates and Derived Templates are both dealing with elements of the same type – typically located at the same level of a hierarchy. However, there is another type of template relationship which reflects the hierarchical structure of elements: Referenced Templates.

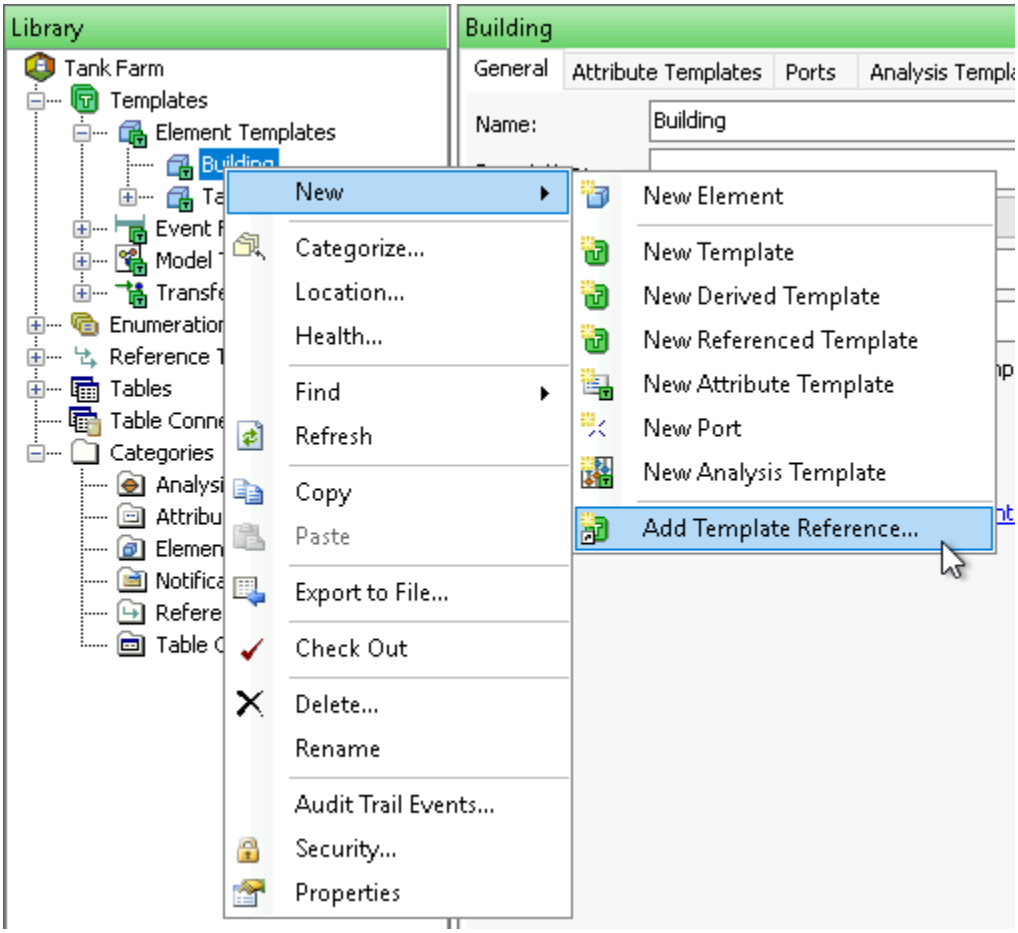
AF has the concept of referenced templates to create a parent-child relationship between templates. This does not enforce that a referenced Child Template is used for the Child elements of a referenced Parent Template, but the reference type will appear in the Choose Element Template window and this reference type will only allow referenced Child Templates to the referenced Parent Template. We will go through an example to see how this works in detail.

In the Tank Farm database we want to establish a certain structure: tank elements should be grouped under a building element which should be part of an area element. Below is a sketch of the plant map. We are only interested in the areas and buildings which contain tanks (red cylinders). Our goal is to reflect the physical structure (areas contain buildings, buildings contain tanks) in the template structure.



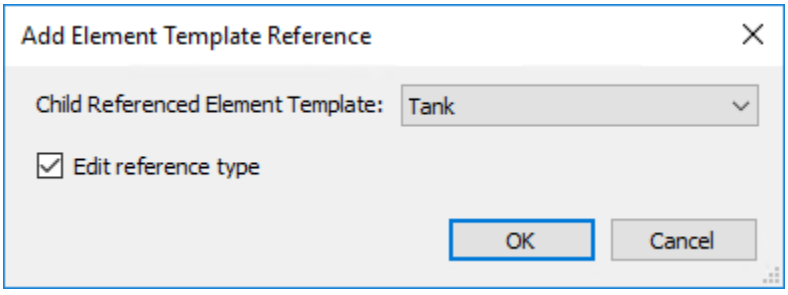
We start with assigning a relationship to the existing templates. The templates of the category Tanks (Tank and Tank with Foam Sensor) should become referenced templates of the Building template.

1. In the Library, right click on the Building template and select New > Add Template Reference…

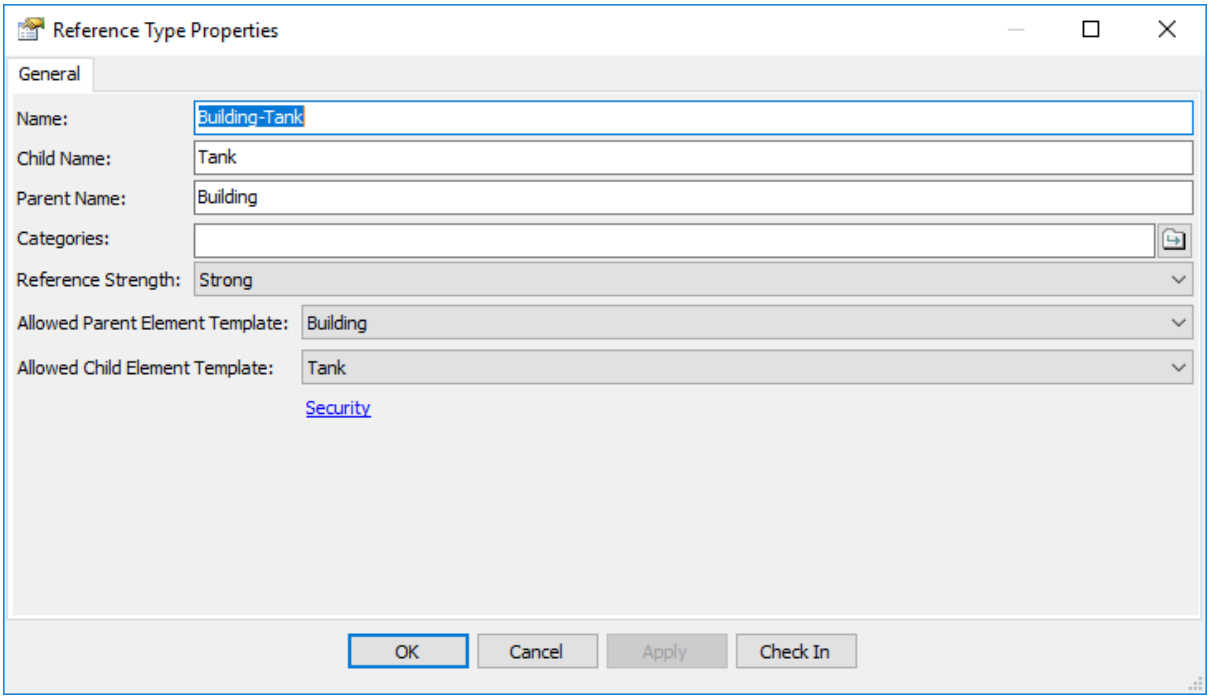


Note: In this example we establish a reference relationship between existing templates. It is also possible to create a new Child Referenced Element Template directly by using the New Referenced Template option as can be seen in the screenshot above.

2. In the Add Element Template Reference window we select Tank as the Child Referenced Element Template. The derived template Tank with Foam Sensor will be implicitly included. To view the configuration for the newly created reference type, we tick Edit reference type and click OK.

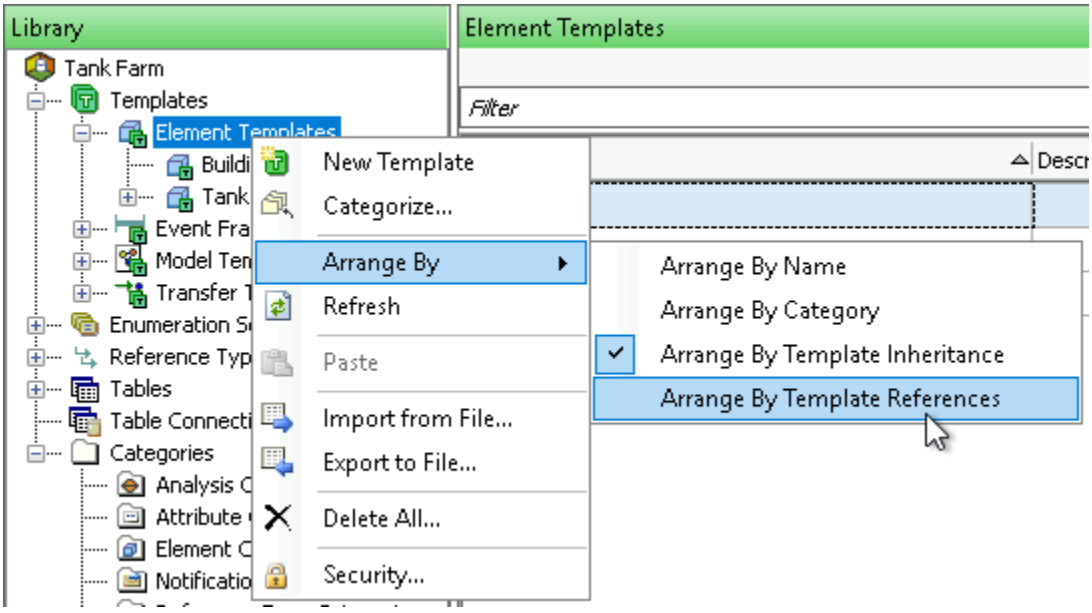


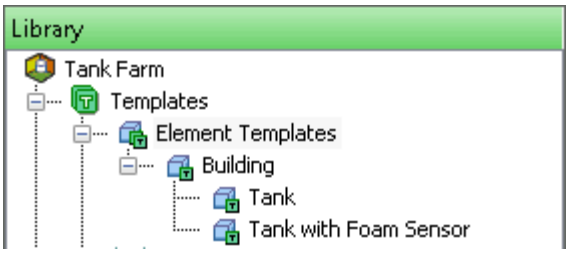
3. The Reference Type Properties window opens up.



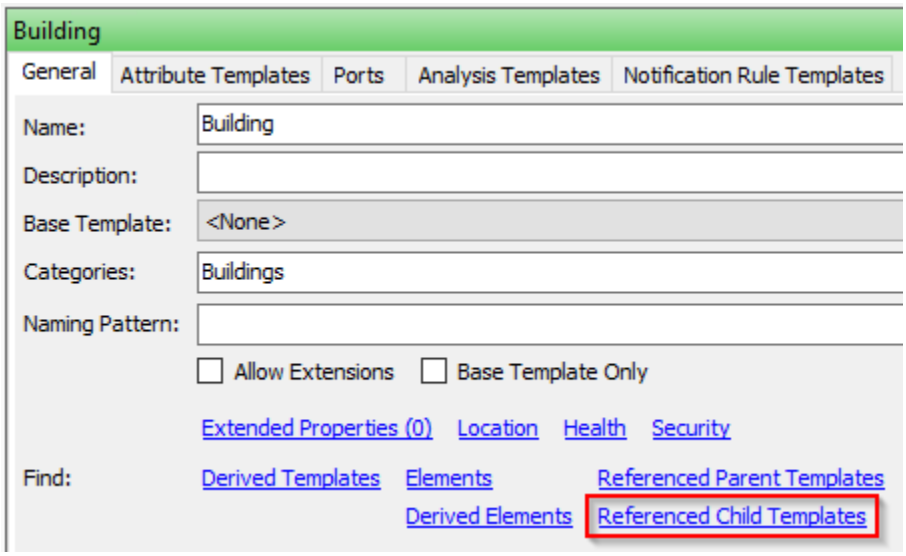
We do not need to make any modifications. Just to mention, here we could change the name of this new reference type and modify which elements are allowed as parent or child. Those settings can be also changed later via the Reference Types section of the Library. Click Check In and OK.

4. Right click on Element Templates and click Arrange By > Arrange By Template Reference to see the relationship between the templates.

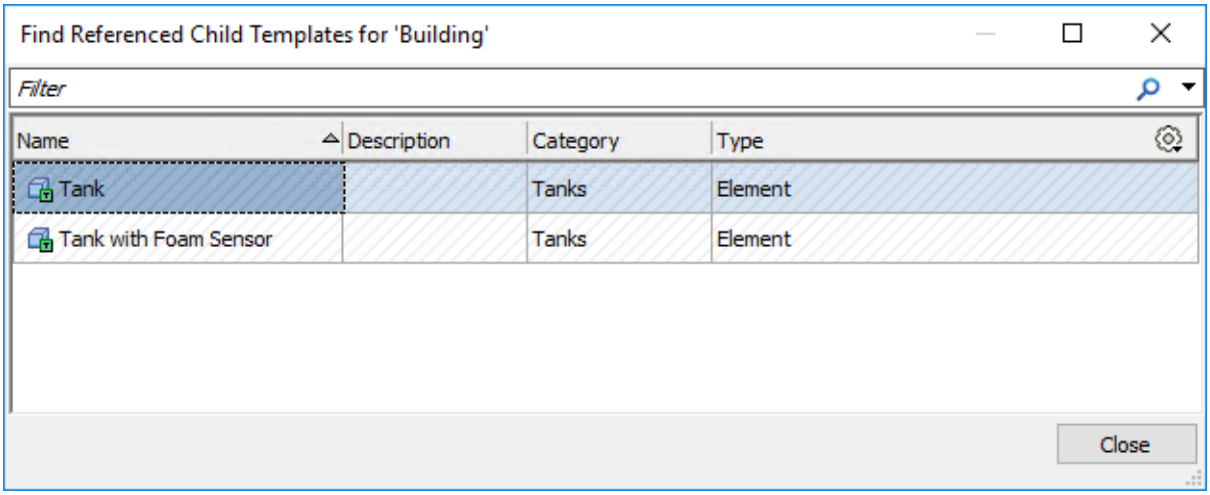




5. We can also search for the reference parent or child templates by navigating to the General tab of a template and use the Find option. For the Building template we click Find: Referenced Child Templates.

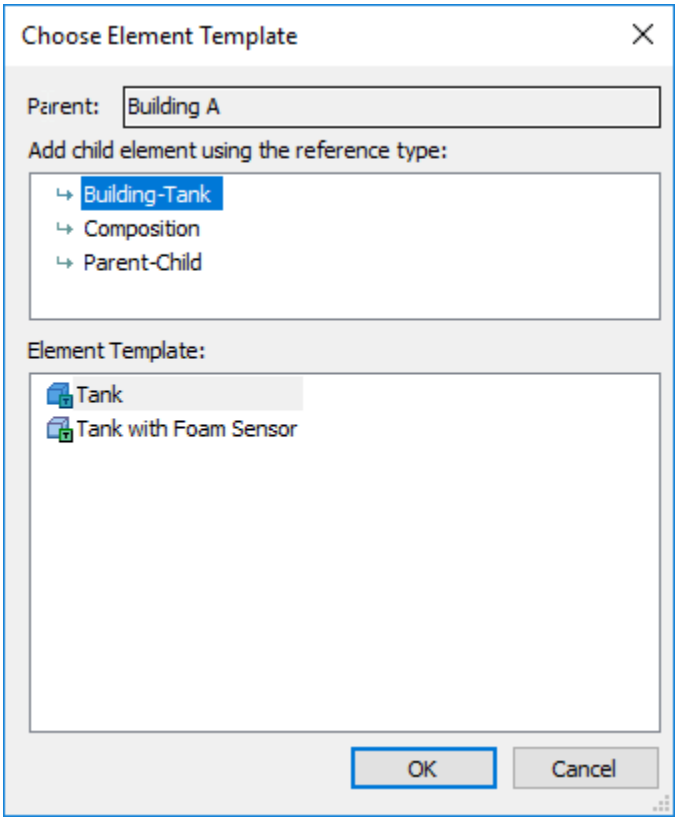


A window opens up with all referenced child templates for the Building template.

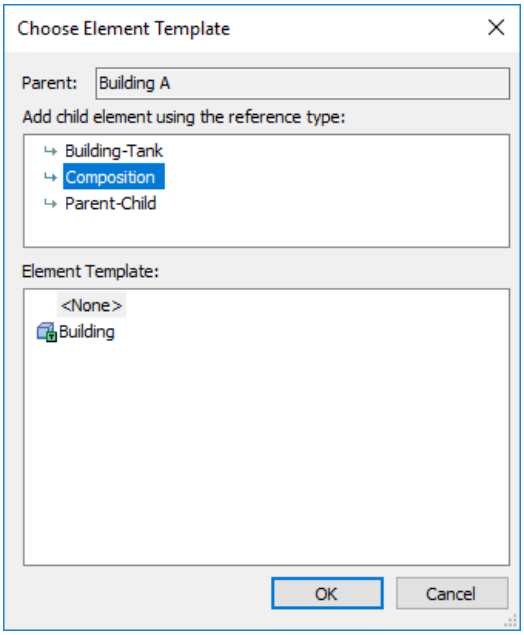


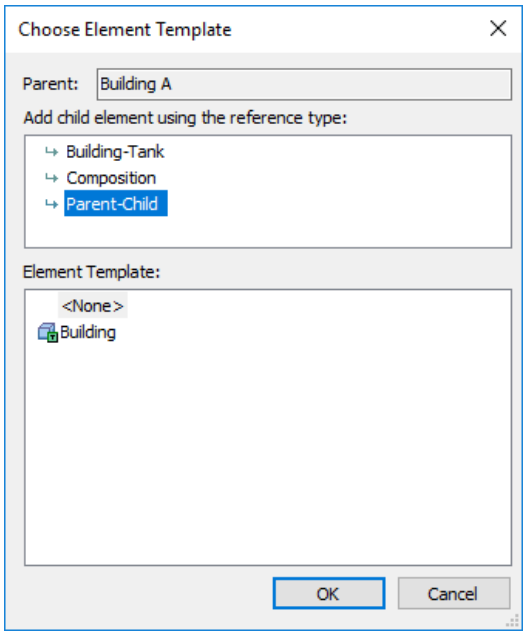
The effect of this new reference type can be seen when a new child element is created for any of the building elements. We don't need to actually create one now,

we just go through the first steps to see what choices we get for selecting an element template. When we select Building-Tank as the reference type, the only options for element templates to use are the two templates from the Tanks category.



In fact, only the Building-Tank reference type will now allow the creation of child element from the Tank or Tank with Foam Sensor templates for a Building element. The other two reference types, Composition and Parent-Child, do not allow the Tank or Tank with Foam Sensor templates for a Building element.





Why is this? Remember when we created the reference type we specified (following the default settings) the allowed parent element template and the allowed child element template.

- If we would choose <Any> for the allowed parent element template, the tank templates could be also used with other reference types as child elements of the Building template. Additionally, this reference type would be available for elements which are not from the Building template.
- If we would choose <Any> for the allowed child element template, the Building-Tank reference type would also allow other templates as children (in our case this would add the Building template to the choices).

Together, setting specific templates as the allowed types ensures that the reference type can only be used for elements from the allowed parent template and that the allowed child template can only get connected through this reference type.

In the next lesson we will create elements using the new reference type and change the reference type for the existing elements.

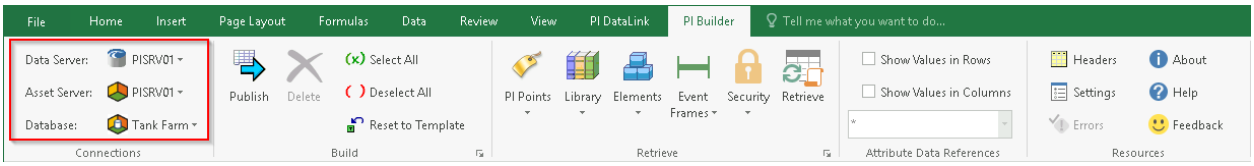
6. PI Builder Tasks

6.1. Working with PI Builder

So far, we have worked with PI System Explorer which is a great tool for AF administration work. Although we have just seen ways to perform bulk edits, there are limits in practicality of creating or editing objects in large numbers in PSE. When it comes to large scale edits PI Builder is the better suited tool. In practise, both tools will probably often be used in combination: Starting out in PSE and taking the existing structures as syntax examples for building out the rest of the database in PI Builder.

In this section we will partly following this approach as we revisit areas where we have started building out our database in PSE. For the purpose of this training, we will stick to rather small scale editing in PI Builder as well, but using the capabilities of Microsoft Excel together with the PI Builder plugin the methods we show can be easily scaled up to large numbers.

Whenever we start working with PI Builder, we have to make sure to be connected to the correct servers and database.



6.2. Build and Edit Templates and Other Library Objects

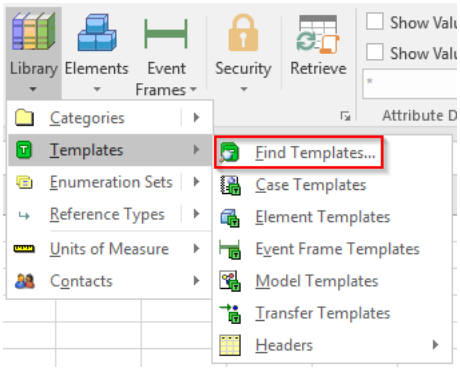
Database: **Tank Farm**

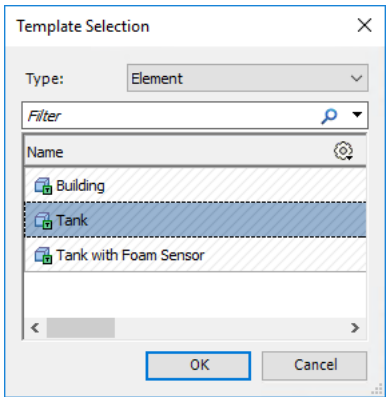
In this section we will go through some examples for applying configuration changes to Library objects with PI Builder.

