Guide to Substitution Syntax for Data References in PI AF and PI Event Frames

Version 2

October 2015

Table of Contents

Revision History

Version	Date	Comments	Author
Version 1	1/7/2015	Original draft	Ales Soudek
Version 2	9/29/2015	Updated for AF 2015 (AF 2.7) - String Builder use in Event Frames - Additional syntax for Table Lookup	Ales Soudek

Introduction

Purpose

The substitution parameters in PI Analysis Framework (PI AF) and PI Event Frames (PI EF) are very powerful for creating a PI AF structure that can leverage the power of the templates and is more maintainable. Since I could not find a user-focused document, which shows practical examples of the substitution syntax and examples for the different Data References in PI AF, I decided to compile one. This document also covers the special syntax used for PI EF.

This document discusses the following:

- PI Point
- Formula
- Table Lookup
- String Builder
- Event Frames

PI Point Data Reference

The PI Data Reference in PI AF requires the user to enter either a **Tag name** or an **Attribute** reference that has a PI Data Reference defined. Details of these approaches are below.

Tag Name

If the site has adhered to a tag naming convention, then a substitution string can be set up in the Attribute of the Element Template. This is the **best** scenario and **least** amount of work to maintain and rollout (see **Mapping Using Substitution Strings** Section below). Unfortunately, in most cases this is not possible since either there are too many different naming conventions or there are no naming conventions at all. (See **Mapping Using a Table** and **Using AF Builder** Sections below)

It is up to the reader to decide which approach they prefer and best fits their situation.

Mapping Using Substitution Strings

The syntax is as follows:

- 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. More possible substitution entities are found in the PI AF help file under PI data reference or some selected examples are shown in **Table 1** below.
- **|** symbol means the string following is an Attribute, if the Attribute is at the same level as the current Attribute, you do not need the | symbol. For use in the **Tag Name** this would be combined with the use of the @ symbol. See the below.

- @ symbol means return the value of the Attribute. For example, **%@.|PITagName%** means replace with the value of the child Attribute called **PITagName**.
- **..\** symbol means go to the parent of the current Element. For example, **%..\Element%** means replace with the name of the parent Element.

Examples:

The following shows examples of using the substitution syntax for PI tag naming:

- Default setting
- Getting value of another Attribute
- Using Element hierarchy
- Substituting Server name

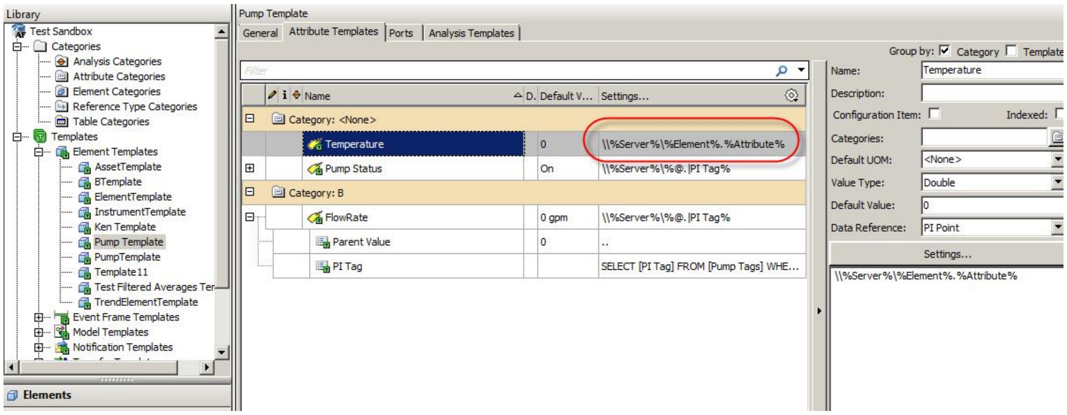
Syntax Examples

Default Setting

%Element%.%Attribute%

Example:

Figure 1 shows the default setting when creating a PI Point Data Reference. The substitution string is ***%Element%.%Attribute%*** . The substitution will use the name of the current Element and the name of the current Attribute. In this example, Element name is **P-101** and the Attribute name is **Temperature**. The resulting PI tag name is **P-101.Temperature**, see Figure 2.

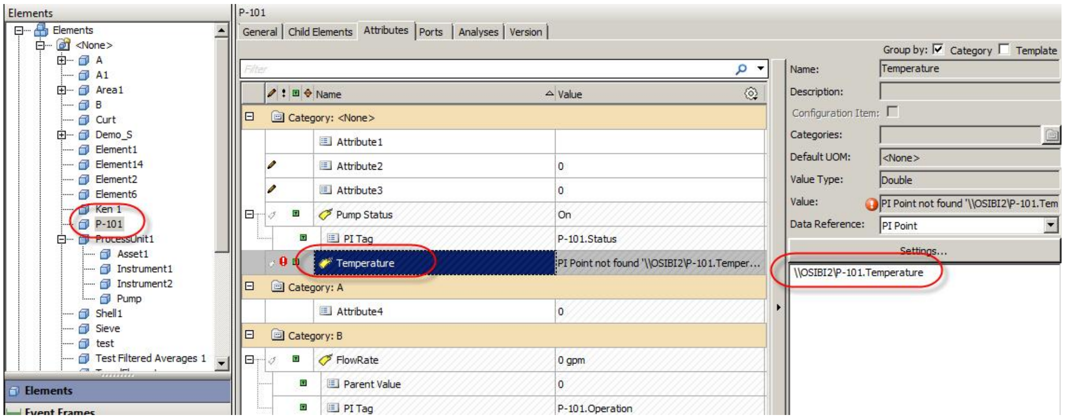


Group by: ☑ Category □ Template

Name	D.	Default V...	Settings...
Category: <None>			
Temperature		0	\\%Server%\%Element%.%Attribute%
Pump Status		On	\\%Server%\%@.|PI Tag%
Category: B			
FlowRate		0 gpm	\\%Server%\%@.|PI Tag%
Parent Value		0	..
PI Tag			SELECT [PI Tag] FROM [Pump Tags] WHE...

Name: Temperature
Description:
Configuration Item: □ Indexed: □
Categories:
Default UOM: <None>
Value Type: Double
Default Value: 0
Data Reference: PI Point
Settings...
\\%Server%\%Element%.%Attribute%

Figure 1



Name	Value
Category: <None>	
Attribute1	
Attribute2	0
Attribute3	0
Pump Status	On
PI Tag	P-101.Status
Temperature	PI Point not found '\\OSIBI2\P-101.Temper...
Category: A	
Attribute4	0
Category: B	
FlowRate	0 gpm
Parent Value	0
PI Tag	P-101.Operation

Name:	Temperature
Description:	
Configuration Item:	□
Categories:	
Default UOM:	<None>
Value Type:	Double
Value:	PI Point not found '\\OSIBI2\P-101.Tem
Data Reference:	PI Point

\\OSIBI2\P-101.Temperature

Figure 2

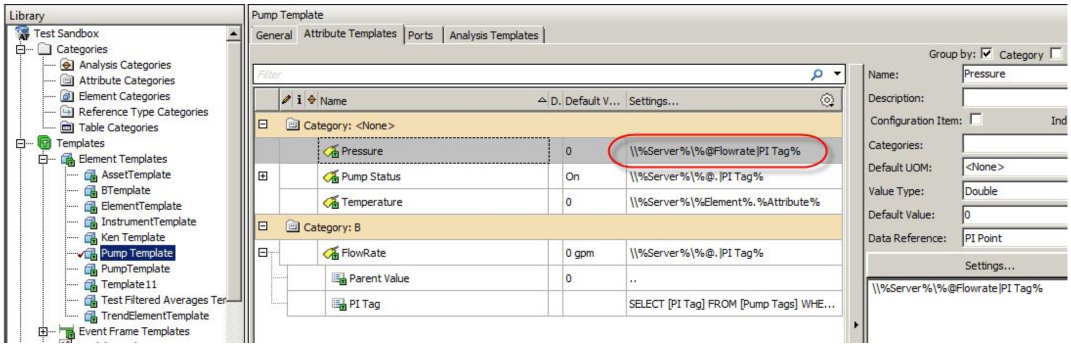
Getting a value from another Attribute

%@<Attribute Name>|<Child Attribute Name>%

Example:

You can also use the substitution parameters to get the value of a particular Attribute. The example below, in Figure 3, shows the syntax for getting a value from a child Attribute in the same Element. The substitution string is ***%@Flowrate|PI Tag%*** *.* This means take the value of the child Attribute **PI Tag** of the Attribute **Flowrate**. Figure 4 shows the resulting substitution.