Build a new attribute template for an existing element template

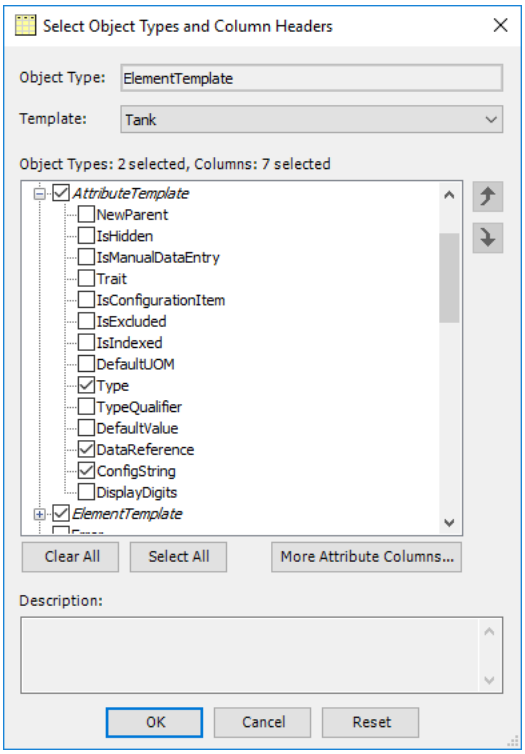
For the Tank template we want to introduce an additional attribute template to display the building letter code (A, B, etc.). This attribute will be very similar to the attribute TankID, only that we will use the name of the parent element and take just one character from the right instead of two. We can therefor use the attribute TankID as a starting point to apply modifications.

1. First, we load in the configuration of the Tank template with its attribute templates. For doing so, we select Library > Templates > Find Templates… and select the Tank template in the following window.





In the Select Object Types and Column Headers window, click Clear All. This clears all not required selections and is a good starting point in many cases when a targeted selection should be made. We then add some columns from the AttributeTemplate category: Type, DataReference and ConfigString and click OK.



2. In the Retrieve Selected Objects window, click Close once the line “The requested action is complete.” appears. We are getting the Tank template and all its attribute templates loaded into the spreadsheet.

	A	B	C	D	E	F	G
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**AttributeType**	**AttributeDataReference**	**AttributeConfigString**
2	x		Tank	ElementTemplate			
3	x	Tank	Capacity	AttributeTemplate	Double		
4	x	Tank	Density	AttributeTemplate	Double	Table Lookup	SELECT Density FROM [Material Properties] WHERE MaterialID = @Product
5	x	Tank	Level	AttributeTemplate	Double	PI Point	\\%Server%\%Element%LI.PV
6	x	Tank	Level|Hi	AttributeTemplate	Double		
7	x	Tank	Level|HiHi	AttributeTemplate	Double		
8	x	Tank	Level|Level.2HoursAgo	AttributeTemplate	Double	PI Point	|Level;RelativeTime=-2h
9	x	Tank	Level|Level.2HoursAverage	AttributeTemplate	Double	PI Point	|Level;TimeMethod=TimeRangeOverride;RelativeTime=-2h;TimeRangeM
10	x	Tank	Level|Lo	AttributeTemplate	Double		
11	x	Tank	Level|LoLo	AttributeTemplate	Double		
12	x	Tank	Level|Maximum	AttributeTemplate	Double		
13	x	Tank	Level|Minimum	AttributeTemplate	Double		
14	x	Tank	Level|Target	AttributeTemplate	Double		
15	x	Tank	Mixer Status	AttributeTemplate	Double	PI Point	\\PISRV01\%Element%.Status;ptclassname=classic;pointtype=Digital;com
16	x	Tank	Mixing Speed	AttributeTemplate	Double	PI Point	\\PISRV01\%Element%.MixingSpeed;UOM=rpm;ptclassname=classic;poin
17	x	Tank	Night Shift Operator	AttributeTemplate	String		
18	x	Tank	Product	AttributeTemplate	AFEnumerationValue		
19	x	Tank	TankID	AttributeTemplate	String	String Builder	Right(%Element%,2);

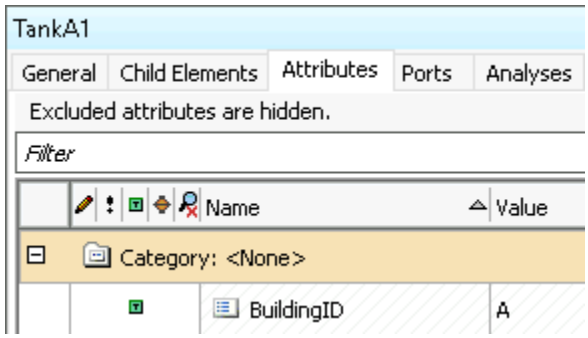
3. We copy the row with the attribute TankID and name the new attribute template BuildingID. We also need to change the AttributeConfigString to Right(%..\Element%,1);

4. We deselect all rows apart from the new attribute template (delete the x in the Selected(x) column or click Deselect All and place an x in the row with BuildingID).

	A	B	C	D	E	F	
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**AttributeType**	**AttributeDataReference**	**AttributeConfigString**
2			Tank	ElementTemplate			
3		Tank	Capacity	AttributeTemplate	Double		
4		Tank	Density	AttributeTemplate	Double	Table Lookup	SELECT Density FROM [Ma
5		Tank	Level	AttributeTemplate	Double	PI Point	\\%Server%\%Element%I
6		Tank	Level|Hi	AttributeTemplate	Double		
7		Tank	Level|HiHi	AttributeTemplate	Double		
8		Tank	Level|Level.2HoursAgo	AttributeTemplate	Double	PI Point	|Level;RelativeTime=-2h
9		Tank	Level|Level.2HoursAverage	AttributeTemplate	Double	PI Point	|Level;TimeMethod=Tim
10		Tank	Level|Lo	AttributeTemplate	Double		
11		Tank	Level|LoLo	AttributeTemplate	Double		
12		Tank	Level|Maximum	AttributeTemplate	Double		
13		Tank	Level|Minimum	AttributeTemplate	Double		
14		Tank	Level|Target	AttributeTemplate	Double		
15		Tank	Mixer Status	AttributeTemplate	Double	PI Point	\\PISRV01\%Element%.St
16		Tank	Mixing Speed	AttributeTemplate	Double	PI Point	\\PISRV01\%Element%.M
17		Tank	Night Shift Operator	AttributeTemplate	String		
18		Tank	Product	AttributeTemplate	AFEnumerationValue		
19		Tank	TankID	AttributeTemplate	String	String Builder	Right(%Element%,2);
20	x	Tank	BuildingID	AttributeTemplate	String	String Builder	Right(%..\Element%,1);

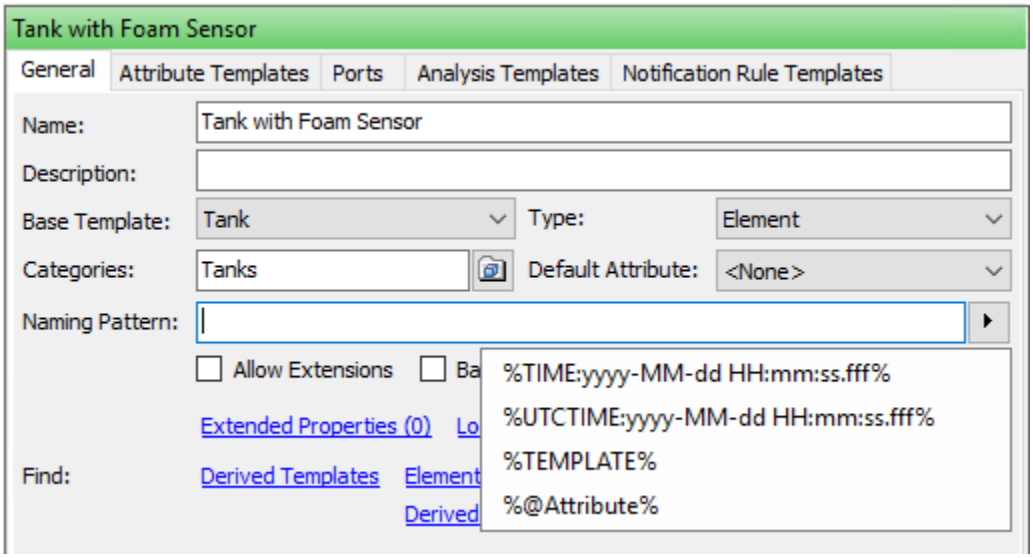
5. Click Publish. Select Edit Mode: Create Only.

6. We can confirm in PSE that the tanks got the new attribute and that it is working correctly.



Edit an element template to change a property: add a naming pattern

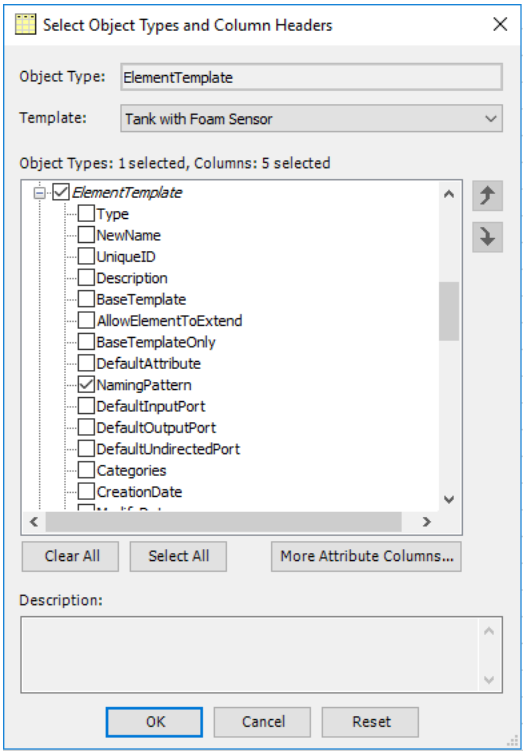
When new elements are created from a template the default naming pattern uses the template name and adds a number. In the case of the template Tank with Foam Sensor we might want to change this to also simply use Tank. We have just created a new attribute BuildingID which we can use in the naming pattern to include the correct Building ID (A, B, etc.). Let us have a look at the syntax options for the naming pattern in PSE:



In our case we will use: Tank%@BuildingID%.

Note: This syntax will always try to create elements with a fixed name (e.g. TankC for child elements of Building C) unless an element with this name already exists. In our example we would have to assign the tank number manually. We could automate this by introducing another attribute which increments the number (for example a counter of the child elements of the Building).

1. On a new spreadsheet we go to Library > Templates > Find Templates… and select Tank with Foam Sensor.
2. In the Select Object Types and Column Headers window we reset the selection by clicking Clear All. Since this time we want to make changes to the property of the element template itself we open the category ElementTemplate, tick NamingPattern and click OK.



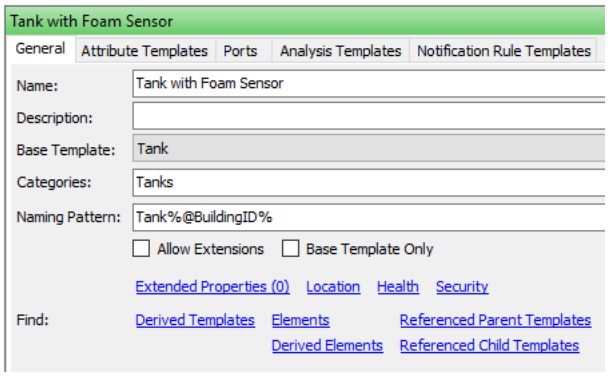
3. In the column NamingPattern we enter for Tank with Foam Sensor Tank%@BuildingID%.

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**NamingPattern**
2	x		Tank	ElementTemplate	
3	x		Tank with Foam Sensor	ElementTemplate	Tank%@BuildingID%

Note: If we want to apply this also to elements created from the Tank template we would have to set this for both templates.

4. We select only the row for Tank with Foam Sensor, click the Publish button and choose Edit Only.

5. We can verify the change in PSE (click Refresh if PSE is still open):

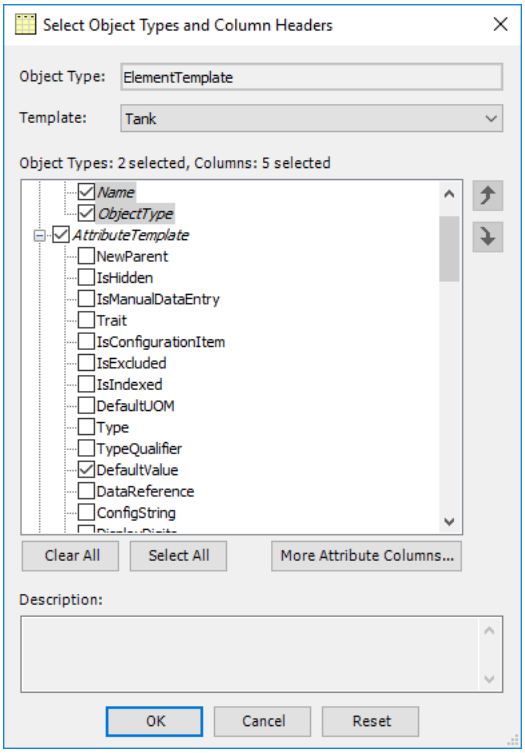


We will test this new setting at the end of this section.

Edit an element template to change/set the default value of an attribute

We want to change the Tank template to provide HC1500 as default value for the attribute Product.

1. On a new spreadsheet, go to Library > Templates > Find Templates…, select Tank and confirm with OK.
2. Click Clear All. We want to set the Default Value of an attribute, so we tick DefaultValue in the category AttributeTemplate and click OK.



3. We enter HC1500 in the column AttributeDefaultValue for the attribute Product and deselect all other rows.

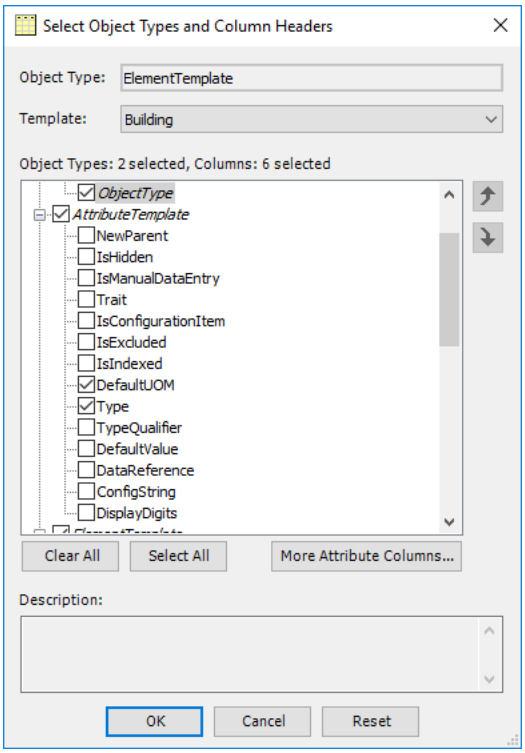
	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**AttributeDefaultValue**
2			Tank	ElementTemplate	
3		Tank	BuildingID	AttributeTemplate	
4		Tank	Capacity	AttributeTemplate	0
5		Tank	Density	AttributeTemplate	0
6		Tank	Level	AttributeTemplate	0
7		Tank	Level|Hi	AttributeTemplate	80
8		Tank	Level|HiHi	AttributeTemplate	90
9		Tank	Level|Level.2HoursAgo	AttributeTemplate	0
10		Tank	Level|Level.2HoursAverage	AttributeTemplate	0
11		Tank	Level|Lo	AttributeTemplate	25
12		Tank	Level|LoLo	AttributeTemplate	15
13		Tank	Level|Maximum	AttributeTemplate	100
14		Tank	Level|Minimum	AttributeTemplate	0
15		Tank	Level|Target	AttributeTemplate	50
16		Tank	Mixer Status	AttributeTemplate	0
17		Tank	Mixing Speed	AttributeTemplate	0
18		Tank	Night Shift Operator	AttributeTemplate	Bob
19	x	Tank	Product	AttributeTemplate	HC1500
20		Tank	TankID	AttributeTemplate	0

4. Publish with Edit Mode: Edit Only.
5. We can verify in PSE that this action had no influence on the Product values of the elements. The only thing that was changed is the default value in the attribute template.

Build a new template

Build a template for the area elements. This template will get an attribute which gives the size of the area.

1. To have an example for the correct format we can load the configuration of the Building template. We go to Library > Templates > Find Templates…, select the Building template and click OK.
2. For the column selection we clear all and then select the properties from the category AttributeTemplate which will be different from the default settings for the attribute Size. We will create an attribute of with no Data Reference type (<None>). For this we will only have to set the DefaultUOM and the Type (value type).



3. We get the following:

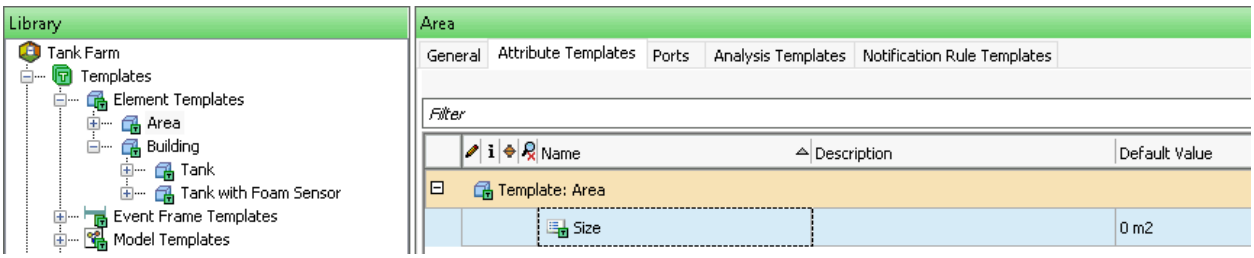
	A	B	C	D	E	F
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**AttributeDefaultUOM**	**AttributeType**
2	x		Building	ElementTemplate		
3	x	Building	AreaID	AttributeTemplate		Int16

Here we replace Building with Area (you can use the Replace function of Excel) and AreaID with Size. In row 3, we set the AttributeDefaultUOM to m2 and the AttributeType to Single.

	A	B	C	D	E	F
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**AttributeDefaultUOM**	**AttributeType**
2	x		Area	ElementTemplate		
3	x	Area	Size	AttributeTemplate	m2	Single

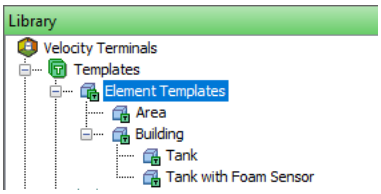
4. Make sure both rows are selected, click the Publish button and choose Create Only as the Edit Mode.

5. We can verify again in PSE that the new element template was created together with the Size attribute template.



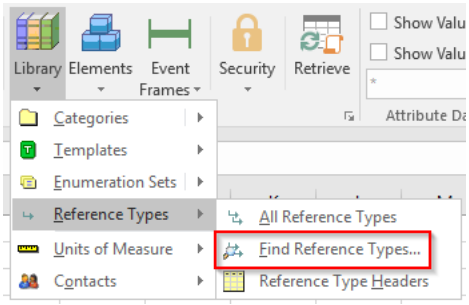
Build a new reference type

Remember that we wanted to establish an element hierarchy through parent-child relationships between the templates. When we look into the Library and arrange the element templates by template references, we see that the Area template is not yet placed into a relationship with the Building and Tank templates.



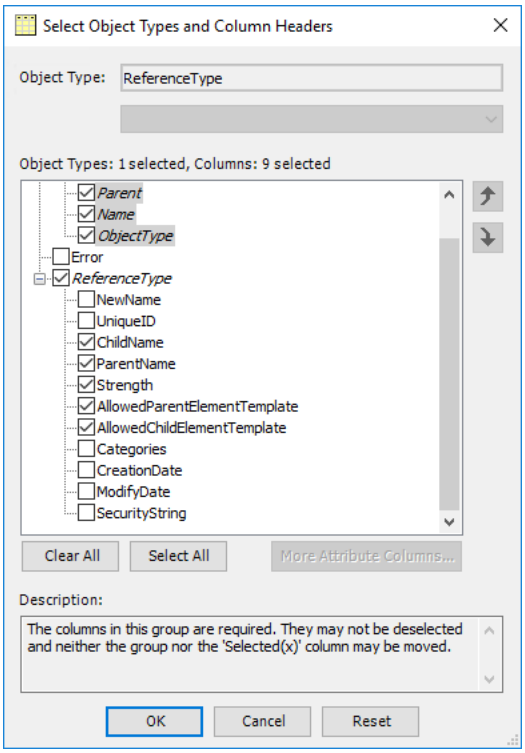
To achieve this, we build a new reference type Area-Building. The configuration of the reference type is the only thing needed to set this relationship. We do not need to make any configuration change on the template(s) involved.

1. Again we load some existing configuration to have some guidance. We go to Library > Reference Types > Find Reference Types…



   From the Reference Type Selection window we choose the Building-Tank reference type since we want to build the new type analogously.

2. Click Clear All and select from the ReferenceType category: ChildName, ParentName, Strength, AllowedParentElementTemplate, AllowedChildElementTemplate.



3. We get the following result:

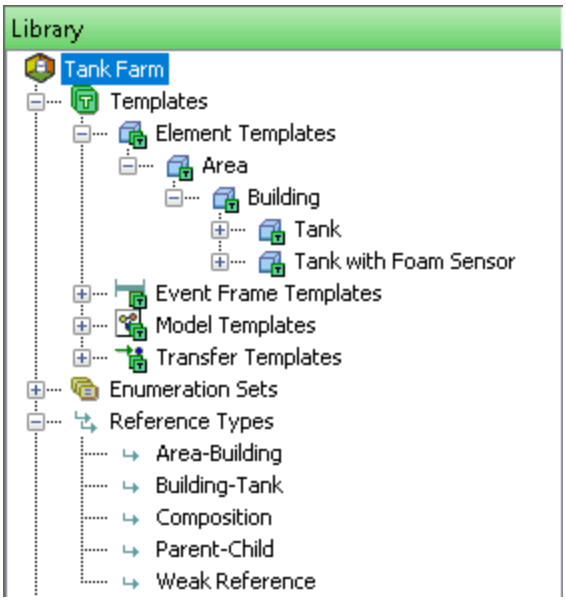
	A	B	C	D	E	F	G	H	I
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**ChildName**	**ParentName**	**Strength**	**AllowedParentElementTemplate**	**AllowedChildElementTemplate**
2	x		Building-Tank	ReferenceType	Tank	Building	Strong	Building	Tank

We can use the Replace function of Excel (Home ribbon > Find & Select > Replace…) to replace Building with Area and afterwards Tank with Building.

	A	B	C	D	E	F	G	H	I
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**ChildName**	**ParentName**	**Strength**	**AllowedParentElementTemplate**	**AllowedChildElementTemplate**
2	x		Area-Building	ReferenceType	Building	Area	Strong	Area	Building

4. Publish with Edit Mode: Create Only.

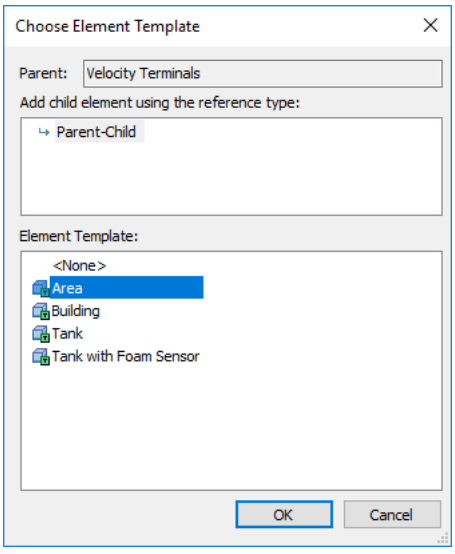
5. Checking in PSE we see that the new reference type was created. This might not be reflected in how the element templates are arranged. If this is the case you might need to close and reopen PSE.