The value of the child Attribute **PI Tag** of the Attribute **Flowrate** is **P-101.Operation**, therefore this is the name of the PI Tag that gets substituted.

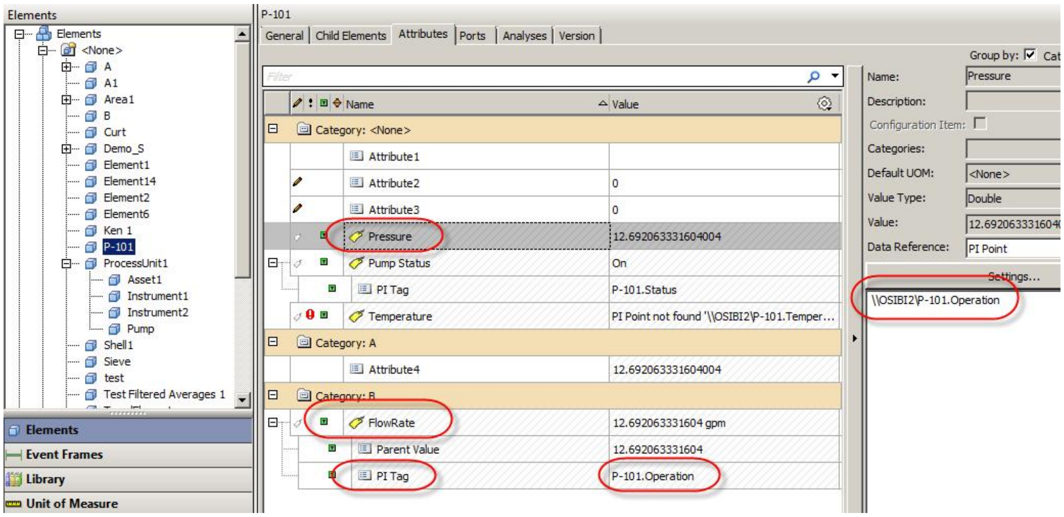


Name	D.	Default V...	Settings...
Category: <None>			
Pressure		0	\\%Server%\%@Flowrate|PI Tag%
Pump Status		On	\\%Server%\%@.|PI Tag%
Temperature		0	\\%Server%\%Element%.%Attribute%
Category: B			
FlowRate		0 gpm	\\%Server%\%@.|PI Tag%
Parent Value		0	..
PI Tag			SELECT [PI Tag] FROM [Pump Tags] WHE...

Name:	Pressure
Description:	
Configuration Item:	□
Categories:	
Default UOM:	<None>
Value Type:	Double
Default Value:	0
Data Reference:	PI Point

\\%Server%\%@Flowrate|PI Tag%

Figure 3



Name	Value
Category: <None>	
Attribute1	
Attribute2	0
Attribute3	0
Pressure	12.692063331604004
Pump Status	On
PI Tag	P-101.Status
Temperature	PI Point not found '\\OSIBI2\P-101.Temper...
Category: A	
Attribute4	12.692063331604004
Category: B	
FlowRate	12.692063331604 gpm
Parent Value	12.692063331604
PI Tag	P-101.Operation

Name: Pressure
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 12.69206333160400
Data Reference: PI Point
\\OSIBI2\P-101.Operation

Figure 4

An alternate syntax to achieve the same result as above is to use following **%@.|PI Tag%** for the Attribute **Flowrate**. The **.** means current object, which is the Attribute, and the **|** symbol means Attribute. So since the current object is an Attribute, the syntax says replace with the value of the child Attribute **PI Tag** of the current Attribute, in the example above, Figure 4, this would be the **FlowRate**.

Using Element hierarchy

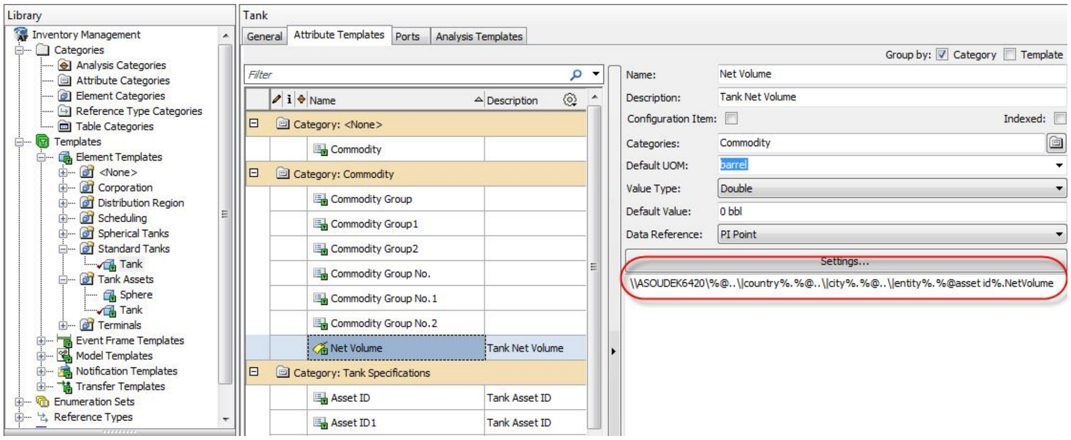
%@..\..\|<Attribute Name>%.%@..\|<Attribute Name>%

Example:

This example shows how to use values of Attributes from the Parent Element in the naming of the PI Tag. Figure 5 shows the substitution string syntax. In this example, the substitution string is ***%@..\|country%.%@..\|city%.%@..\|entity%.%@asset id%.NetVolume***

Taking the first part of the string, ***%@..\|country%*** , the @ symbol means take the value of, the **..** symbol means go up one level in the Element hierarchy, the **|** symbol means Attribute, and **country** is the name of the Attribute. The parent Element is **Brasil 2** (see Figures 6 and 7), which has an Attribute named **country**, whose value is **Brasil**, therefore the substitution string ***%@..\|country%*** is substituted with **Brasil**. Similarly for the ***%@..\|city%*** and ***%@..\|entity%*** substitution strings. The final part of the substitution string ***%@asset id%.NetVolume*** use the value of the Attribute **asset id** in the current Element and append the **.NetVolume** string. The final substitution result is shown in Figure 6, which is

Brasil.Porto Alegre.MPC Branch LP Terminal.1-7.NetVolume

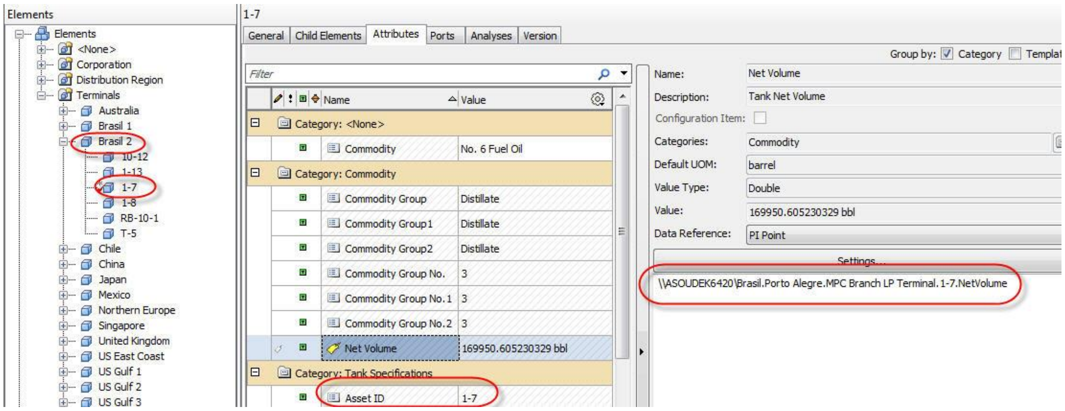


Group by: ☑ Category □ Template

Name	Description
Category: <None>	
Commodity	
Category: Commodity	
Commodity Group	
Commodity Group1	
Commodity Group2	
Commodity Group No.	
Commodity Group No.1	
Commodity Group No.2	
Net Volume	Tank Net Volume
Category: Tank Specifications	
Asset ID	Tank Asset ID
Asset ID1	Tank Asset ID