Testing out what we have just built

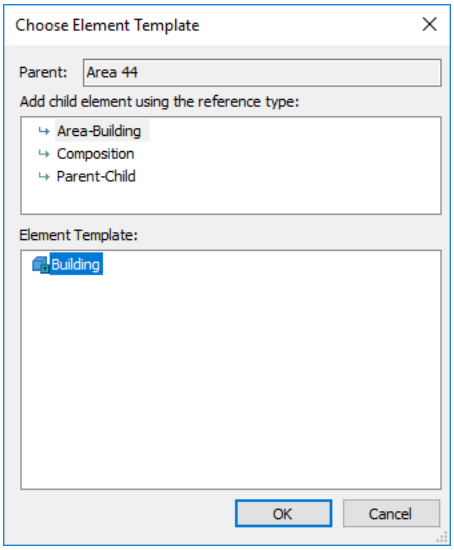
Let's take a moment to test if the newly built template and reference type are working as expected. In PI System Explorer, we will create an Area element Area 44 together with a child element Building D. Building D should get one child element for now: TankD1.

1. Right click on Elements and select New Element. Select the Area template and click OK.



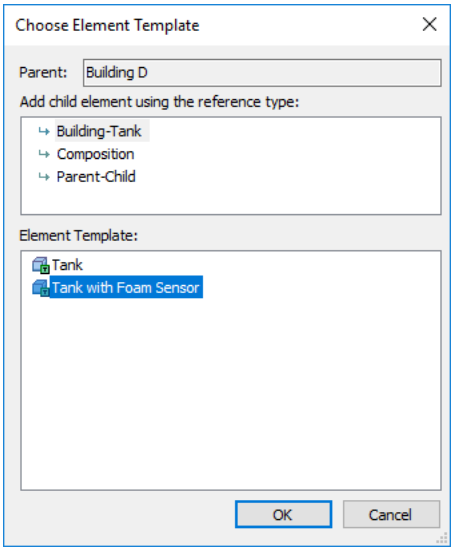
2. Rename the new area element by pressing F2 or changing the name on the General tab to Area 44. Navigate to the Attributes tab and assign a value of 15,000 square meters to the attribute Size.

3. Right click on Area 44 and select New > New Child Element. In the Choose Element Template window, we get the new reference type Area-Building as an option. Selecting it, we can only create child elements using the Building template, which is what we had intended. Click OK.



4. Rename the new building element to Building D.

5. Create a child element for Building D by right clicking and selecting New > New Child Element. Select the reference type Building-Tank which we defined earlier. Now, we can only choose between the two tank templates. We select Tank with Foam Sensor and click OK.



6. The newly created tank element automatically got the name TankD following the naming pattern we have set. Rename to set to TankD1. Check in the changes.

7. Check the attributes of TankD1. Product shows a value of HC1500 which we have set as the default value and the newly created attribute BuildingID picked up the last character from the parent element and is showing “D”

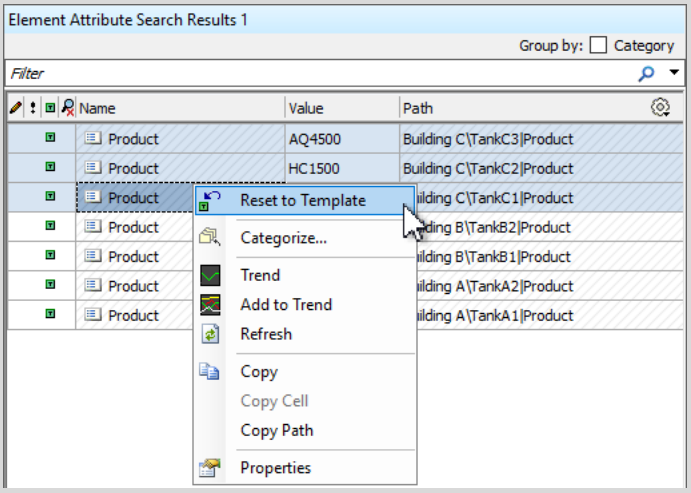
6.3. Resetting Attributes to Template in Bulk

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Resetting Attributes to Template in Bulk

Database: **Velocity Terminals – To RESET**

In this video we are going to revert attributes back to their template using PI Builder. In PSE there is an option to Create and Update Data References for all Elements. We can reset single attributes back to template by right clicking on an attribute and resetting that to template. In PI Builder we can accomplish the bulk reset.

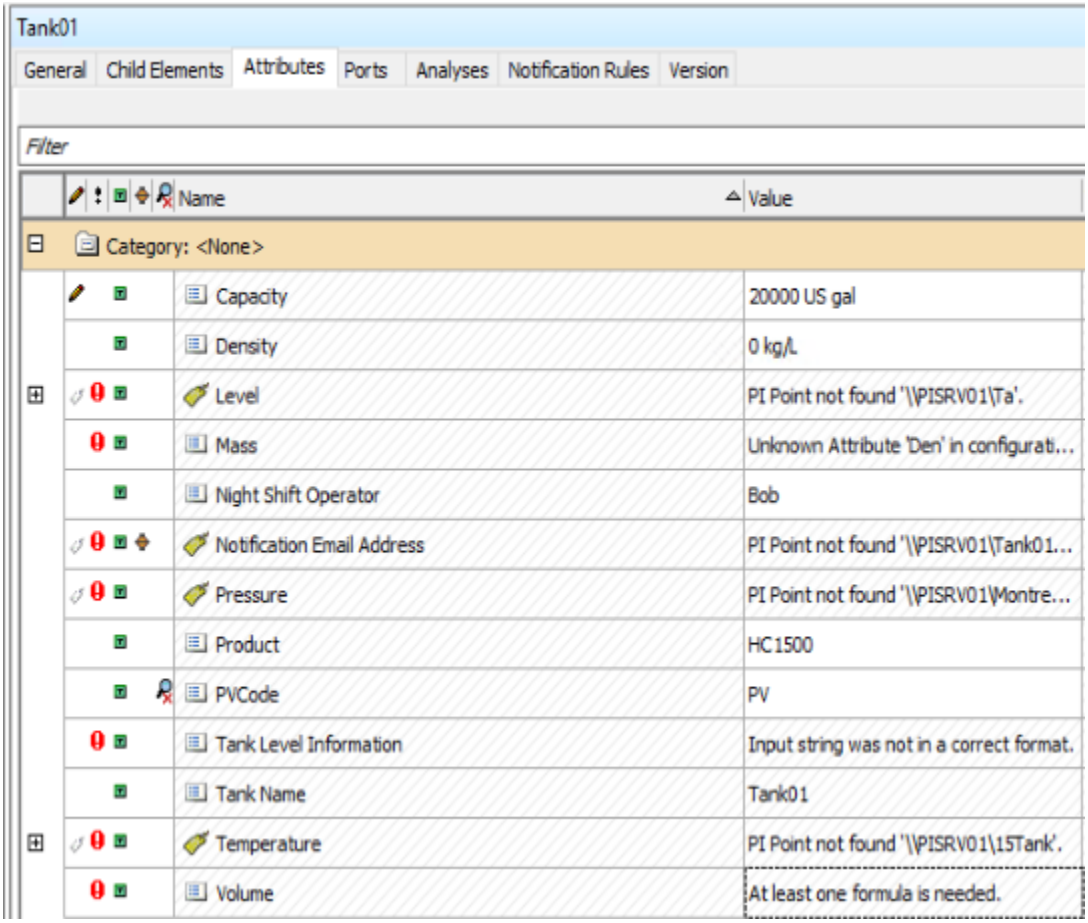
Note: In PSE it is also possible to perform multi-select editing if you know how to approach it: To select multiple objects at once to edit, they need to be displayed together in the Viewer pane. For attributes this can be achieved via Search > Attribute Search. In the below example, we searched for the attribute Product in the Tank Farm database. We can then reset multiple attributes to template by selecting them (using Strg or Shift key) and using the right click menu on one of the selected items.



Exercise (Use the Velocity Terminals To RESET database):

This section will be an exercise where you notice that many of your attributes have been altered away from their template. Your goal is to revert them back to template by using PI Builder. Notice that many of the attributes under Tank01 for Montreal

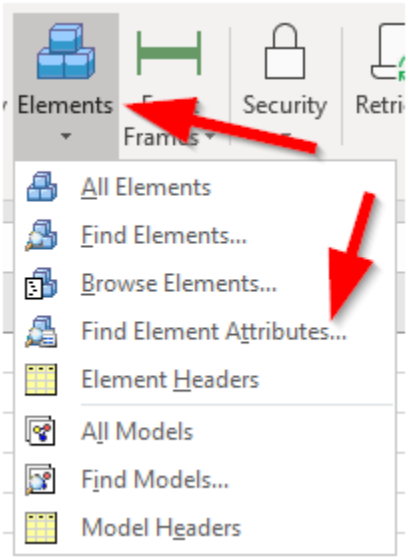
are showing errors since the configuration strings have deviated away from the template definition.



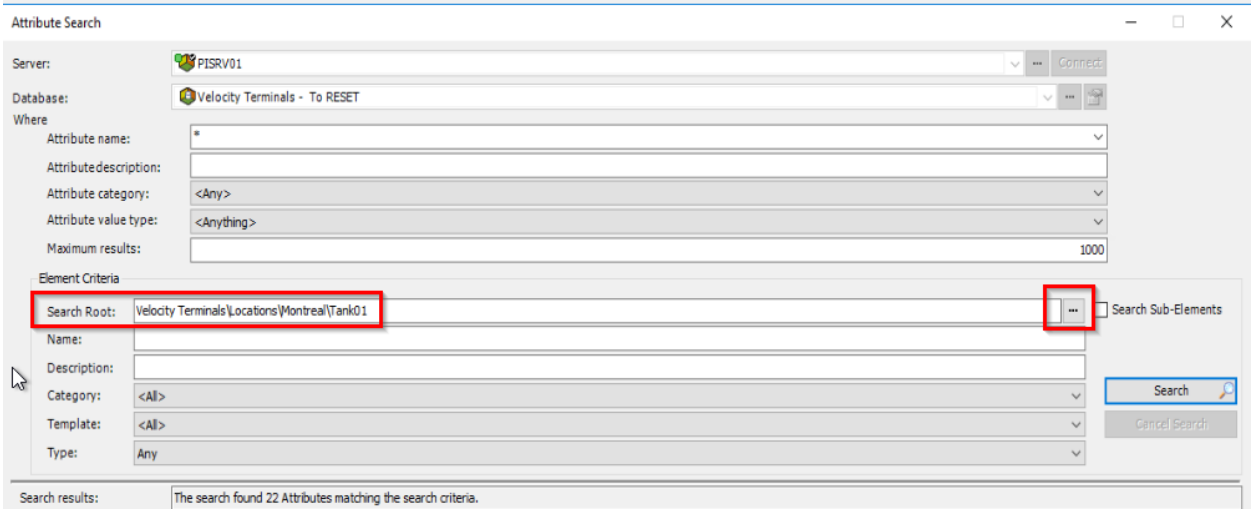
**Note the Product Value of HC1500.

Step-by-step:

1. Open Excel and load the PI Builder Add In.
2. Be sure to set the database to Velocity Terminals – To RESET.
3. Select "Elements" and "Find Element Attributes".

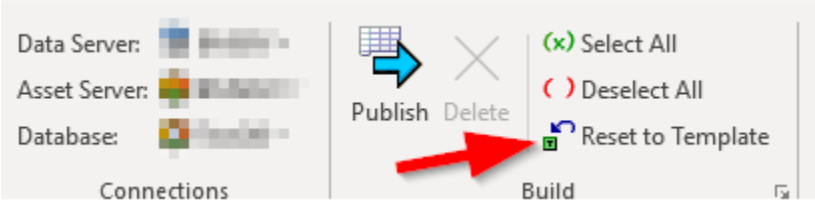


3. Enter any search criteria you need so that you search for the relevant attributes. If you would like to load all attributes leave everything blank and select "OK". For this case, we'll use the Tank01 element under Montreal as our search root and then click search:



4. Ensure that "Value" and "Data Reference" under Attribute are selected in the next window, then select "OK".

5. Within the Excel workbook press Control + A to select all items in the workbook, then select the button "Reset to Template".



6. Now select "Publish"
7. In the Publish Options box make sure that the Edit Mode is set to "Edit Only".
8. Navigate back to PI System Explorer → Elements → Velocity Terminals TO Reset → Locations → Montreal → Tank01 and verify the attributes have been reset. A Refresh may be necessary.
9. Note that the Density and Mass values have not updated. In resetting the template, we reset the Product to NULL. Enter/Select HC1500 for the Product and refresh the screen. The Density and the Mass should update.

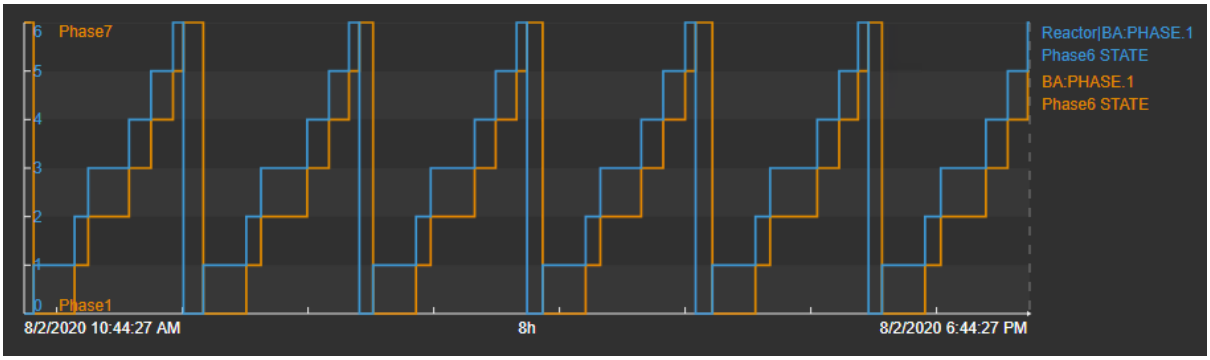
Note: Use caution when using reset to template. This example is one situation to be aware of when resetting to template. As there was no Default value in the product field, it is apparent the field needs to be updated. If we had a default value, how long would it have taken to determine a problem?

6.4. Working with Digital Tags in AF

Database: **Tank Farm**

When creating attributes which are referring to PI Points with a Point Type digital, the Value Type can be set to digital to use the digital state set defined on the Data Archive to display the string values. However, this method has the disadvantage that in contrast to the order of digital states we have on the Data Archive, AF would not be aware of an order and the string values might be displayed differently than expected when trending the attribute.

Let's look at an example. In the trend below, the orange line is from the Data Archive tag directly, while the blue line is the representation of the same tag via an AF attribute with Value Type String.



The difference in the two trends is coming from a different association of the states ("Phases" in this example) with numbers. While for the tag Phase1 is associated with 0 on the Data Archive, the corresponding string arbitrarily gets associated with 1 in AF.

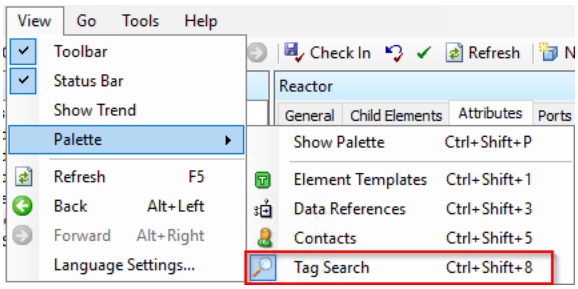
To avoid this issue the recommended method is to create an Enumeration Set in AF from the Digital State Set. We discuss three ways to do this, first we will look into two methods introduced in version 2018 SP2. After this we will also show in a video how this can be achieved in a simple way also on older systems via the PI Builder.

Note that all methods discussed below work on a per database basis. If State Sets are used in multiple AF databases the according Enumeration Sets would need to be created for each separately.

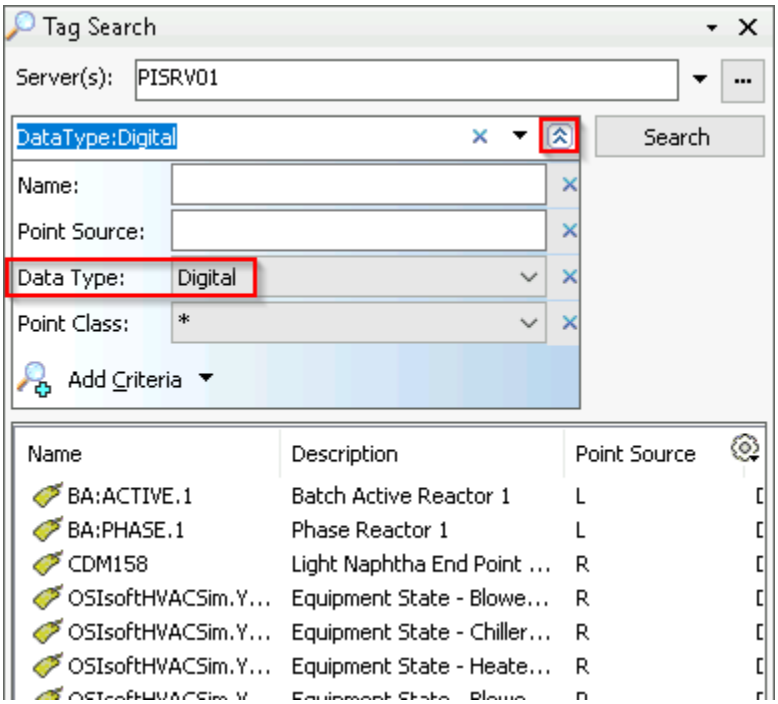
Create an Enumeration Set for a new attribute

For this example, we create a new element Reactor under the root level and choose <None> for Element Template. We use the Palette to quickly build attributes:

1. The Palette can be opened via the Menu or through key shortcuts. We will use the Tag Search, so the key shortcut is Ctrl+Shift+8.

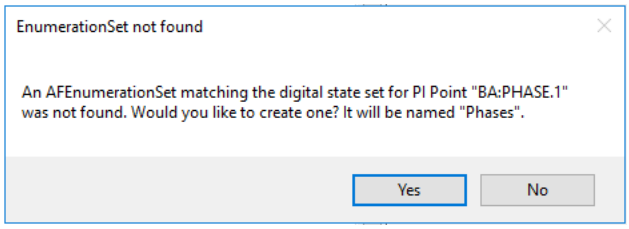


2. We want to create an attribute for a digital tag, so open up the menu by using the double arrows and select Digital as the Data Type. In this example, we will use the tag BA:PHASE.1.

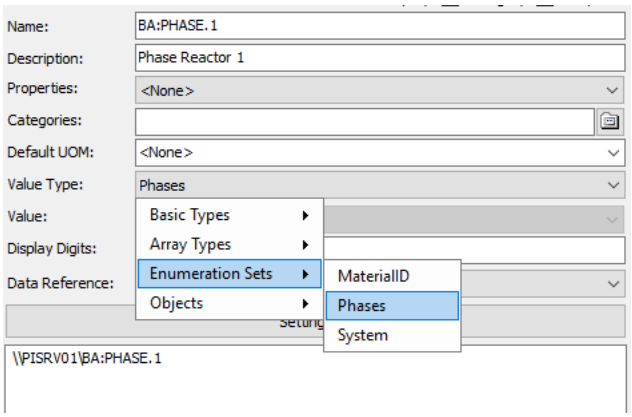


3. To create an attribute for the tag BA:PHASE.1 drag the tag name from the Tag Search Palette into the attributes pane of the Reactor element.

4. Since we do not have an Enumeration set matching the digital state set used for this tag a dialog box appears to create one. We choose Yes to create an Enumeration Set named “Phases” (same name as the Digital State Set).

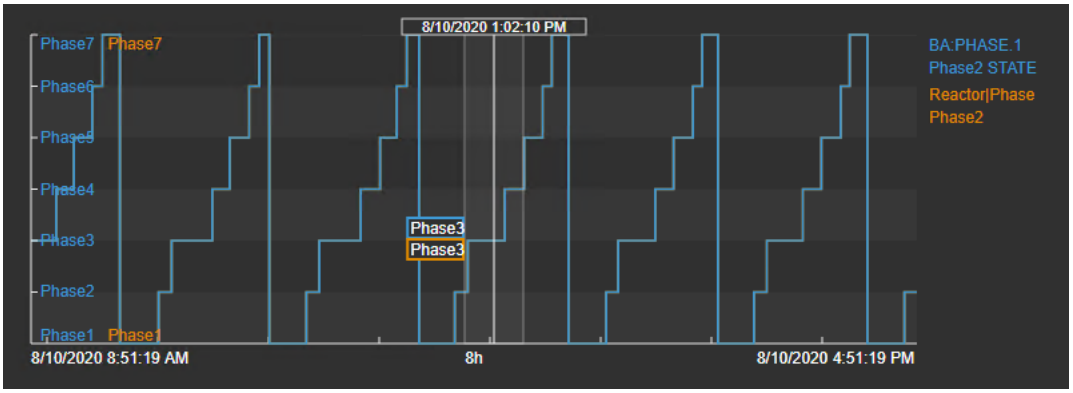


5. The new set should get automatically selected as the Value Type of the attribute.



6. The attribute can also be renamed with a friendlier name. Select the attribute and press F2 to rename or click in the name field and change BA:PHASE.1 to Phase.

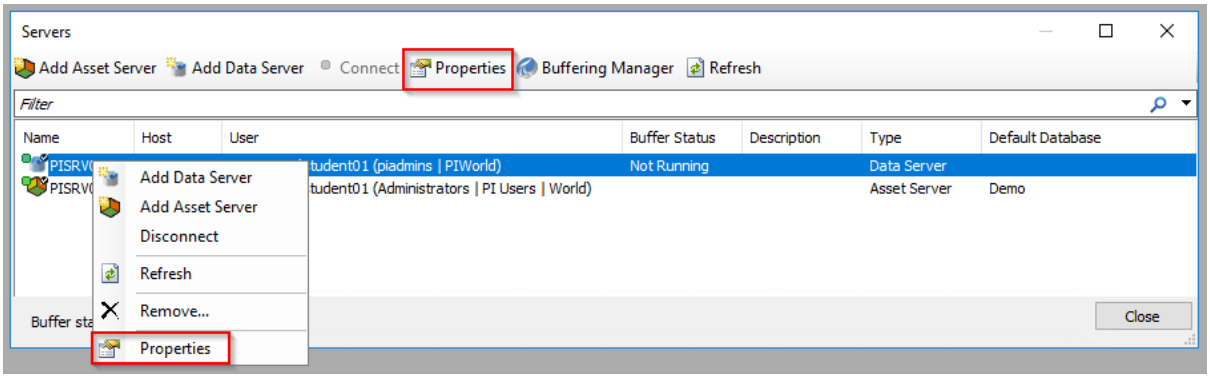
7. We can verify that this solved our trending issue by creating a trend in PI Vision. The trends of both PI tag and AF attribute are now aligned.