Name: Net Volume
Description: Tank Net Volume
Configuration Item: □ Indexed: □
Categories: Commodity
Default UOM: barrel
Value Type: Double
Default Value: 0 bbl
Data Reference: PI Point
Settings...
\\ASOUDEK6420\%@..\|country%.%@..\|city%.%@..\|entity%.%@asset id%.NetVolume

Figure 5

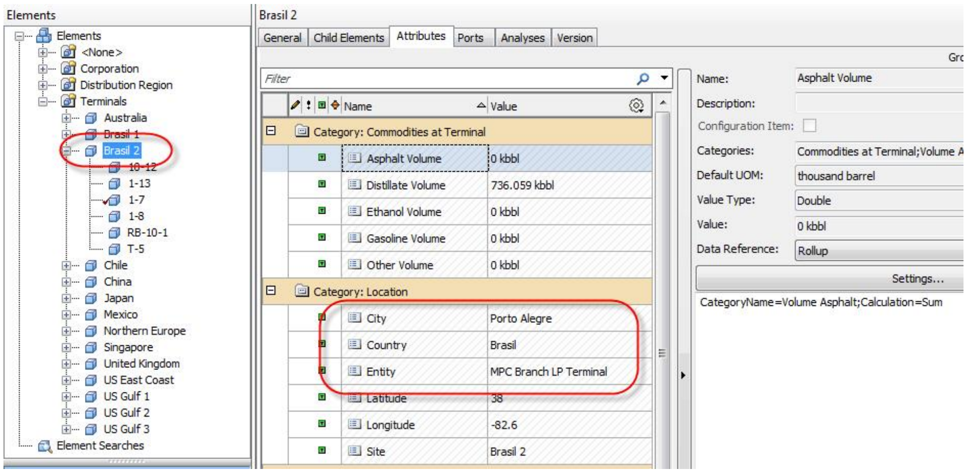


Group by: ☑ Category □ Templat

Name	Value
Category: <None>	
Commodity	No. 6 Fuel Oil
Category: Commodity	
Commodity Group	Distillate
Commodity Group1	Distillate
Commodity Group2	Distillate
Commodity Group No.	3
Commodity Group No.1	3
Commodity Group No.2	3
Net Volume	169950.605230329 bbl
Category: Tank Specifications	
Asset ID	1-7

Name: Net Volume
Description: Tank Net Volume
Configuration Item: □
Categories: Commodity
Default UOM: barrel
Value Type: Double
Value: 169950.605230329 bbl
Data Reference: PI Point
Settings...
\\ASOUDEK6420\Brasil.Porto Alegre.MPC Branch LP Terminal.1-7.NetVolume

Figure 6



Name	Value
Category: Commodities at Terminal	
Asphalt Volume	0 kbbl
Distillate Volume	736.059 kbbl
Ethanol Volume	0 kbbl
Gasoline Volume	0 kbbl
Other Volume	0 kbbl
Category: Location	
City	Porto Alegre
Country	Brasil
Entity	MPC Branch LP Terminal
Latitude	38
Longitude	-82.6
Site	Brasil 2

Name: Asphalt Volume
Description:
Configuration Item: □
Categories: Commodities at Terminal;Volume A
Default UOM: thousand barrel
Value Type: Double
Value: 0 kbbl
Data Reference: Rollup
Settings...
CategoryName=Volume Asphalt;Calculation=Sum

Figure 7

Substituting Server name

%Server%

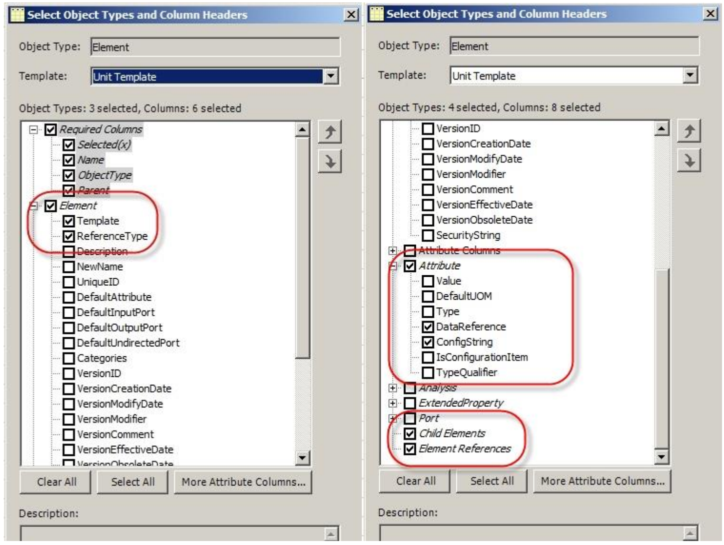
Example:

When configuring the PI Point Data Reference the default setting for the PI Server name is **%Server%**. This will be substituted by the current PI Archive Server name.

Note: The problem with this approach is that if the AF database is exported and then imported to a new system with a new PI Archive Server, the old PI Archive Server name will still be in the PI Point Data Reference settings of the Attributes. So one of my maxims - never use default settings, always change them explicitly, because you never know when they might cause you headaches.

If you encounter this situation, the following approaches will place the correct PI Archive Server name when you move the PI AF to a new system:

- **Method 1** - Import all the Elements into an Excel Worksheet using **PI Builder**. In the settings, make sure to select the **DataReference** and **ConfigString** options. Do a global replace on the old server name with the new server name. Then export the Elements to the PI AF database. (See PI Builder User Manual). You will need the following settings in **PI Builder**, see Figure 8 below.



Select Object Types and Column Headers

Object Type: Element

Template: Unit Template

Object Types: 3 selected, Columns: 6 selected

- ☑ Required Columns
  - ☑ Selected(x)
  - ☑ Name
  - ☑ ObjectType
  - ☑ Parent
- ☑ Element
  - ☑ Template
  - ☑ ReferenceType
  - ☐ Description
  - ☐ NewName
  - ☐ UniqueID
  - ☐ DefaultAttribute
  - ☐ DefaultInputPort
  - ☐ DefaultOutputPort
  - ☐ DefaultUndirectedPort
  - ☐ Categories
  - ☐ VersionID
  - ☐ VersionCreationDate
  - ☐ VersionModifyDate
  - ☐ VersionModifier
  - ☐ VersionComment
  - ☐ VersionEffectiveDate
  - ☐ VersionObsoleteDate

Select Object Types and Column Headers

Object Type: Element

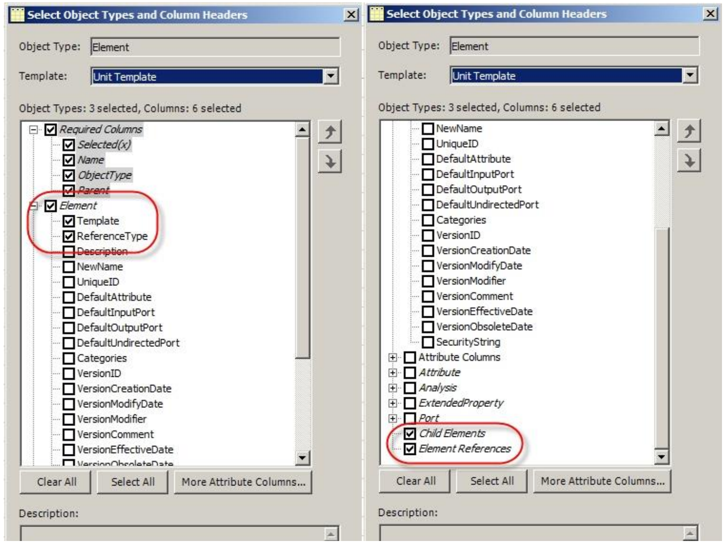
Template: Unit Template

Object Types: 4 selected, Columns: 8 selected

- ☐ VersionID
- ☐ VersionCreationDate
- ☐ VersionModifyDate
- ☐ VersionModifier
- ☐ VersionComment
- ☐ VersionEffectiveDate
- ☐ VersionObsoleteDate
- ☐ SecurityString
- ☐ Attribute Columns
- ☑ Attribute
  - ☐ Value
  - ☐ DefaultUOM
  - ☐ Type
  - ☑ DataReference
  - ☑ ConfigString
  - ☐ IsConfigurationItem
  - ☐ TypeQualifier
- ☐ Analysis
- ☐ ExtendedProperty
- ☐ Port
- ☑ Child Elements
- ☑ Element References

Figure 8

- **Method 2** - **If the PI AF database has already been moved to the new server**. Import all the Elements into an Excel Worksheet using **PI Builder** as above but excluding the DataReference and ConfigString options. Then using **PI Builder** delete the Elements in the AF database (**Note** – only do this if all the Elements have an Element Template, and you have not modified any values in the Element Attributes outside of the Element Templates). Then recreate the Elements using **PI Builder**. You will need the following settings in **PI Builder**, see Figure 9 below.