Create Enumeration Sets for Digital State Sets on the Data Archive Server in bulk

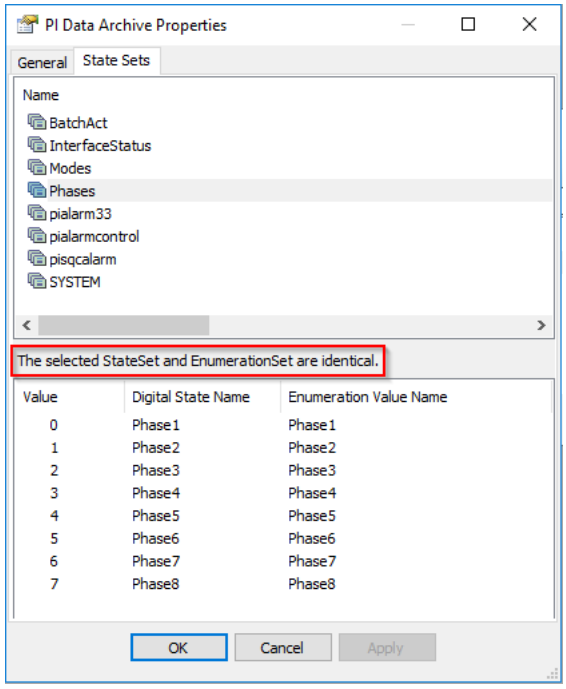
Instead of following the above method to create Enumeration Sets from the Digital State Sets as needed, we can also do this globally for all sets or for some selected sets.

1. Open the Servers window by selecting File > Connections. Right click the Data Server and select Properties or mark the Data Server and click on Properties above.

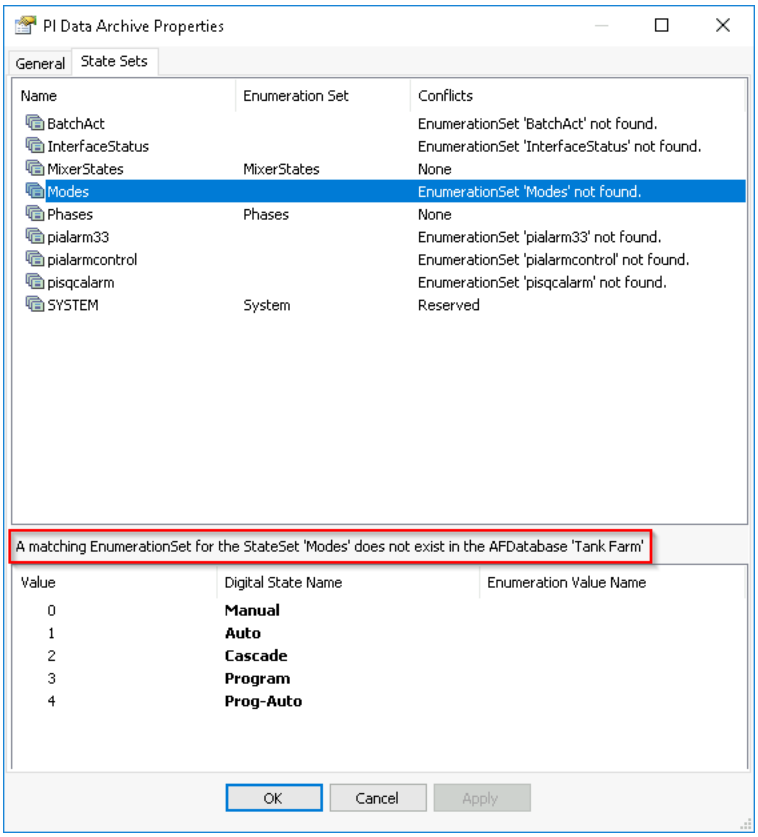


2. In the PI Data Archive Properties window open the State Sets tab. Here we can review the Digital State Sets on the Data Archive Server and if they are matched in the current AF database.

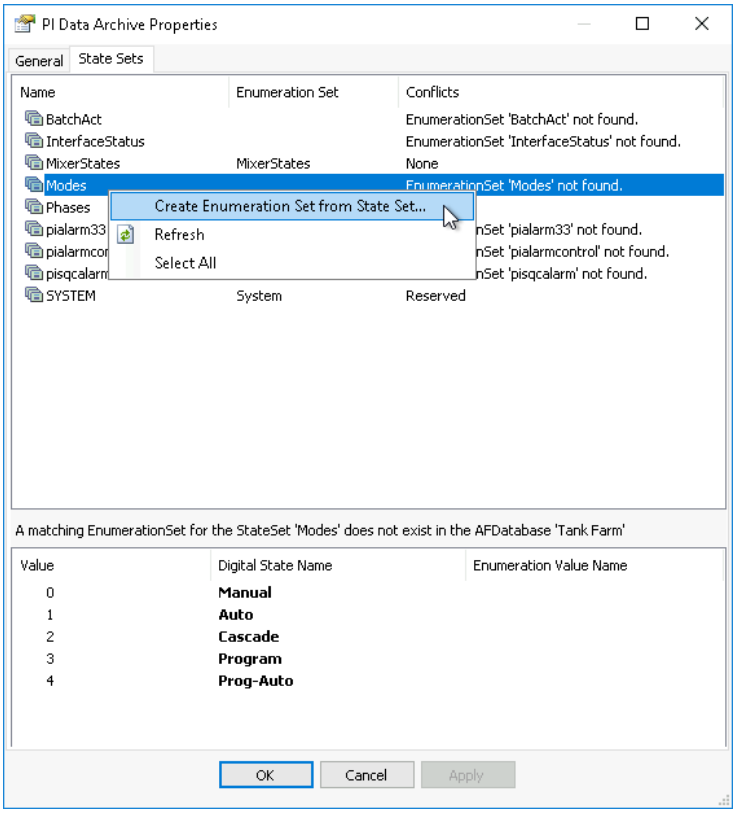
   For example, the set “Phases” which we used in the previous example now exists as a State Set and Enumeration Set and we get a message displayed that both are identical:



   If we look at a set which has no correspondence in the current database yet, e.g. Modes, we see the following:

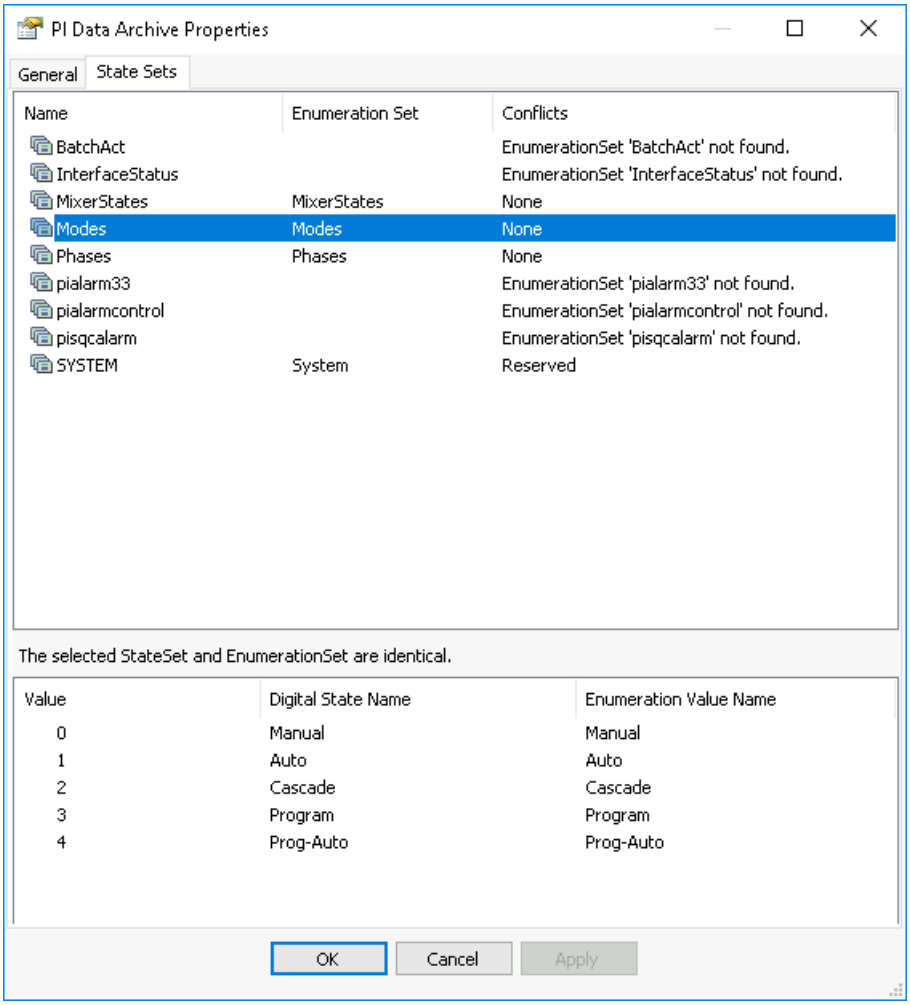


3. We can right click on the set name and click Create Enumeration Set from State Set… to create a match for the database.



Note: At this step, multiple or all State Sets can be selected to create Enumeration Sets in bulk.

4. Click Yes on the dialog window to confirm this action. The PI Data Archive Properties window now shows that the Enumeration Set Modes has been created.



For full information refer to "*Review digital state sets on a PI Data Archive*" in the OSIsoft Documentation.

6.5. Moving Digital States to Enumeration Sets

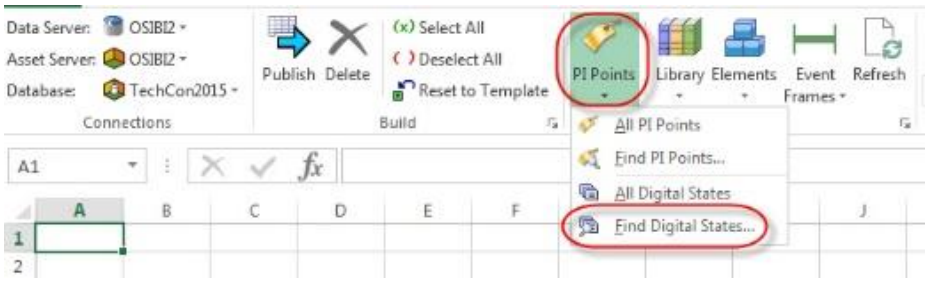
Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Moving Digital States to Enumeration Sets

Database: **Velocity Terminals**

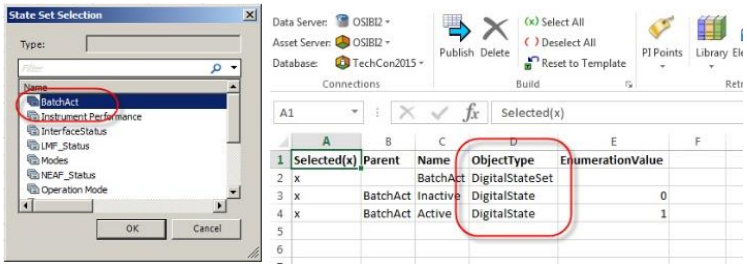
Create Enumeration Sets in PI AF from the Digital States you have created in the PI Archive.

Step by step:

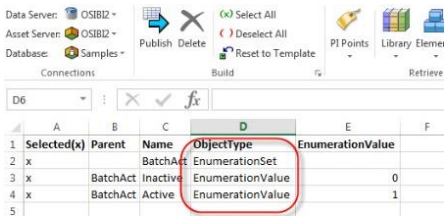
a. Create a new workbook. In the ribbon click the down arrow under the **PI Points** and select **Find Digital States**.



b. Select **BatchAct** and click the **OK** button.

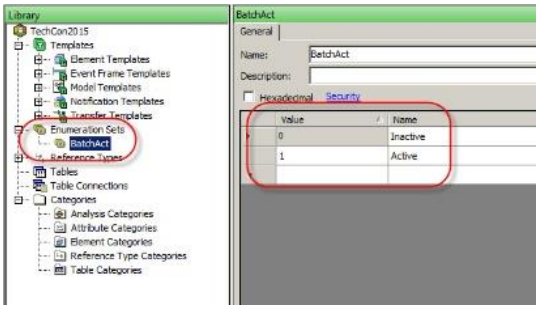


c. Under the **ObjectType** property change the **DigitalStateSet** to **EnumerationSet**, and **DigitalState** to **EnumerationValue**.



d. Click the Publish in the ribbon, select Create Only, then click the **OK** button on the next dialog, and the **Close** button on the last dialog that pops up.

e. Open the **PI System Explorer** if it is not already open, and press the **Ctrl+3** key combination to navigate to the **Library** view. Expand the **Enumeration Sets** in the **Browser** pane and you will see the **BatchAct** Enumeration Set and the Enumeration values. If BatchAct does not display, perform a refresh.



6.6. Expand and Edit the Unit of Measure Database

Database: **Tank Farm**

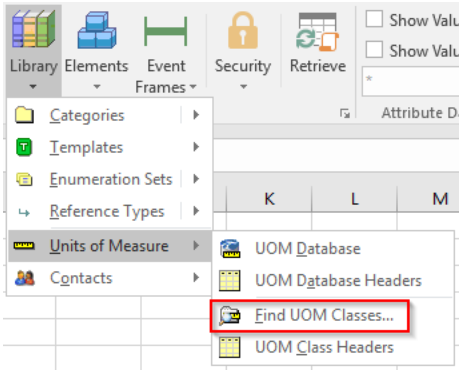
The PR team wants to communicate the size of some areas within the plant. For a press release they want to give the size in soccer fields for the European journals and in football fields for the American journals. To have a convenient way to calculate the numbers automatically, we will add the two field sizes as new units of measure.

For the purpose of this exercise, we will use the following size definitions:

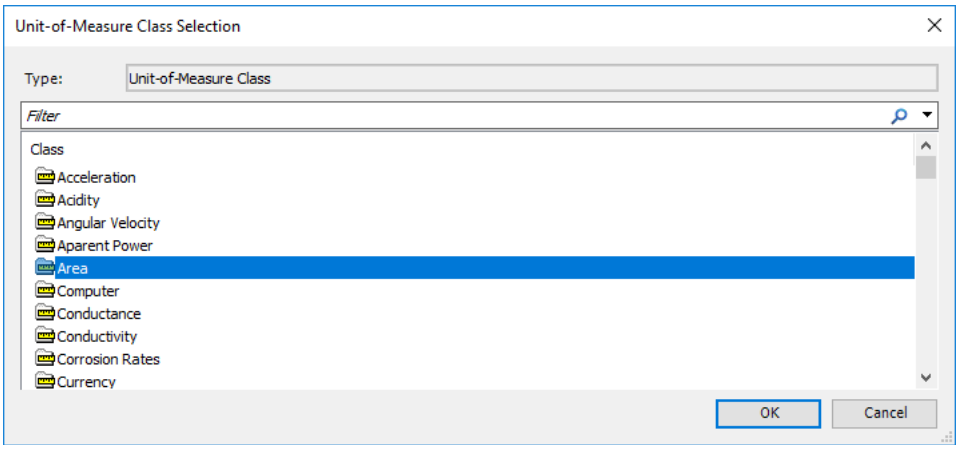
Standard Soccer field: 7,140 square meters

American Football field: 57,300 square feet

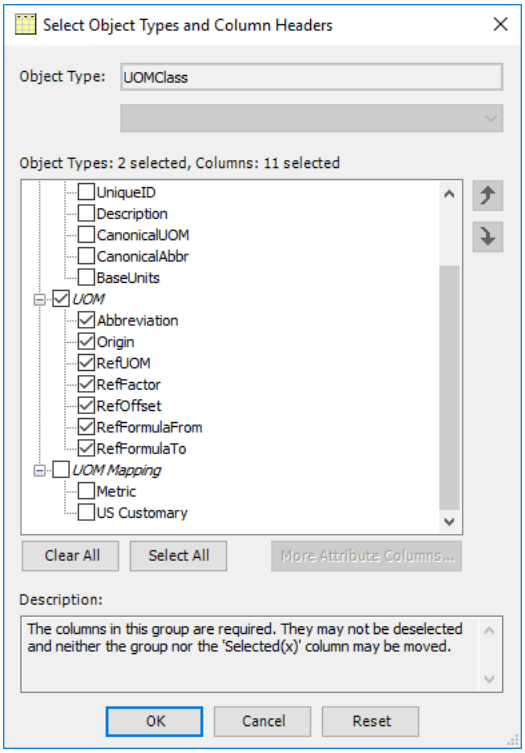
1. We will first load the relevant classes and units by going to Library > Units of Measure > Find UOM Classes…



2. We select Area from the Unit-of-Measure Class Selection window and click OK.



3. In the Select Object Types and Column Headers window, click Clear All to reduce the selection to the required columns. Since we want to introduce new units we will need the columns of the category UOM. We select the whole category UOM and click OK.



4. We get the classes Area and Length (as the base class) with all their units loaded into the spreadsheet.

	A	B	C	D	E	F	G	H	I	J	K
1	Selected(x)	Parent	Name	ObjectType	Abbreviation	Origin	RefUOM	RefFactor	RefOffset	RefFormulaFrom	RefFormulaTo
2	x		Length	UOMClass							
3	x	Length	meter	UOM	m	SystemDefined		1	0		
4	x	Length	inch	UOM	in	SystemDefined	meter	0.0254	0		
5	x	Length	U	UOM	U	UserDefined	inch	1.75	0		
6	x	Length	mils	UOM	mils	UserDefined	inch	0.001	0		
7	x	Length	mil	UOM	mil	UserDefined	inch	0.001	0		
8	x	Length	foot	UOM	ft	SystemDefined	inch	12	0		
9	x	Length	furlong	UOM	fl	UserDefined	foot	660	0		
10	x	Length	F	UOM	Feet	UserDefined	meter	0	0		
11	x	Length	DM	UOM	dm	UserDefined	meter	0	0		
12	x	Length	Ales	UOM	afeet	UserDefined	foot	1.1	0		
13	x	Length	32nd of an inch	UOM	32nd in.	UserDefined	inch	0.03125	0		
14	x	Length	sixteenth of an inch	UOM	sxi	SystemDefined	inch	0.0625	0		
15	x	Length	32nd inch	UOM	32nd in	UserDefined	sixteenth of an inch	0.5	0		
16	x	Length	µM	UOM	µm	UserDefined	meter	0	0		
17	x	Length	centimeter	UOM	cm	SystemDefined	meter	0.01	0		
18	x	Length	International nautical mile	UOM	nmi	SystemDefined	meter	1852	0		
19	x	Length	kilometer	UOM	km	SystemDefined	meter	1000	0		
20	x	Length	millimeter	UOM	mm	SystemDefined	meter	0.001	0		
21	x	Length	mile	UOM	mi	SystemDefined	inch	63360	0		
22	x	Length	yard	UOM	yd	SystemDefined	inch	36	0		
23	x		Area	UOMClass							
24	x	Area	square meter	UOM	m2	SystemDefined		1	0		
25	x	Area	square inch	UOM	in2	SystemDefined	square meter	0.00064516	0		
26	x	Area	square foot	UOM	ft2	SystemDefined	square inch	144	0		
27	x	Area	acre	UOM	acre	SystemDefined	square foot	43560	0		
28	x	Area	square mile	UOM	mi2	SystemDefined	square foot	27878400	0		
29	x	Area	square yard	UOM	yd2	SystemDefined	square foot	9	0		
30	x	Area	hectare	UOM	ha	SystemDefined	square meter	10000	0		
31	x	Area	square centimeter	UOM	cm2	SystemDefined	square meter	0.0001	0		
32	x	Area	square kilometer	UOM	km2	SystemDefined	square meter	1000000	0		
33	x	Area	square millimeter	UOM	mm2	SystemDefined	square meter	0.000001	0		

5. We add the definition of the two new units as follows (lines 2-33 as shown above have been deselected and are hidden in this screenshot for presentation purposes):

	A	B	C	D	E	F	G	H	I	J	K
1	Selected(x)	Parent	Name	ObjectType	Abbreviation	Origin	RefUOM	RefFactor	RefOffset	RefFormulaFrom	RefFormulaTo
34	x	Area	soccer field	UOM	soccer field		square meter	7140	0		
35	x	Area	football field	UOM	football field		square foot	57300	0		

Leave the column Origin blank as it will be automatically set to UserDefined.

6. Publish with Edit Mode: Create Only.

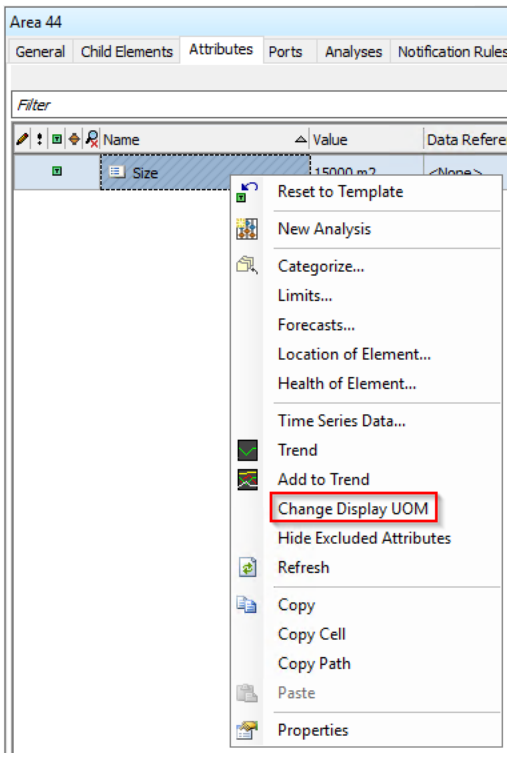
7. We can check the result also in PSE.

Area

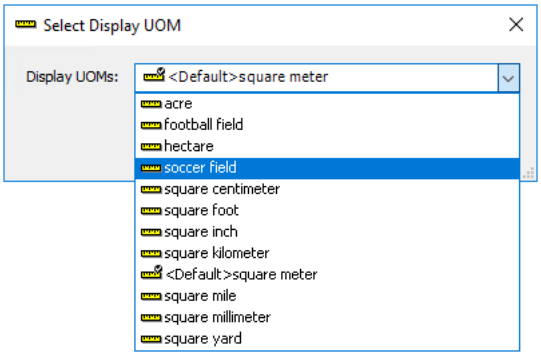
Filter

Name	Abbreviation	Class	Origin	Canonical	Reference
acre	acre	Area	System Defined	4046.8564224 m2	43560 ft2
football field	football field	Area	User Defined	5323.344192 m2	57300 ft2
hectare	ha	Area	System Defined	10000 m2	10000 m2
soccer field	soccer field	Area	User Defined	7140 m2	7140 m2
square centimeter	cm2	Area	System Defined	0.0001 m2	0.0001 m2
square foot	ft2	Area	System Defined	0.09290304 m2	144 in2
square inch	in2	Area	System Defined	0.00064516 m2	0.00064516 m2
square kilometer	km2	Area	System Defined	1000000 m2	1000000 m2
square meter	m2	Area	System Defined	1 m2	
square mile	mi2	Area	System Defined	2589988.110336 m2	27878400 ft2
square millimeter	mm2	Area	System Defined	1E-06 m2	1E-06 m2
square yard	yd2	Area	System Defined	0.83612736 m2	9 ft2

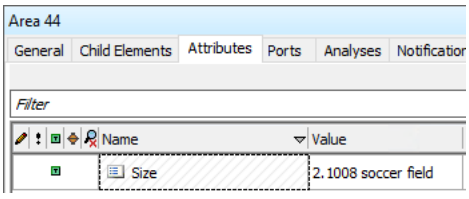
8. The new units can be used in any client tool. As a test we can see if the conversion works for Area 44. Right click on the Size attribute and select Change Display UOM.