Select Object Types and Column Headers

Object Type: Element

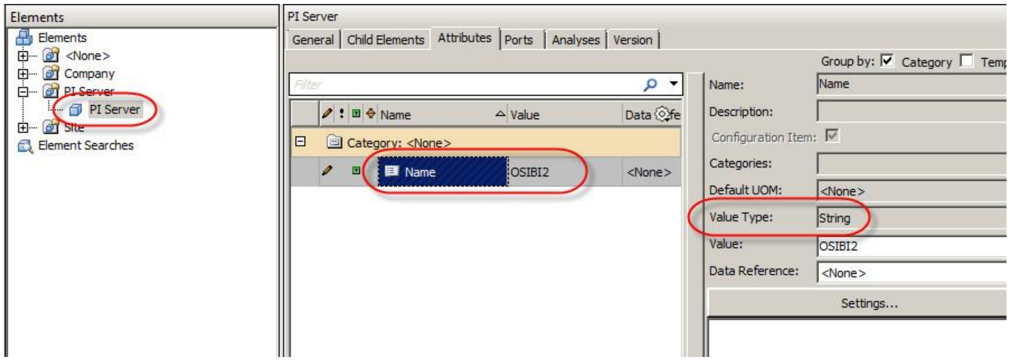
Template: Unit Template

Object Types: 3 selected, Columns: 6 selected

- ☑ *Required Columns*
  - ☑ *Selected(x)*
  - ☑ *Name*
  - ☑ *ObjectType*
  - ☑ *Parent*
- ☑ *Element*
  - ☑ Template
  - ☑ ReferenceType
  - ☐ Description
  - ☐ NewName
  - ☐ UniqueID
  - ☐ DefaultAttribute
  - ☐ DefaultInputPort
  - ☐ DefaultOutputPort
  - ☐ DefaultUndirectedPort
  - ☐ Categories
  - ☐ VersionID
  - ☐ VersionCreationDate
  - ☐ VersionModifyDate
  - ☐ VersionModifier
  - ☐ VersionComment
  - ☐ VersionEffectiveDate
  - ☐ VersionObsoleteDate
  - ☐ SecurityString
- ☐ Attribute Columns
- ☐ *Attribute*
- ☐ *Analysis*
- ☐ *ExtendedProperty*
- ☐ *Port*
- ☑ *Child Elements*
- ☑ *Element References*

Figure 9

- **Method 3** - Set up an Element and Attribute that will contain name of the PI Archive Server (Also remember to convert this Element to a Template. Another one of my maxims – every Element should be associated with a Template). This is my preferred approach. The PI Point Data References in the Attributes are then set up with a different substitution string to the default ***%Server%***. (Remember my maxim from above – never use default settings. See the example below). Then to change the server name in all the PI Point Data References simply change the name in this Attribute with the appropriate PI Archive Server name. Note: this only works for AF 2.6 and later. The example in Figure 10 shows the Element named **PI Server** with an Attribute called **Name** which is of a string value type. The value of the **Name** Attribute is the name of the PI Archive Server.