9. Choose soccer field and click OK.



10. We can see the conversion resulted in a value of roughly 2 soccer fields.



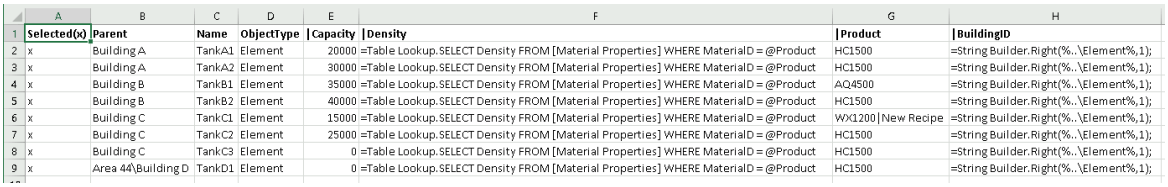
Analogously, we can change the display unit to football fields.

6.7. Retrieve Attribute Values

Database: **Tank Farm**

Instead of going to PSE (or another client tool) to check the values of an attribute as we usually do, we can also get the values displayed in a spreadsheet with PI Builder.

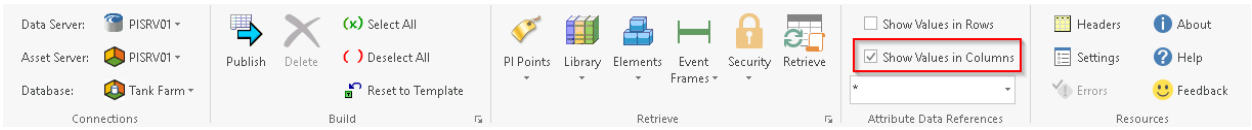
1. Go to Elements > Find Elements… and search for all elements from the Tank template (which include the derived template Tank with Foam Sensor). Click OK.
2. Click Clear All and expand the Attribute Columns to select a couple of attributes: Capacity, Density, Product and BuildingID. Click OK.



	A	B	C	D	E	F	G	H
1	Selected(x)	Parent	Name	ObjectType	|Capacity	|Density	|Product	|BuildingID
2	x	Building A	TankA1	Element	20000	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	HC1500	=String Builder.Right(%..\Element%,1);
3	x	Building A	TankA2	Element	30000	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	HC1500	=String Builder.Right(%..\Element%,1);
4	x	Building B	TankB1	Element	35000	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	AQ4500	=String Builder.Right(%..\Element%,1);
5	x	Building B	TankB2	Element	40000	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	HC1500	=String Builder.Right(%..\Element%,1);
6	x	Building C	TankC1	Element	15000	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	WX1200|New Recipe	=String Builder.Right(%..\Element%,1);
7	x	Building C	TankC2	Element	25000	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	HC1500	=String Builder.Right(%..\Element%,1);
8	x	Building C	TankC3	Element	0	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	HC1500	=String Builder.Right(%..\Element%,1);
9	x	Area 44\Building D	TankD1	Element	0	=Table Lookup.SELECT Density FROM [Material Properties] WHERE MaterialID = @Product	HC1500	=String Builder.Right(%..\Element%,1);

For Capacity and Product we see the values since those attributes have the data reference set to <None>.

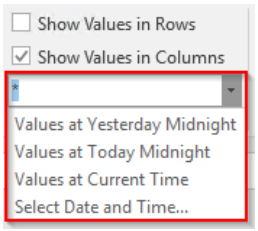
3. For attributes with a data reference, PI Builder displays the configuration string and not the current value. To show the values instead of the configuration, we can tick Show Values in Columns (for attributes listed in rows there is the Show Values in Rows option to populate the column AttributeValue).



4. Click the Retrieve button. This will give us the actual values of the attributes at the current time.

	A	B	C	D	E	F	G	H
1	Selected(x)	Parent	Name	ObjectType	|Capacity	|Density	|Product	|BuildingID
2	x	Building A	TankA1	Element	20000	3422	HC1500	A
3	x	Building A	TankA2	Element	30000	3422	HC1500	A
4	x	Building B	TankB1	Element	35000	2100	AQ4500	B
5	x	Building B	TankB2	Element	40000	3422	HC1500	B
6	x	Building C	TankC1	Element	15000	9213	WX1200|New Recipe	C
7	x	Building C	TankC2	Element	25000	3422	HC1500	C
8	x	Building C	TankC3	Element	0	3422	HC1500	C
9	x	Area 44\Building D	TankD1	Element	0	3422	HC1500	D

5. Different time contexts can be chosen via the field and associated drop-down list below.

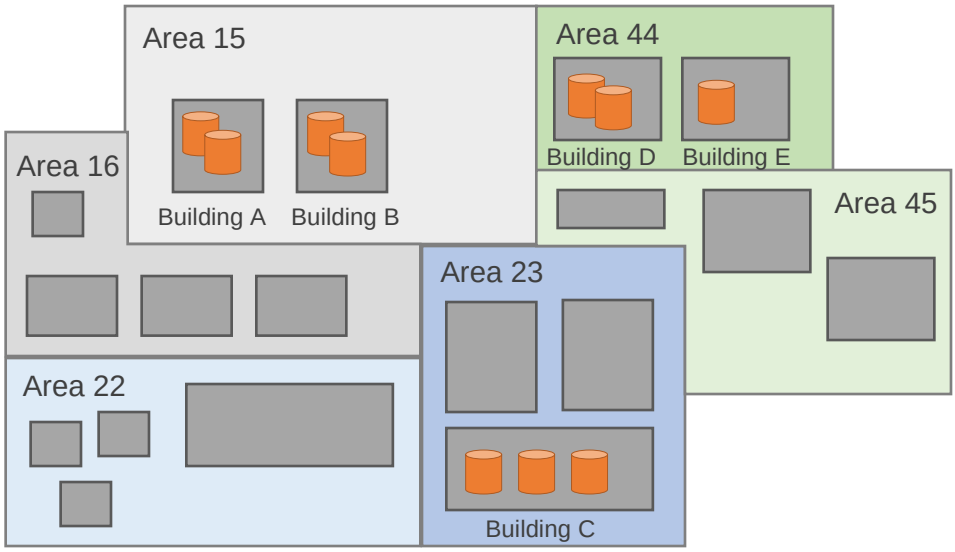


For full information refer to "*Show values from data references*" in the OSIsoft Documentation.

6.8. Create New Elements

Database: **Tank Farm**

We will now proceed to create the remaining elements as shown in the plant map below.



In the final version, we want to have all buildings grouped under area elements. We will not do this yet for the buildings which have already been created at the root level. In the next section we will see how to move elements in the hierarchy and will then finalize the configuration. For now, we will create the remaining elements and apply the correct hierarchy wherever this is possible without moving existing elements. Also, we need to consider where attribute values have to be set when creating the elements.

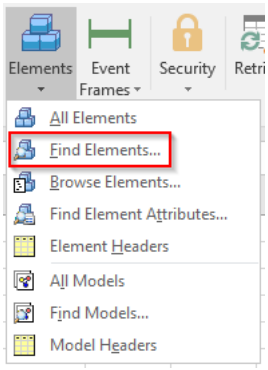
We need to create:

- TankD2 (Parent: Building D; template: Tank with Foam Sensor)

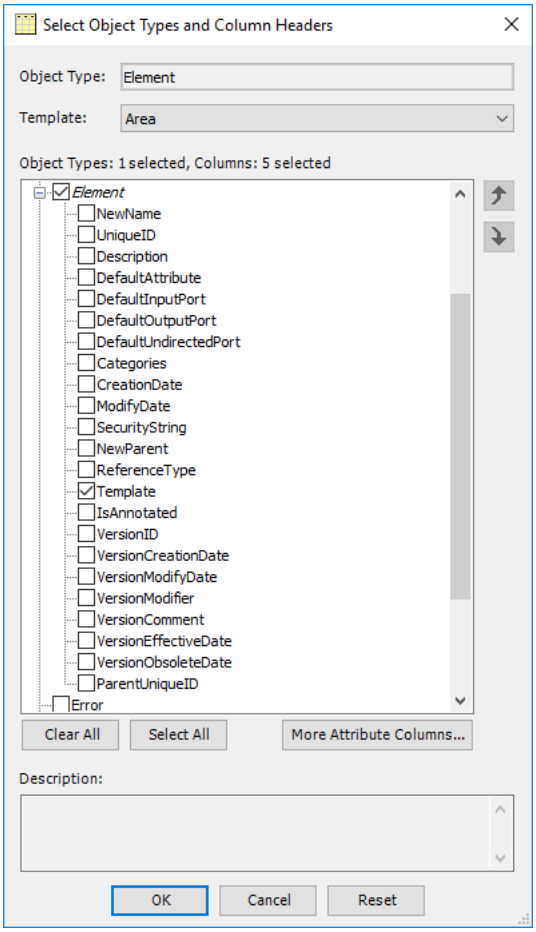
- Building E (Parent: Area 44; template: Building)
- TankE1 (Parent: Building E; template: Tank)
- Area 15 (template: Area; Size: 32,000 square meters)
- Area 23 (template: Area; Size: 24,000 square meters)

We will first create the remaining building and tank elements.

1. To see in PI Builder what we already have and the structure we need to create for the new elements we first load the existing elements into a spreadsheet. Open Elements > Find Elements…



2. Clear all search criteria to find all elements in the database. Click OK. In the Select Object Types and Column Headers, click Clear All. Since the new elements should be generated from templates, we need to add the Template column from the Element category.



3. We get the following loaded into the spreadsheet:

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**Template**
2	x		Area 44	Element	Area
3	x		Building A	Element	Building
4	x		Building B	Element	Building
5	x		Building C	Element	Building
6	x	Area 44	Building D	Element	Building
7	x		Data Archive	Element	
8	x		Reactor	Element	
9	x	Building A	TankA1	Element	Tank
10	x	Building A	TankA2	Element	Tank
11	x	Building B	TankB1	Element	Tank
12	x	Building B	TankB2	Element	Tank
13	x	Building C	TankC1	Element	Tank with Foam Sensor
14	x	Building C	TankC2	Element	Tank with Foam Sensor
15	x	Building C	TankC3	Element	Tank with Foam Sensor
16	x	Area 44\Building D	TankD1	Element	Tank with Foam Sensor

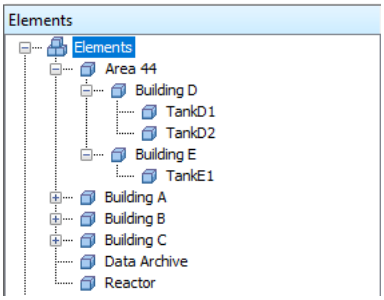
We can follow the example of Building D to build Building E as a child of Area 44 (row 6) and TankD1 to build TankD2 and TankE2 (row 16). Note that all other tanks and buildings still need to be moved under area elements which we will build soon and are therefore not a good syntax example in the sense of our desired final configuration.

4. We copy row 6 once and row 16 twice and make modifications to get what we show in rows 17 to 19:

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**Template**
2			Area 44	Element	Area
3			Building A	Element	Building
4			Building B	Element	Building
5			Building C	Element	Building
6		Area 44	Building D	Element	Building
7			Data Archive	Element	
8			Reactor	Element	
9		Building A	TankA1	Element	Tank
10		Building A	TankA2	Element	Tank
11		Building B	TankB1	Element	Tank
12		Building B	TankB2	Element	Tank
13		Building C	TankC1	Element	Tank with Foam Sensor
14		Building C	TankC2	Element	Tank with Foam Sensor
15		Building C	TankC3	Element	Tank with Foam Sensor
16		Area 44\Building D	TankD1	Element	Tank with Foam Sensor
17	x	Area 44	Building E	Element	Building
18	x	Area 44\Building D	TankD2	Element	Tank with Foam Sensor
19	x	Area 44\Building E	TankE1	Element	Tank

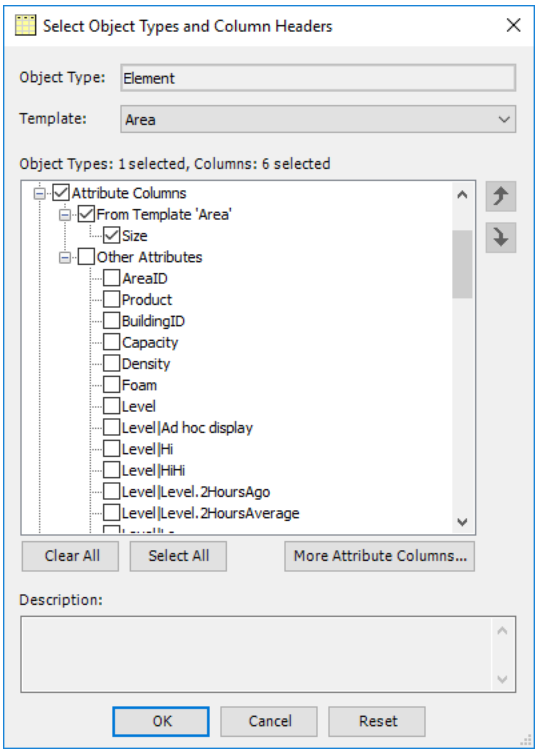
5. We select only those three rows and click Publish. We choose Edit Mode: Create Only and tick Create or update PI points to create the OPC tags right away.

6. In PSE, we now see the following hierarchy:



Next, we need to create the remaining area elements.

7. In a new spreadsheet, we go on Elements > Browse Elements…, select Area 44 in the Element Browser and click OK. We can use the previous column selection as a starting point (Required Columns and Template from the Element category). Since we want to specify the Size attribute for the elements we want to create, we additionally select the column Size from the Attribute Columns category. Click OK and Close.



8. We get the following:

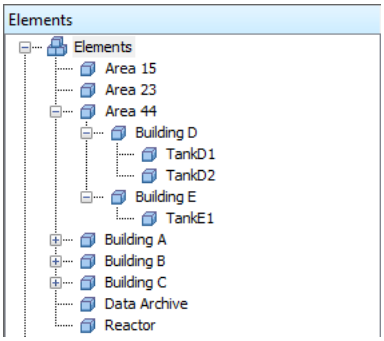
	A	B	C	D	E	F
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**Size**	**Template**
2	x		Area 44	Element	15000	Area

9. We copy row 2 twice, deselect row 2 and make modifications to rows 3 and 4 to arrive at the following:

	A	B	C	D	E	F
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**Size**	**Template**
2			Area 44	Element	15000	Area
3	x		Area 15	Element	32000	Area
4	x		Area 23	Element	24000	Area

10. Click Publish with Edit Mode: Create Only and click OK.

11. Checking the hierarchy in PSE (Refresh first), we should now see the following:



In the next section we will arrange the elements into the final configuration.

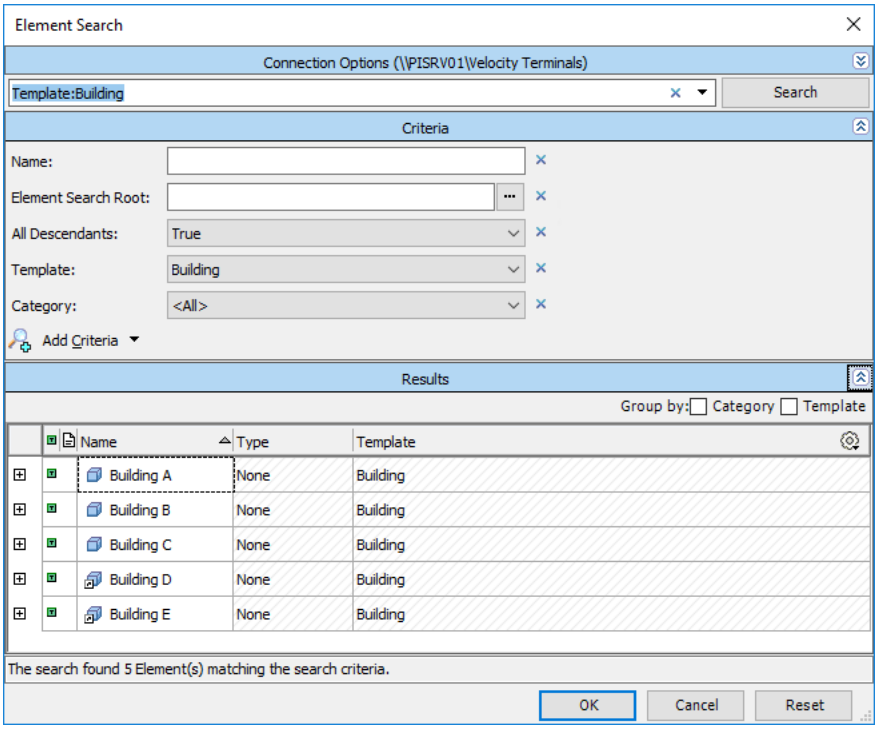
6.9. Move Elements within the Hierarchy

Database: **Tank Farm**

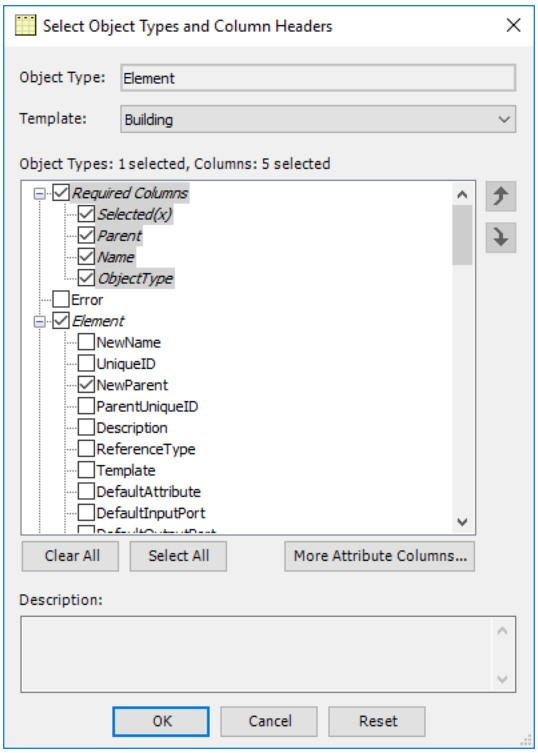
As you build an AF hierarchy you may need to move elements around. This can be done in PSE or in PI Builder, but we will only show the steps for PI Builder below. You might remember from your basics AF training that elements can be moved in PSE by dragging the element while pressing the Shift key. Later in this course you will find a refresher on this topic.

To arrive at our desired hierarchy we need to move Buildings A and B under Area 15 and Building C under Area 23. In PI Builder we will achieve this by assigning new parent elements.

1. Go to Elements > Find Elements…, select the Building template in the Element Search and click Search to find all building elements. Click OK.



2. In the Select Object Types and Column Headers window click Clear All and then select NewParent from the Element category. Click OK and Close.



3. We get the following:

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**NewParent**
2	x		Building A	Element	
3	x		Building B	Element	
4	x		Building C	Element	
5	x	Area 44	Building D	Element	
6	x	Area 44	Building E	Element	

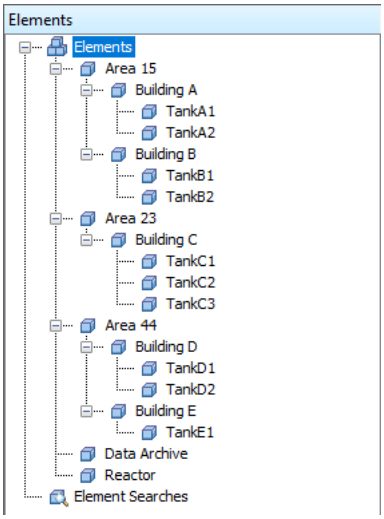
The buildings in Area 44 (rows 5 and 6) are already correctly placed. For the other buildings we need to assign the area as parent element in the column NewParent.

4. We apply the changes to arrive at the following:

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**NewParent**
2	x		Building A	Element	Area 15
3	x		Building B	Element	Area 15
4	x		Building C	Element	Area 23
5		Area 44	Building D	Element	
6		Area 44	Building E	Element	

Rows 5 and 6 were deselected here since nothing needs to be changed.

5. Click Publish and choose Edit Mode: Edit Only. Click OK and Close.
6. We can verify the result in PSE. Click Refresh and expand the hierarchy in the Browser. If the hierarchy is not correctly presented you might need to close and reopen PSE.

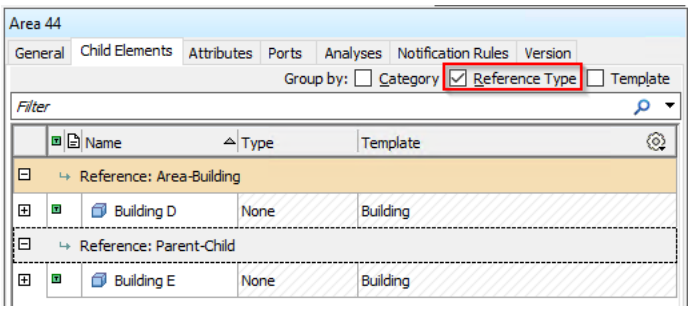


6.10. Change Reference Types

Database: **Tank Farm**

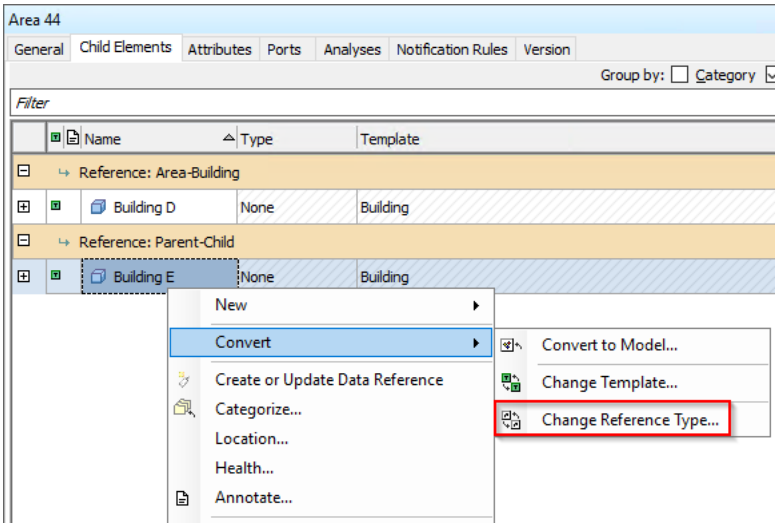
Since we also have dedicated reference types (see section *Build and Edit Templates and Other Library Objects* earlier in this lesson) for the relationship between areas and buildings and between buildings and tanks we will also set the reference types where it is still on the standard Parent-Child.
Let us first see how this can be done in PSE before we make the remaining changes in bulk in PI Builder.

1. In the Browser select Area 44 and open the Child Elements tab. Select Group by Reference Type.

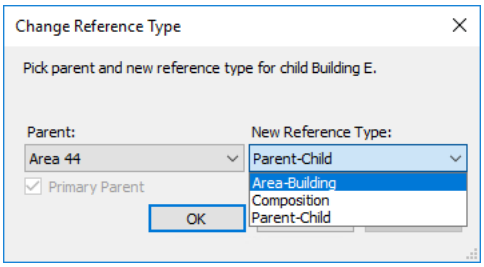


   Building D already uses the reference type Area-Building, while Building E has a reference type of Parent-Child.

2. To change the reference type for Building E, we right-click on Building E and select Convert > Change Reference Type…

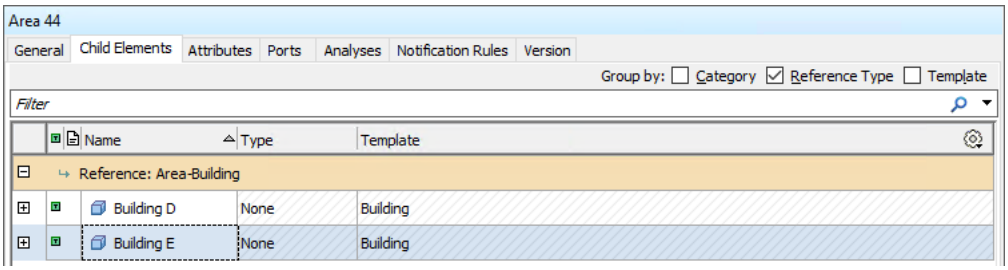


3. In the Change Reference Type window, select Area-Building as the new reference type.

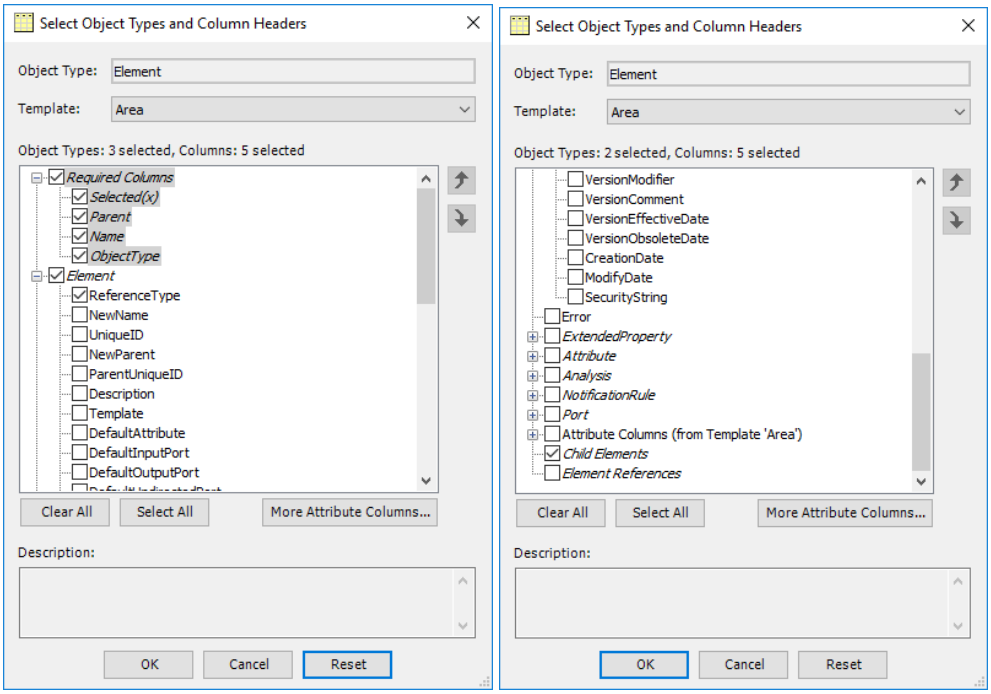


Click OK. Check In the change.

4. Both buildings are now shown with reference type Area-Building (you might need to refresh the view).



5. For changing the remaining reference types, we use PI Builder. Go to Elements > All Elements. In the Select Object Types and Column Headers window, Clear All and then select ReferenceType from the category Element and Child Elements at the bottom of the list. This is necessary to get not only elements from the root level. Click OK and Close.



6. We want to change the Parent-Child reference types to Area-Building or Building-Tank, respectively.

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**ReferenceType**
2	x		Area 15	Element	
3	x	Area 15	Building A	Element	Parent-Child
4	x	Area 15\Building A	TankA1	Element	Parent-Child
5	x	Area 15\Building A	TankA2	Element	Parent-Child
6	x	Area 15	Building B	Element	Parent-Child
7	x	Area 15\Building B	TankB1	Element	Parent-Child
8	x	Area 15\Building B	TankB2	Element	Parent-Child
9	x		Area 23	Element	
10	x	Area 23	Building C	Element	Parent-Child
11	x	Area 23\Building C	TankC1	Element	Parent-Child
12	x	Area 23\Building C	TankC2	Element	Parent-Child
13	x	Area 23\Building C	TankC3	Element	Parent-Child
14	x		Area 44	Element	
15	x	Area 44	Building D	Element	Area-Building
16	x	Area 44\Building D	TankD1	Element	Building-Tank
17	x	Area 44\Building D	TankD2	Element	Parent-Child
18	x	Area 44	Building E	Element	Area-Building
19	x	Area 44\Building E	TankE1	Element	Parent-Child
20	x		Data Archive	Element	
21	x		Reactor	Element	

7. Change the reference type for all tanks to Building-Tank and for all buildings to Area-Building.

	A	B	C	D	E
1	**Selected(x)**	**Parent**	**Name**	**ObjectType**	**ReferenceType**
2	x		Area 15	Element	
3	x	Area 15	Building A	Element	Area-Building
4	x	Area 15\Building A	TankA1	Element	Building-Tank
5	x	Area 15\Building A	TankA2	Element	Building-Tank
6	x	Area 15	Building B	Element	Area-Building
7	x	Area 15\Building B	TankB1	Element	Building-Tank
8	x	Area 15\Building B	TankB2	Element	Building-Tank
9	x		Area 23	Element	
10	x	Area 23	Building C	Element	Area-Building
11	x	Area 23\Building C	TankC1	Element	Building-Tank
12	x	Area 23\Building C	TankC2	Element	Building-Tank
13	x	Area 23\Building C	TankC3	Element	Building-Tank
14	x		Area 44	Element	
15	x	Area 44	Building D	Element	Area-Building
16	x	Area 44\Building D	TankD1	Element	Building-Tank
17	x	Area 44\Building D	TankD2	Element	Building-Tank
18	x	Area 44	Building E	Element	Area-Building
19	x	Area 44\Building E	TankE1	Element	Building-Tank
20	x		Data Archive	Element	
21	x		Reactor	Element	

8. Click Publish and choose Edit Mode: Edit Only. Click OK and Close.
9. In PSE, we can check the Child Element tabs of the area and building elements to verify the modification.

7. Security

7.1. Introduction to AF Security

AF Identities and Mappings

AF Identity	Description
Administrators	By default, this identity has all access permissions to every collection and object on the AF server, including all databases. It cannot be modified or deleted. It is recommended that access to this identity is restricted to only a few users.
Engineers	This identity has the same privileges as *Administrators*, with the exception of the *Admin (a)* permission. This identity is also not allowed to delete AF databases. It is recommended that this identity be restricted to those users who are defining the asset database. Additional identities should be created to narrow the scope of access within AF.
World	This identity has read access permissions to every collection and object on the AF server. More information see below.

The following table describes the access permissions you can assign to AF identities for all objects in the AF hierarchy.

Access right	Abbreviation	Definition
Read	r	Enables a user to view the object.
Write	w	Enables a user to create and modify an object. The exception is that event frames and transfers also require Write Data permission on the element template from which they are created. Additionally, if users do not have Write permission on the AF database, they cannot modify any object within the database, regardless of the specific permission on that object.
Read/Write		Enables a user to read and write to the associated object.

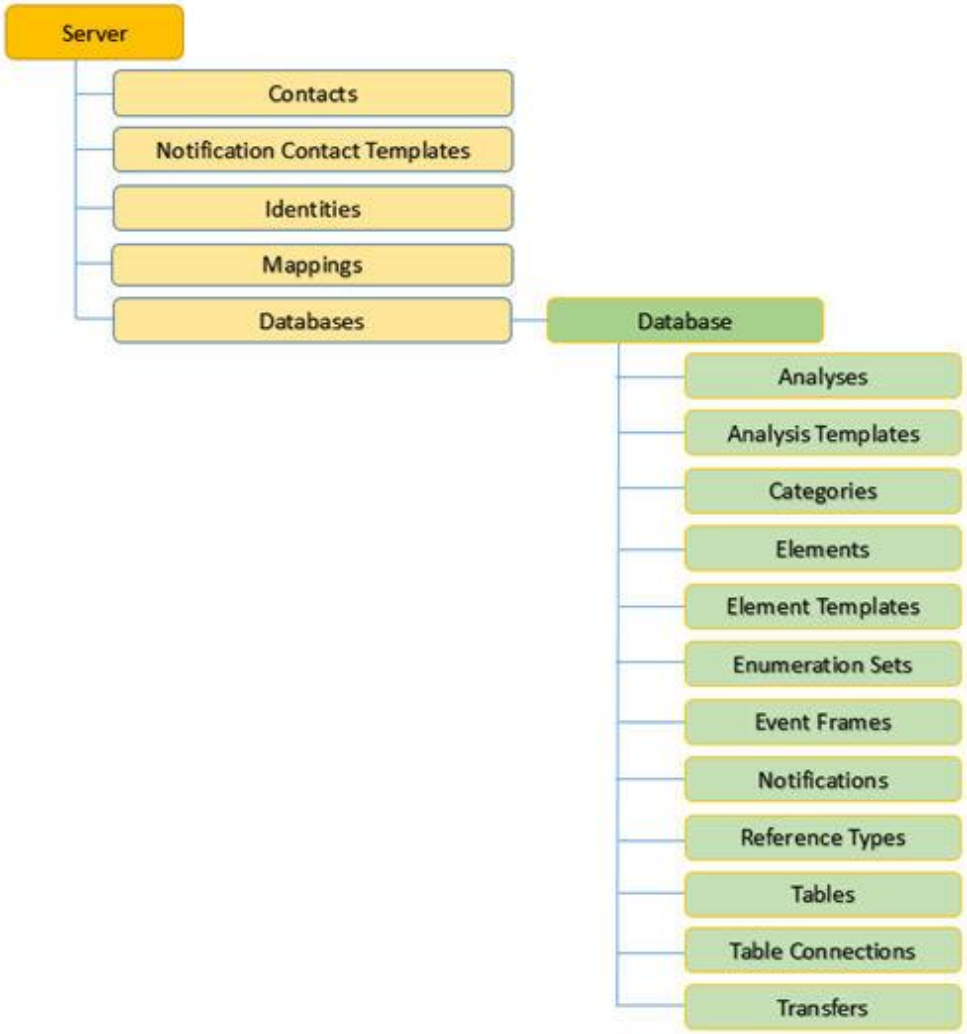
Read Data	rd	Enables a user to read element's attribute values (non-configuration items).
Write Data	wd	Enables a user to modify element's attribute values (non-configuration items). Additionally, this permission controls whether a user can create or modify event frames.
Subscribe	s	Enables a user to subscribe and unsubscribe to a notification.
SubscribeOthers	so	Enables a user to subscribe and unsubscribe other users to a notification.
Delete	d	Enables a user to delete an object.
Execute	x	Enables a user to perform most actions on an analysis case. Only used in Pimsoft SigmafineTM data reconciliations. The PI Analysis Service does not use this permission. The Write permission is required to modify, run, and stop asset analyses.
Admin	a	Enables a user to modify the security settings, or owner, of an object. Also allows to force an Undo Check Out on an object that is checked out to another user, as well as to lock and unlock an event frame..

Setting permissions can be done for individual AF objects or for collections of objects. When you create new objects, except for child elements, the collection security is used as the default security. When you create a child element, the security descriptor of the parent element becomes its default security.

AF Security Hierarchy

The following chart shows the structure of the AF objects in an AF Server. Each securable AF object (element, event frame, and notification, and so on) throughout the hierarchy has an associated security descriptor that contains the access permissions information for that object.

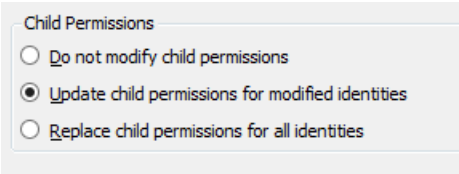
All AF objects of the same type belong to a collection. For example, every AF element in a database belongs to the Elements collection for that database. Each collection also has an associated security descriptor that contains access permission information.



Permission Inheritance

When you change the access permissions for an element, the access permissions for any parent or child elements might also change. The behavior depends on the reference type.

When an object or collection is created, a default set of access permissions is assigned, based on the access permissions that are set on the parent. When access permissions are set on a parent, the following Child Permission settings on the Permissions page of the Security Configuration window are evaluated:

Reference Type	Behavior
Composition	Access permissions for child and parent are always the same.
Weak	Access permissions are never inherited.
Parent – Child	When access permissions are set on a parent, the Child Permission settings in the Security Configuration window depend on option used 

Option	Description
Do not modify child permissions	Prevents access permissions that have been set for the current object or collection from being replicated to child collections and objects in the AF hierarchy. Default for AF server 2.5 and earlier
Update child permissions for modified identities	For each selected item on the Items to Configure list in the Security Configuration window, replicates the access permissions for all child collections and objects for each identity on the Identities list whose access permissions have been modified. Default for AF server 2.6 and later.

Replace child permissions for all identities	For each selected item on the Items to Configure list in the Security Configuration window, replaces all child permissions for every identity on the Identities list with the parent access permissions. Hint: Before you apply this option, review access permission settings for all items on the Items to Configure list to avoid unintentionally overwriting custom permissions that may have been applied elsewhere in the collection hierarchy!

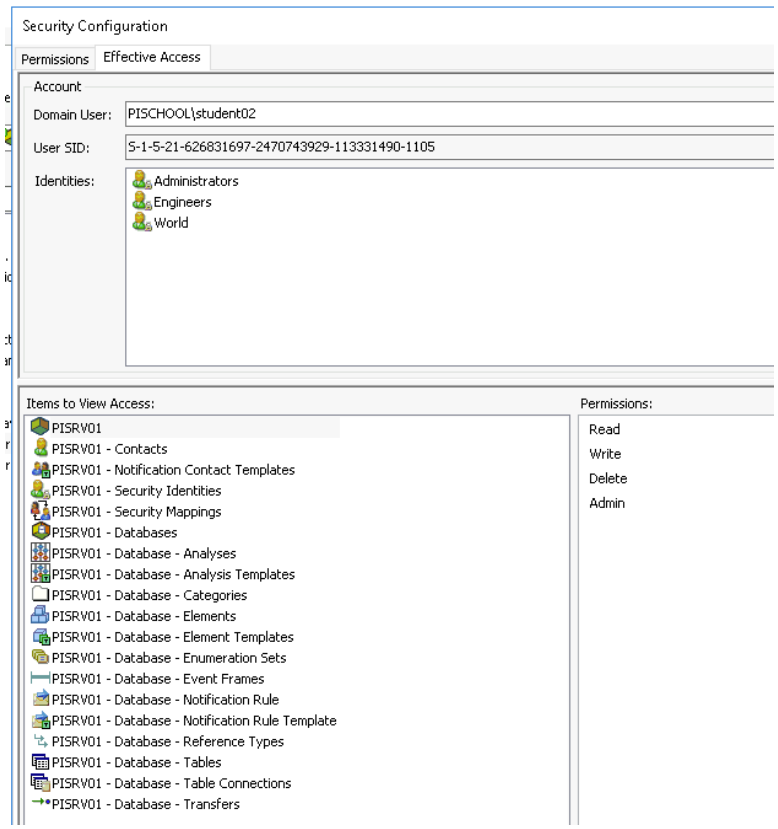
See also *Security configuration in PI AF* in the OSIsoft Documentation.

View Effective Access by User

1. Choose one of the following methods to open the Security Configuration window:

To open from ...	Do this ...
PI AF Server Properties Window	On the toolbar, click  . In the PI AF Server Properties window, click the Security link.
Select Database Window	On the toolbar, click the Database button. In the Select Database window, click the Edit Security button.
PI System Explorer Browser	Right -click an object or collection and click Security. In the PI AF Server Properties window, click the Security link.

2. In the Security Configuration window, click the Effective Access tab.
3. In the Domain User field, enter a user's Windows domain name whose permissions you want to view.



4. Press Tab or Enter to validate your entry.

   Note:You can click  next to the Domain User field to search for a user's Windows domain name.

   The fields and lists on the Effective Access page are populated with the following information:

   a. User SID field: Displays the user's security identifier

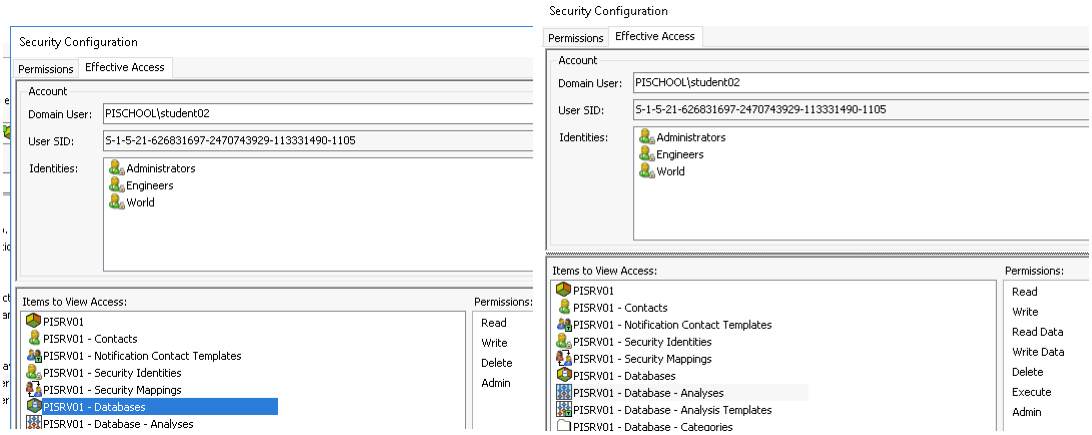
   b. Identities list: Lists all the PI AF identities that are mapped to the Windows domain user account entered in step 3

   c. Items to View Access list: Displays the item(s) selected for security configuration. This list is reflective of the item(s) shown in the Items to Configure list on the Permissions card.

   d. Permissions list: Displays the access permissions for the currently selected item in the Items to Configure list.

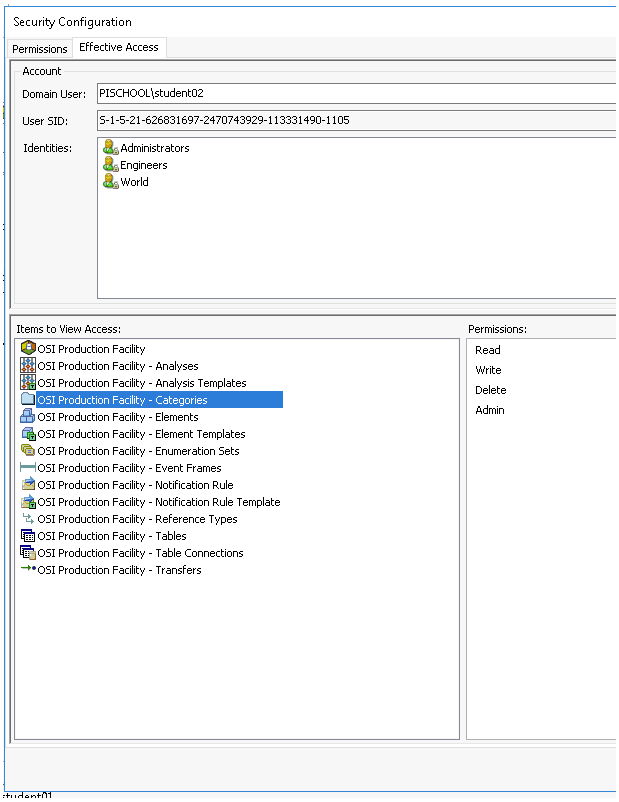
5. Select a different item in the Items to View Access list. The user's access permissions for the selected item are displayed in the Permissions list.

   Examples:



6. Repeat step 6 as needed to view access permissions for other items.

7. Click OK to exit the Security Configuration window.



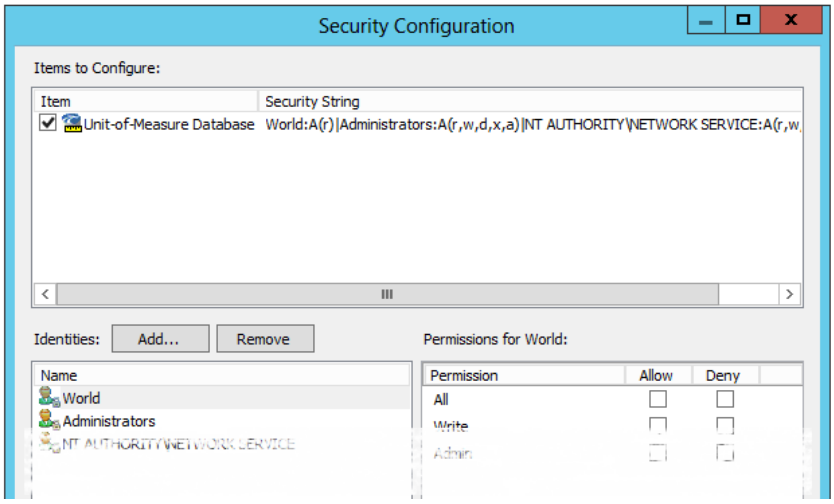
7.2. UOM Database Security

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: UOM Database Security

The AF Unit of Measure (UOM) database is shared across all AF databases. You cannot set permissions for individual UOMs or UOM classes. Permissions for individual UOMs or UOM classes. Permissions can only be set for the entire UOM database.

To open UOM *Security Configuration* window:

1. In the Navigator pane, select **Unit of Measure.**
2. On the toolbar, click the **UOM Security** button.



3. Examine the difference between World and Administrators Identity Permissions.
    a. World has no permissions to make changes within the UOM database.
    b. Administrators have access to make changes to the UOM database.
4. Effective permissions can be reviewed to verify individual access. Examine student01low's permissions.
    a. Select the Effective Access Tab.
    b. Search for a domain user; student01low.
    c. Review effective access for individual. The individual has access to read all units of measure configured within the UOM database.

Security Tips

- Administrator privileges at the server level provides access to **EVERY OBJECT**, regardless of their security settings
- If you want to edit an element, you need *write* permissions on the Elements collection **AND** on the particular element
- Library objects, such as templates, enumeration sets, UOMs, and references types **ALWAYS** have read permission regardless of their security settings
- Deny settings override any granted allow permissions

7.3. AF Permissions

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: AF Permissions

Database: **OSI Production Facility**

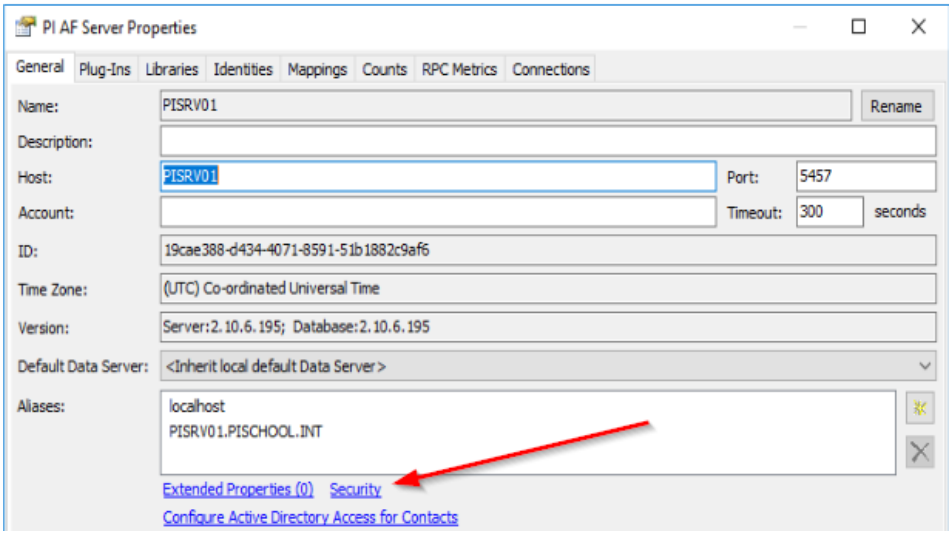
Exercise:

Use your knowledge of AF security to accomplish the following use case in the OSI Production Facility database: You want to create an AF identity which, when used for a user or a user group, will allow that identity to create elements from template, but that identity should NOT be able to edit library objects at all i.e. they should not be able to edit the template. (you can use the pischool\student01low account for your testing. This account has the same password as student01 and it is currently mapped to the ElementBuilders identity.

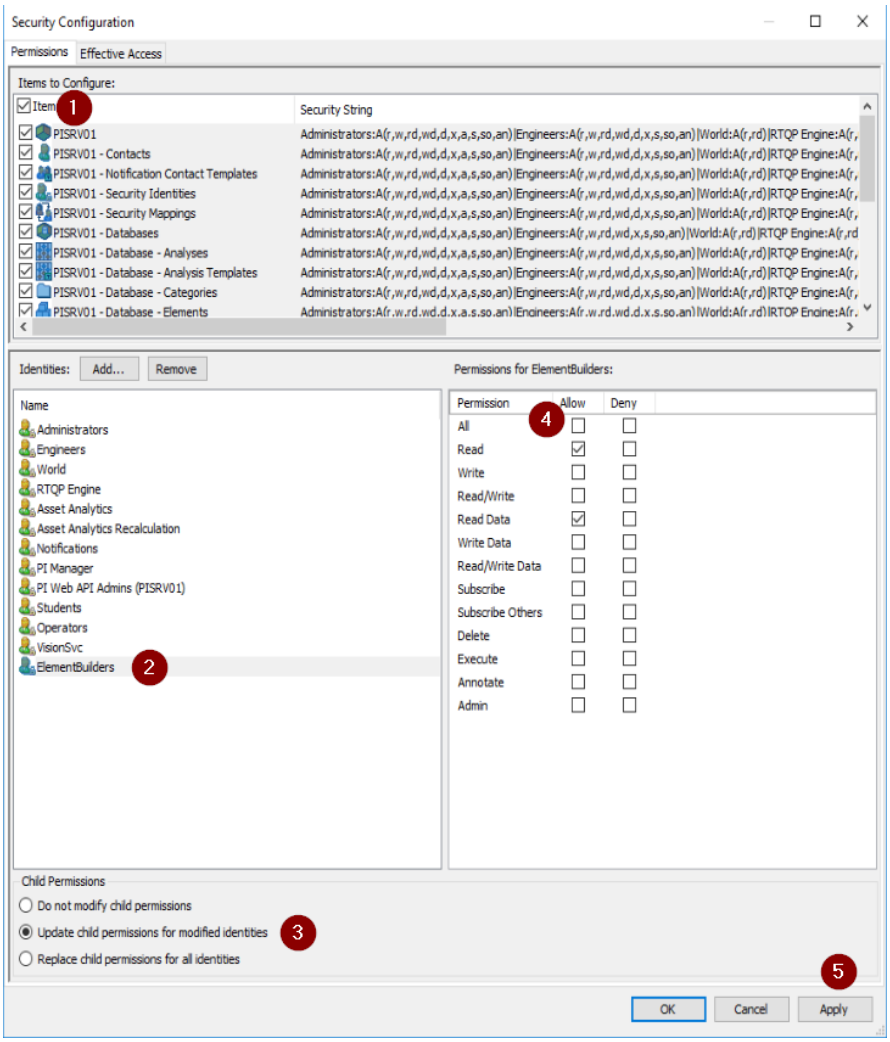
Step by Step:

The main idea here is that we want to give the user access to the non template collections, but not the template collections. This will allow them to create instances of templates, but not create or alter the templates themselves.

1. Sign in to the AF Server with an account with Administrator's permissions (for example Student01) and go to File > Server Properties > Security:



2. Next to identities click 'Add…' to open the 'Select Identity' dialog box and choose ElementBuilders and OK.
3. Next ensure that you have all 'Items to Configure' selected, choose ElementBuilders and then click the check box by Allow: All to remove all permissions and then select only Read and Read Data. Finally click 'Apply'



This will give the ElementBuilders identity read access to the entire AF server.

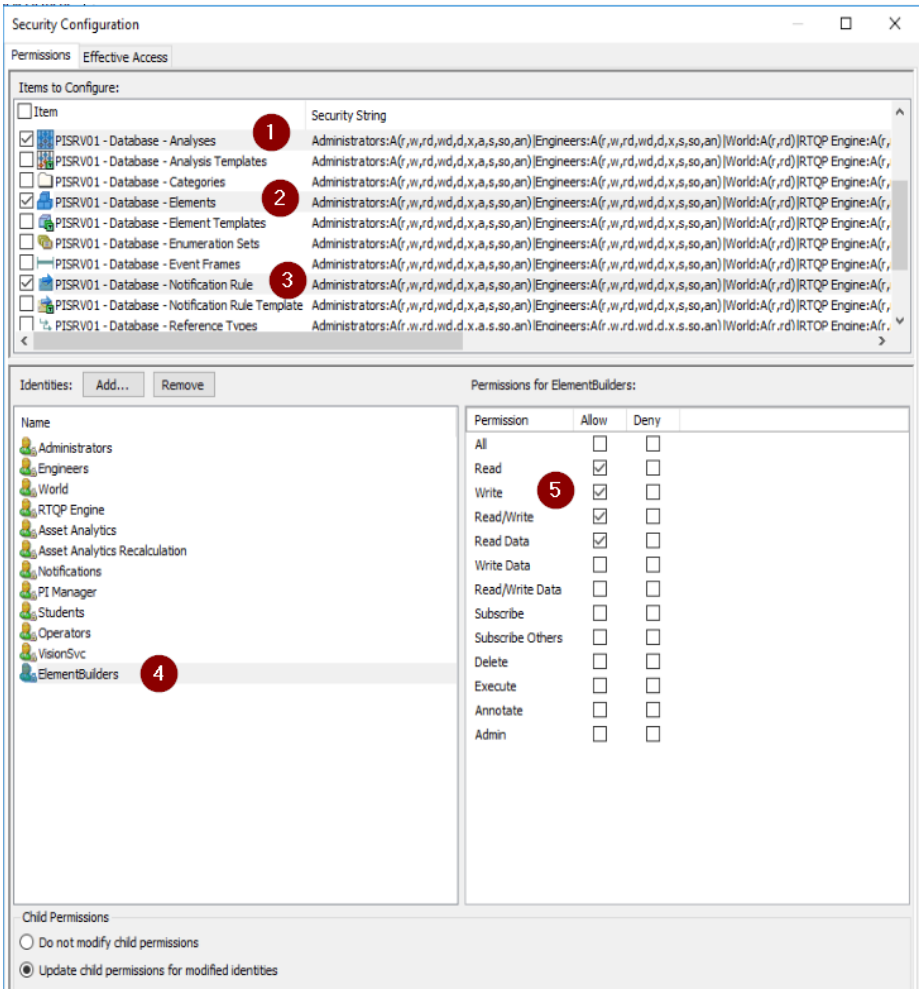
4. Then with this same window still open, let's select the checkbox at the very top to unselect all the Items. Next check the boxes next to

    a. PISRV1 – Database – Elements,

    b. PISRV1 – Database – Analyses,

    c. and PISRV1 – Database – Notification Rules

    while making sure that you then select ElementBuilders again. Next select the option for Write. Your screen should look like this:



Now hit OK to apply the changes and close the window. This should allow your test account to create and check in Elements created form templates, but not allow that account to create or make changes to templates.

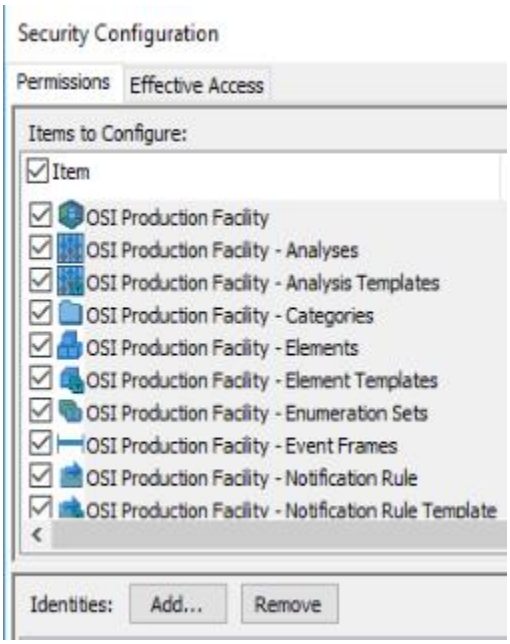
5. Upon closing and reopening the AF security (using previously outlined steps), verify the ElementBuilders permissions are correct. The boxes may appear different as all Items are selected in the Security Configuration options. Boxes filled in completely do not apply to all items selected; in this case, the Write permission is limited to the 3 previously identified databases.

7.4. Database Level Security Settings

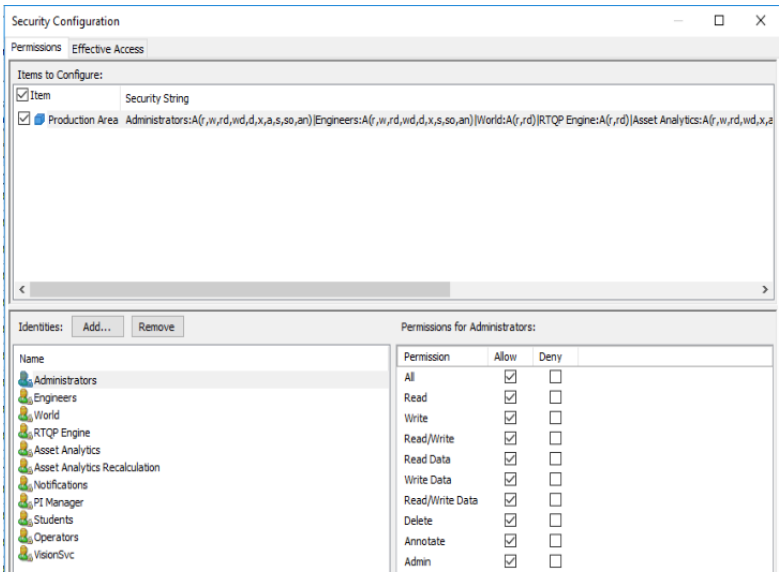
Database: **OSI Production Facility**

What we did in the previous exercises was changing permissions on the AF server on a server wide level. But say for instance, you would like to do something similar on a database level or you only want to make changes within a particular database that you manage without changing server level permissions. In this case, you could open the database security by opening File > Database… > Right click the database of interest and then choose Security…

Notice how we have a very similar dialog box to what we were working with in the last exercise, but this one lists the 'Items to Configure' in a slightly different way:



Here we see the form <DatabaseName> - <collection type> instead of <server> - [Database] – [<Collection>] So here we can see that any changes we make here will apply to this database and the objects in it. The server level is not affected. Essentially, we can only edit security of the Database (green) section in the security hierarchy diagram rather than the Server (yellow) sections. In addition, we can edit security of individual objects by right clicking on the object and choosing security. For example, we can do this for an element and we would see this:



So, depending on the granularity you need in your AF system: server level, database level, object level, you can get the security you require.

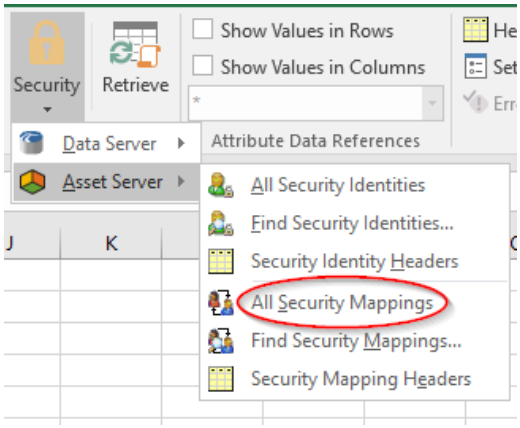
7.5. Editing Security in PI Builder

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Editing Security in PI Builder

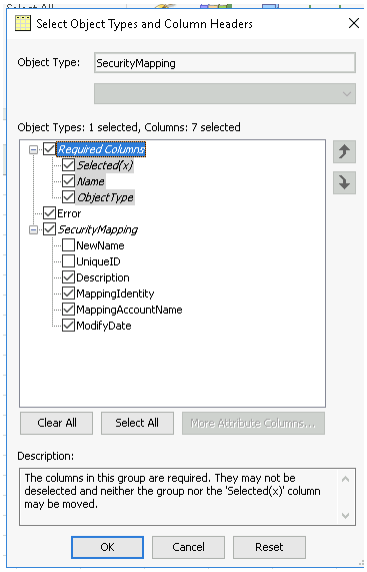
We are going to use the PI Builder functionality to add an **Engineers** mapping to correspond with the **Students** Active Directory (AD) group.

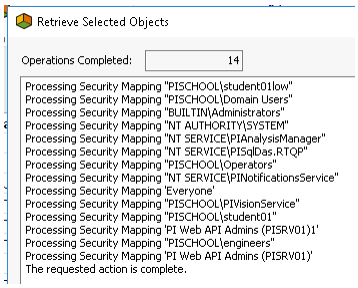
Step by step:

1. Create a new workbook or sheet. In the ribbon click the down arrow under **Security** > Asset Server > **All Security Mappings**



2. The **Select Object Types and Column Headers** dialog pops up. Here you can make selections on what information about the PI AF Mapping you would like to retrieve into the workbook. Just accept the default by clicking the **OK** button.



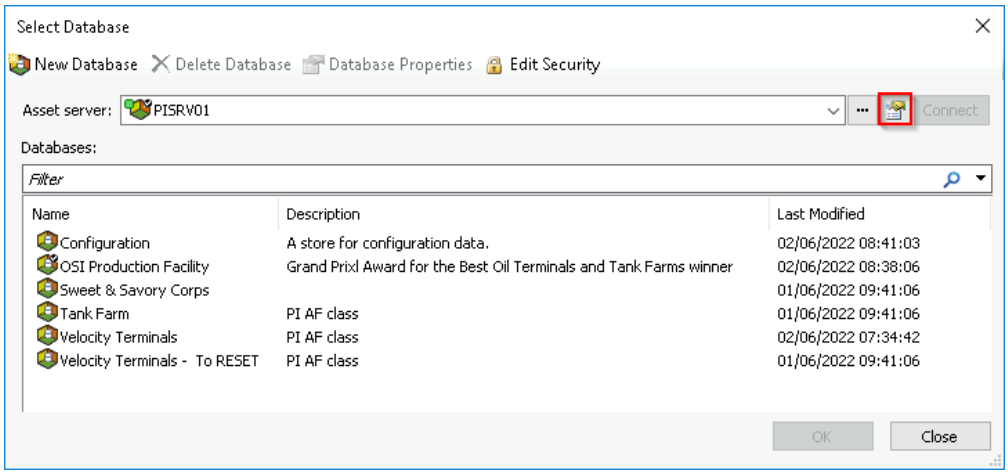


3. Copy the row with ‘Everyone’ in column B and paste it at the base of the spreadsheet. Select only the new bottom row and change the rest of that row as shown below.

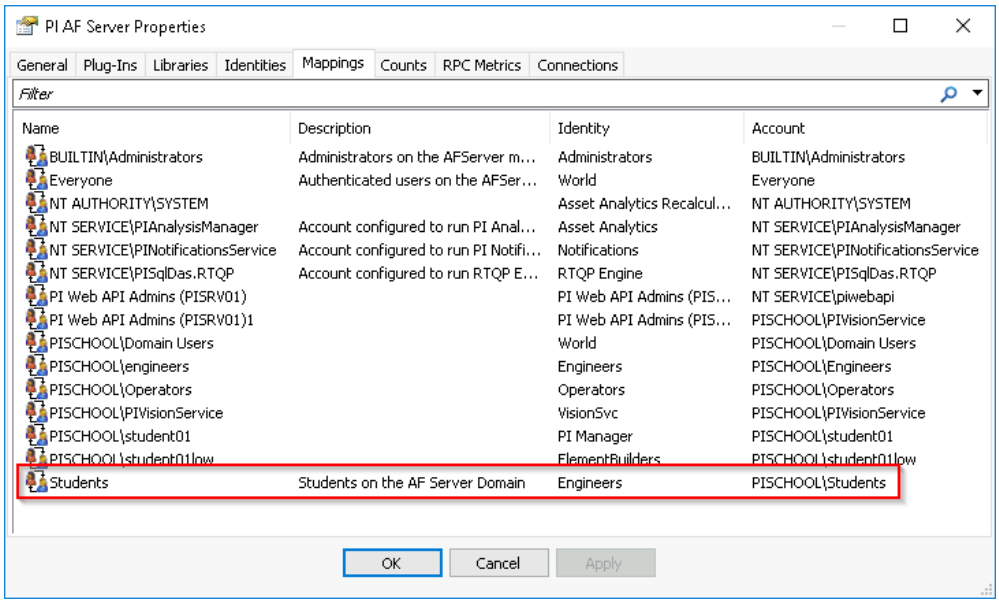
	A	B	C	D	E	F	G	H
1	Selected(x)	Name	ObjectType	Error	Description	MappingIdentity	MappingAccountName	ModifyDate
2		PISCHOOL\student01low	SecurityMapping			ElementBuilders	PISCHOOL\student01low	12/02/2021 02:45
3		PISCHOOL\Domain Users	SecurityMapping			World	PISCHOOL\Domain Users	12/02/2021 02:57
4		BUILTIN\Administrators	SecurityMapping		Administrators on the AFServer machine	Administrators	BUILTIN\Administrators	20/12/2019 22:27
5		NT AUTHORITY\SYSTEM	SecurityMapping			Asset Analytics Recalculation	NT AUTHORITY\SYSTEM	17/07/2020 07:25
6		NT SERVICE\PIAnalysisManager	SecurityMapping		Account configured to run PI Analysis Service	Asset Analytics	NT SERVICE\PIAnalysisManager	20/12/2019 22:35
7		NT SERVICE\PISqlDas.RTQP	SecurityMapping		Account configured to run RTQP Engine	RTQP Engine	NT SERVICE\PISqlDas.RTQP	20/12/2019 22:29
8		PISCHOOL\Operators	SecurityMapping			Operators	PISCHOOL\Operators	01/06/2020 04:30
9		NT SERVICE\PINotificationsService	SecurityMapping		Account configured to run PI Notifications Service	Notifications	NT SERVICE\PINotificationsService	20/12/2019 22:36
10		Everyone	SecurityMapping		Authenticated users on the AFServer Domain	World	Everyone	20/12/2019 22:27
11		PISCHOOL\PIVisionService	SecurityMapping			VisionSvc	PISCHOOL\PIVisionService	17/07/2020 01:27
12		PISCHOOL\student01	SecurityMapping			PI Manager	PISCHOOL\student01	30/12/2019 03:20
13		PI Web API Admins (PISRV01)1	SecurityMapping			PI Web API Admins (PISRV01)	PISCHOOL\PIVisionService	17/07/2020 05:08
14		PISCHOOL\engineers	SecurityMapping			Engineers	PISCHOOL\Engineers	05/06/2020 05:13
15		PI Web API Admins (PISRV01)	SecurityMapping			PI Web API Admins (PISRV01)	NT SERVICE\piwebapi	30/12/2019 23:04
16	x	Students	SecurityMapping		Students on the AFServer Domain	Engineers	PISCHOOL\Students	

4. Click the Publish button in the ribbon. Click **OK** and **Close** on the next two dialogs that pop up.

5. Open the **PI System Explorer** and click Database. Once there click on the **AF Server Properties.**



6. Then inspect the **Mappings** tab to confirm the change that was made.



8. Working with the Elements

8.1. Copying, Moving, Creating References, and Arrange Elements

Before reading this section, please refer to the following course YouTube video:
OSIsoft Learning: Copying, Moving, Creating References, and Arrange Elements

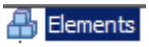
Database: **Velocity Terminals**

Elements can be referenced, copied, or moved using the mouse by dragging and dropping. By using the **Ctrl** or the **Shift** key you can control the behavior of the drag and drop result.

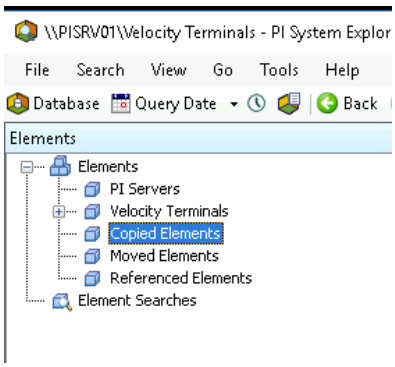
Within **Velocity Terminals**, this is already a complete referenced hierarchy already created. The initial hierarchy is under the Locations with the referenced hierarchy being organized slightly different under Division.

Add three new Elements and name them **Copied Elements**, **Moved Elements**, and **Referenced Elements** based on no Template. You are going to use these to be parents for copied, moved, and referenced child-Elements, respectively. Arrange the Elements view to be by **Template**. Copy a tank Element as a child-Element to **Copied Elements**, move a tank Element as a child-Element to **Moved Elements**, reference a tank Element as a child-Element to **Referenced Elements**.

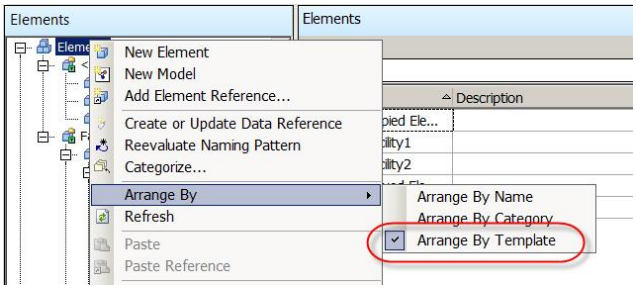
Step by step:

1. If you are not in the **Elements** view press **Ctrl+1** key combination.
2. Right click the 

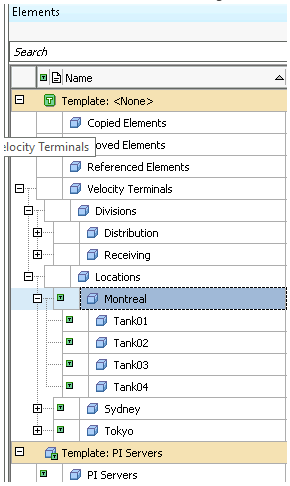
in the **Browser** pane and select **New Element** (select <None> as the Template). Rename the new Element by selecting it, pressing the **F2** key, and type **Copied Elements**. Create two more Elements in the same way and rename them to **Moved Elements** and **Referenced Elements**, respectively.



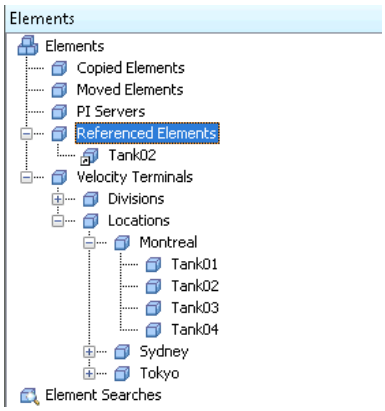
3. Select the **Elements** in the **Browser** then right click it. Select **Arrange By>Arrange By Template**.



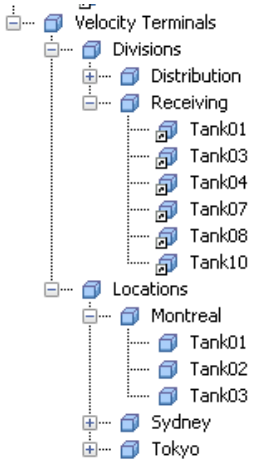
4. Expand the tree in the **Browser** pane to show the Elements under **<None>** and the Elements under **Velocity Terminals>Locations>Montreal>Tank02.**



5. Drag **Locations>Montreal>Tank02** Element to the **Referenced Elements** Element. Select **Weak Reference** in when the **Choose Reference Type** dialog pops up. This now makes a reference to the Tank02 Element, but the two Elements are identical. **Note** that the mouse icon changes to  when you hover over the destination Element.

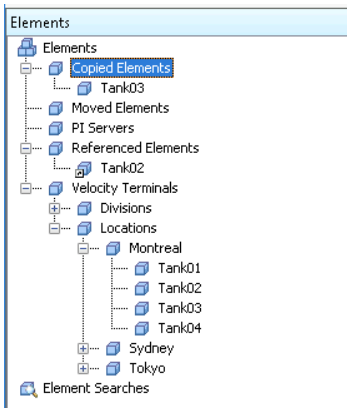


6. In the original Velocity Terminal, Division is set up as a Referenced Element copy organized in another hierarchy of Locations.



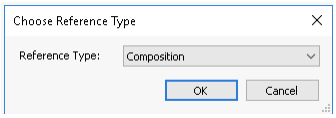
7. Hold down the **Ctrl** key and drag **Locations>Montreal>Tank03** Element to the **Copied Elements** Element. This now makes a copy **Tank03** and the two Elements are now different.

   Note that the mouse icon changes to when you hover over the destination Element.



8. Hold down the **Shift** key and drag **Locations>Montreal>Tank04** Element to the **Moved Elements** Element. This now moves that Tank04 from one location to the other.

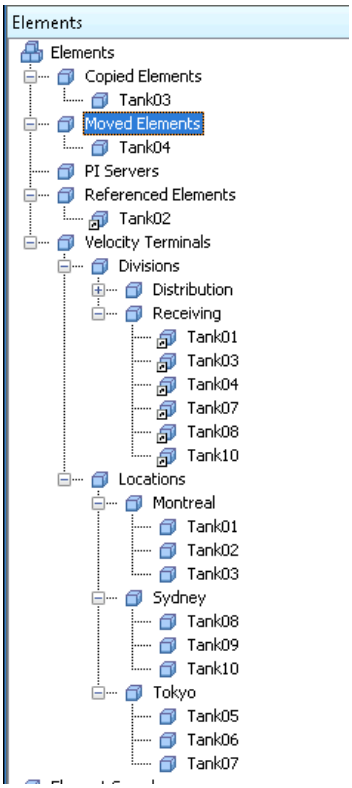
   During the move, a dialog screen to select the type of reference type is requested.



   A reference type with a *reference strength of composition* means that the child element is really a part of the parent and does not exist without the parent. If you delete a parent element that has a child element that is compositionally referenced, you delete the child element also.

   For example, use a composition reference when the two objects in the relationship are considered one item. For example, a meter might be composed of two sensors, and so you would use a composition reference between the *Meter* element and each of its two child *Sensor* elements. When you delete the *Meter* element, the child *Sensor* elements are also automatically deleted.

   Note that the mouse icon changes to  when you hover over the destination Element.



9. Be sure to check in changes to save the database modifications.

For more information see also *Element references in the asset hierarchy* in the OSIsoft Documentation.

8.2. Locate Other References to the Same Element

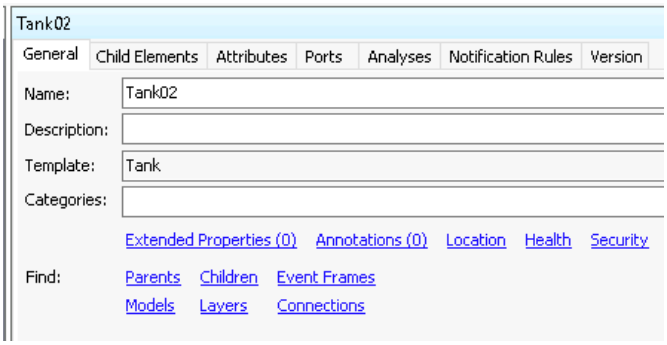
Database: **Velocity Terminals**

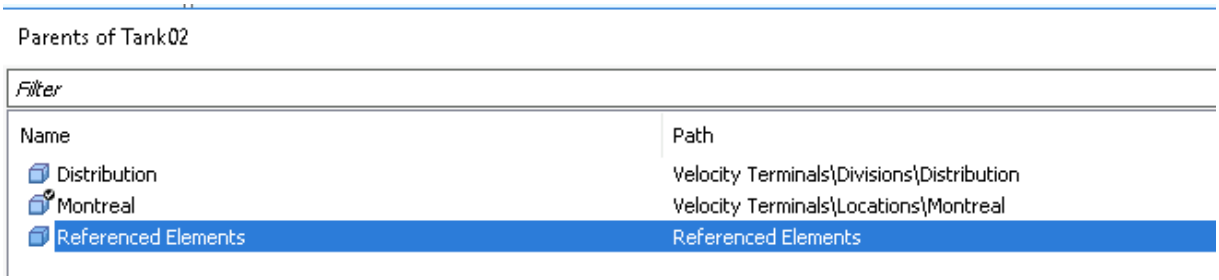
A single element can be referenced in multiple places in a hierarchy. You can find all the elements in the hierarchy that contain the element or a reference to the element.

Procedure

1. Select the element Tank02 in the Elements browser.
2. In the General tab, click the Parents link.

   The element's parents are displayed in the Parents of *Element* window.





   The check mark next to Montreal indicates the original element.

3. Click Close

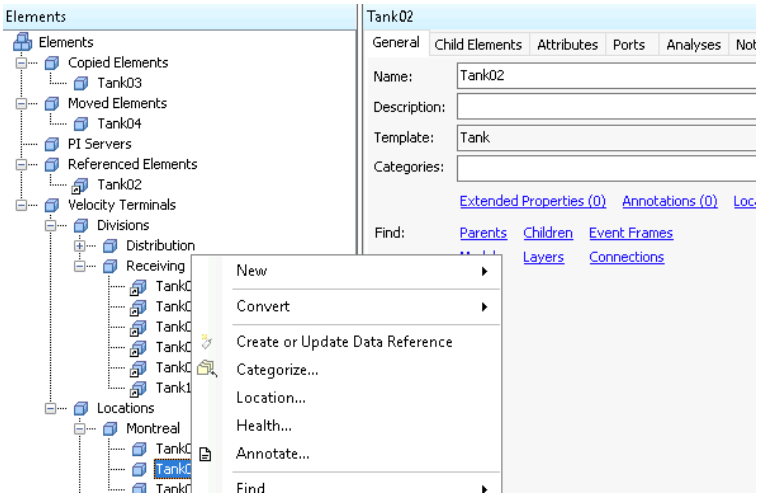
8.3. Annotations

You can add annotations to Elements and Event Frames to document something about them and attach certain document types to the annotation.

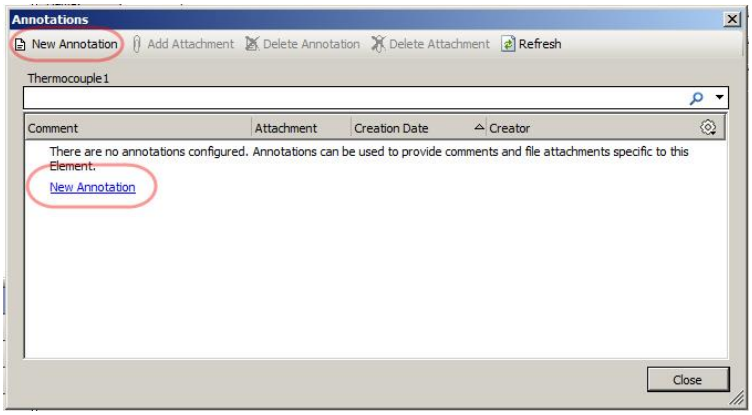
Approach: Add a comment to the **Tank02** Element and attach the **Tank02.pdf** document to the comment.

Step by step:

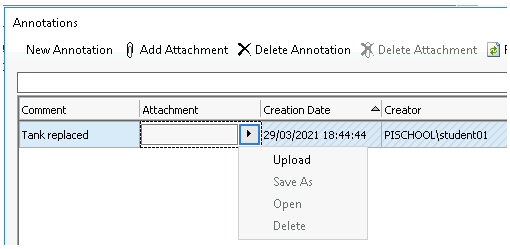
1. Press the **Ctrl+1** key combination to navigate to the **Elements** view and click on the **Tank02** Element.
2. You can either right-click the **Tank02** Element or you can select it and then double-click the **Annotations (0)** in the **Viewing Pane**.



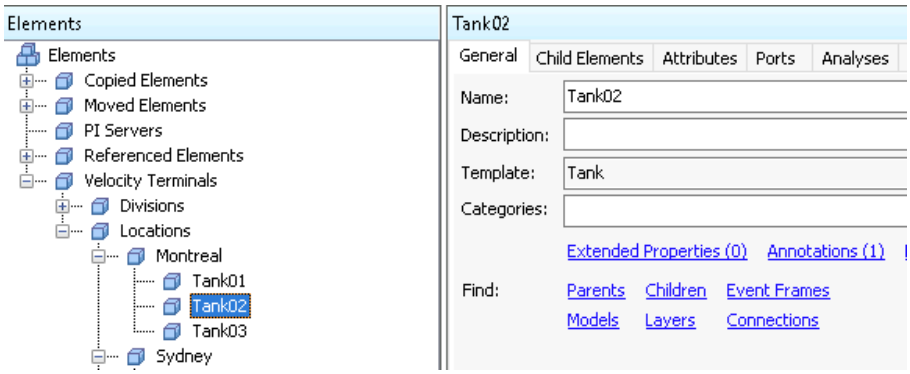
3. In the **Annotations** window click on **New Annotation**.



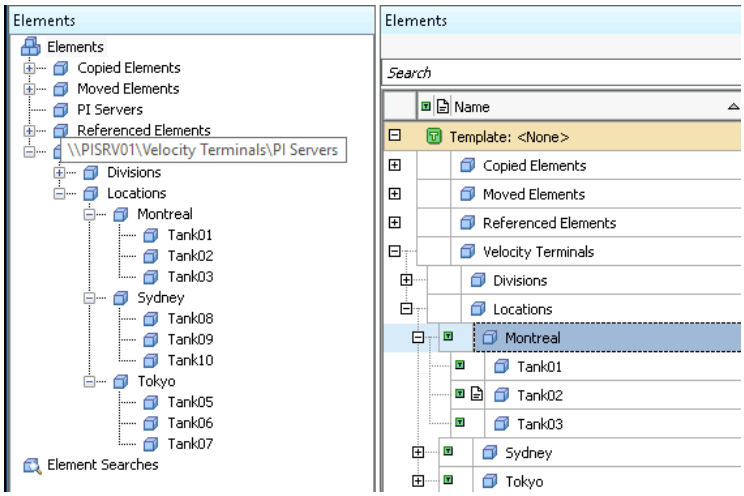
4. In the **Comment** field type in, **Tank replaced.** Click on the **Attachment** field and then on the symbol. Select **Upload**, navigate to the Desktop and select the **Tank02_specs.pdf** file. Then click on **Close** to close the window.



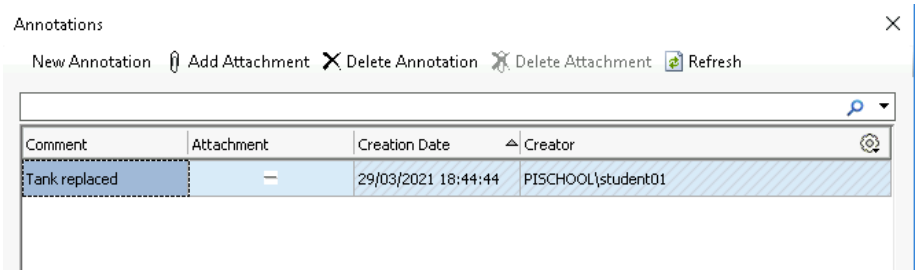
5. Notice that when you select the Element you can see how many annotations that particular Element has in the **General** tab of the **Viewing Pane**. **For Tank02** Element it shows **Annotations (1)**.



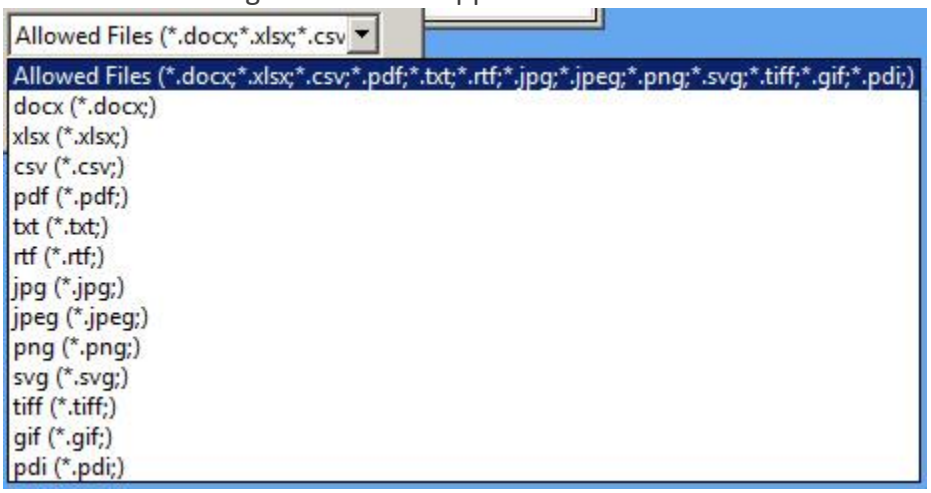
6. Click on **Elements** in the **Browser Pane** and in the **Viewing Pane,** you can see which Elements have annotations by the icon. Hover the mouse cursor over the icon and a tooltip appears with the comment and the name of any attachment.



7. Double-click the icon that you just hovered over. The **Annotations** window opens, select the **Annotation** field, and click on the symbol. Select the **Open** option. The document is opened in Acrobat Reader. This is a good way to attach important instructions and documentation about the asset so that it is available for all users. Close the Acrobat Reader.



Note: The following formats are supported as attachments.



9. Administrator Tasks (optional)

9.1. Customization Options in PI System Explorer

You can customize many aspects of PI System Explorer to suit your needs with Tools > Options.

General tab

You use the General tab to control display options for several features. You can control:

- Keystroke to open **Check-In** and **Undo-Checkout** windows. For more information, see *Check-in of database changes*.
- Title bar appearance.
- Page size for browser objects. For more information, see *Configure browser page size*.
- Number of queries returned in object searches. For more information, see *Search result paging*.
- Unit of measure appearance for attributes. For more information, see *Show attribute values in source unit of measure*.
- Display of attribute values to the units of measure that are mapped to a selected UOM group. When no UOM group is selected, attribute values are based on the default UOM defined in attribute templates or their source unit of measure (if Use Source Unit-Of-Measure for attribute display is selected). For more information, see *UOM groups*.
- Display of excluded attributes. For more information, see *Excluded attribute property*.
- Number of digits displayed for attribute values. For more information, see *Control the display of attribute and attribute template values*.

Time Context tab

You use the Time Context tab to define the time or time range that PI System Explorer uses to display attribute values. For more information, see *Set time context for displayed attribute values*.

Server Options tab

You use the Server Options tab to define how PI System Explorer should connect to a PI AF collective or PI Data Archive collective. For more information, see *Manage connection preferences for PI System Explorer*.

Source: *PI System Explorer customization options*

9.2. Viewing RPC metrics

You can view remote procedure call (RPC) metrics for the current server connection. The RPC Metrics page displays performance data about RPC calls by name, including the number of calls and call duration.

Note: Only users with administrator privileges can view the RPC Metrics page. Beginning with PI AF 2018 SP2, only certain column headers on the RPC Metrics page are visible.

Procedure

1. On the PI System Explorer toolbar, click  .
2. In the PI AF Server Properties window, click the RPC Metrics tab.
3. You can view, sort, and copy data on the RPC Metrics page:
    - To sort data by a particular column heading such as Calls or Total Duration, click the column heading.
    - To copy one row of connection data, right-click the row and then click Copy.
    - To copy all connection data, right-click the row, click Select All and then click Copy.
4. Review information about each call as needed. You can expand the width of the columns as needed.
5. The RPC Metrics page displays the following information:

Column Title	Explanation
RPC Name	The name of the client process executed by the server
Calls	The number of total calls for the individual RPC since the server was last started
Total Duration	The total length of all the calls to the server from a particular client
Per Call	The average length of each call
Calls (Delta)	A count of the number of calls since the list was last displayed or refreshed
Duration (Delta)	The total length of the calls to each RPC since the list was last displayed or refreshed
Per Call (Delta)	The average length of the calls to each RPC that occurred since the list was last displayed or refreshed
Calls (Client)	The number of calls to each RPC that was made by the client retrieving the RPC metrics

Column Title	Explanation
Duration (Client)	The length of the RPC call made by the client retrieving the RPC metrics
Per Call (Client)	The average length of the calls to each RPC that was made by the client retrieving the RPC metrics
Retries	The number of times the client process has been attempted

Note: Use the Refresh button to update the RPC Metrics page with the latest RPC call data.
Source: *View RPC metrics*

9.3. AF SDK Capture

You use the AFGetTrace utility (afgettrace.exe) to capture event trace output from the AF SDK. Event tracing can help you debug an application and perform capacity and performance analysis.

Note: Starting with PI AF 2018, the **AFGetTrace** utility includes a graphical user interface (GUI) that allows you to configure and view event trace sessions. By default, the AFGetTrace utility runs in GUI mode. To run AFGetTrace in the old console mode, use the /NoGUI (/NG) switch.

Procedure

1. Open a command window and change directory to PIPC\AF.
2. Choose from the following actions:

To ...	Do this ...
Display syntax and parameters	At the command prompt, type: afgettrace /?
Run AFGetTrace with default settings	At the command prompt, type: afgettrace Default output goes to standard output.
Run AFGetTrace with specific parameters	At the command prompt, type: afgettrace /*parameter* Refer to *AFGetTrace utility parameters* for details on the parameters you can use.
Terminate event tracing	In the command window, type: X **Note:** If you close the command window without terminating afgettrace, trace events continue to be generated, which can slow down your AF SDK applications.

Source: *Capture AF SDK event trace output*

9.4. Audit Trail Functionality

The Audit Trail feature allows you to review changes to PI AF objects, as well as certain system and security settings. You must have administrative privileges to turn on or off Audit Trail and view audit information. See *Audit Trail implementation* for instructions on how to enable Audit Trail.

Note: Audit Trail requires the use of SQL Server Change Data Capture (CDC), which is a feature of Microsoft SQL Server. Audit Trail does not capture changes related to enabling or disabling the CDC table.

What is audited?

The following PI AF objects are tracked when Audit Trail is enabled:

- Element objects:
    - Databases
    - Elements
    - Static attribute values that are not data references
    - Analyses
    - Notifications
    - Models
- Event frame objects:
    - Event frames
    - Transfers
    - Cases
- Library objects:
    - Templates
    - Enumeration sets
    - Reference types
    - Tables and table connections
    - Categories

What user actions are audited?

The following user actions are recorded when Audit Trail is enabled:

- When an AF object is added, modified, or deleted. For example, when an object is renamed or when its description is changed.
- Changes to the security rights for a system collection, a database collection or an individual object.
- Changes to the UOM database. For example, adding a UOM.
- Changes to the AF Server. For example, turning on the Audit Trail feature or removing a plug-in.
- When the Audit Trail feature is turned on
- Changes to an AF security certificate (when a certificate is added, modified or deleted)
- When an AF plug-in is added or deleted
- When a trusted AF plug-in provider is added or deleted
- When an AF database is deleted
- When analyses, legacy notifications or notification rules are enabled or disabled

To enable the audit trail,

- administration level privileges are required on the AF Server
- sysadmin privileges to the AF database (PIFD)
- working knowledge of PI System administration

To enable the audit trail, use the AFDiag Utility (from the command window with elevated privileges and navigate to Program Files\PIPC\AF) and EnableAuditTrail (/AT) parameter: Afdiag /AT

To disable the audit trail, use the /ATD parameter. Upon disabling the audit trail, the collected audit trail records are permanently deleted and not recoverable.

Once enabled, with sufficient privileges, can right-click an object in the browser or an object on a list in the viewer and click **Audit Trail Events**. This allows review audit data specific to the selected object only. Each row in the table of the AF Audit Trail window contains data that identifies a specific change to an AF object. Double clicking the row will display details about the change(s). The details of the change can be copied (CTRL + C) and pasted (CTRL + V) from the window into a document.
There is an *Audit Trail utility* which can be accessed by individuals with administrator privileges to the AF Server.

Source: *Overview of Audit Trail*

10. Final Exam

The final exam in this course is taken online. Please check the course listing online for more details.