Group by: ☑ Category ☐ Temp

Name	Value	Data
Category: <None>		
Name	OSIBI2	<None>

Name: Name

Description:

Configuration Item: ☑

Categories:

Default UOM: <None>

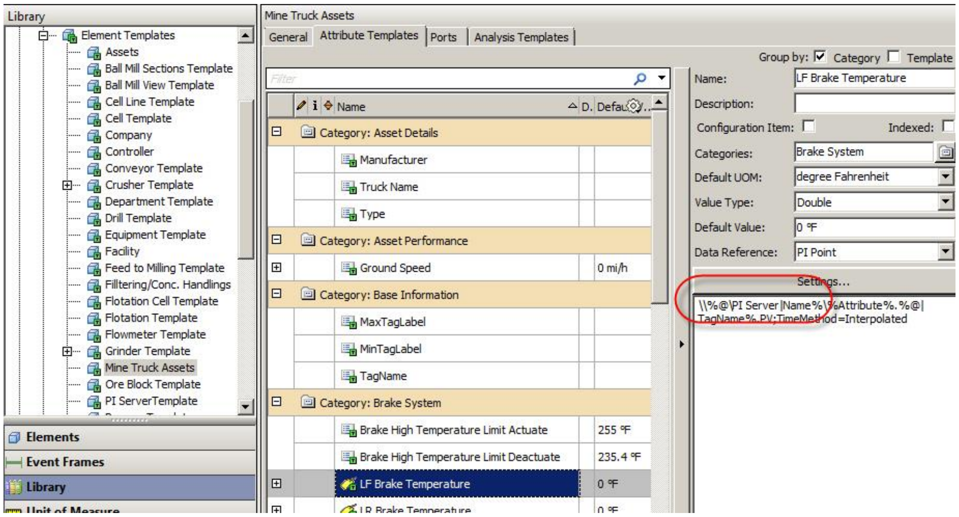
Value Type: String

Value: OSIBI2

Data Reference: <None>

Figure 10

The syntax for the substitution string in the PI Point Data Reference, shown in Figure 11, is **%@\PI Server|Name%** . The @ symbol means take the value of, the **\PI Server** means from top of the Element hierarchy use the Element named **PI Server**, the **|Name** means use the Attribute called **Name**.



Name	Default
Category: Asset Details	
Manufacturer	
Truck Name	
Type	
Category: Asset Performance	
Ground Speed	0 mi/h
Category: Base Information	
MaxTagLabel	
MinTagLabel	
TagName	
Category: Brake System	
Brake High Temperature Limit Actuate	255 °F
Brake High Temperature Limit Deactuate	235.4 °F
LF Brake Temperature	0 °F
LR Brake Temperature	0 °F

Group by: ☑ Category □ Template

Name: LF Brake Temperature

Description:

Configuration Item: □ Indexed: □

Categories: Brake System

Default UOM: degree Fahrenheit

Value Type: Double

Default Value: 0 °F

Data Reference: PI Point

```
\\%@\PI Server|Name%\%Attribute%.%@|TagName%.PV;TimeMethod=Interpolated
```

Figure 11

Table of useful substitution parameters for naming PI tags:

Substitution Parameter	Description
%Attribute%	Replaced with the name of the current Attribute
%..|Attribute%	Replaced with the name of the parent Attribute of the current Attribute containing the PI Point Data Reference
%Element%	Replaced with the name of the current Element
%..\Element%	Replaced with the name of the parent Element of the current Element
%Description%	Replaced with the description of the current Attribute
%ParentAttributeDescription%	Replaced with the description of the parent Attribute of the current Attribute containing the PI Point Data Reference
%ElementDescription%	Replaced with the description of the current Element
%ParentElementDescription%	Replaced with the description of the parent Element of the current Element
%Server%	Replaced with the current default PI Server

Mapping in Using a Table

This approach requires the creation of an Attribute/PI tag naming mapping table in PI AF and then using two Attributes for each value – one to lookup the name of the PI tag from this table and the second one to use the value of the first Attribute in a substitution string for the PI Point Data Reference. The benefit of this approach is that this can be templatized and all the tag naming is managed in a table that is part of the AF database. The table can be created externally and then imported or linked to PI AF.

There are a couple of **negatives** to consider with this approach.

1. The Tables in PI AF are not recommended to have more than about 10,000 rows. Performance may be adversely affected as a Table grows bigger.
2. The user needs to maintain the mappings in the Table(s)

Syntax Examples

%@.|<Attribute Name>%

Example:

The example below can be set up in different ways, but the general syntax is the same.

Steps (the naming for Attributes and Elements in the steps relate to the example in the Figures below

1. In the Template, create an Attribute, **Power Draw** and a child Attribute **TagName**, and set the Date Reference to PI Point for the parent Attribute. The setting for the tag name is **%@.|TagName%**

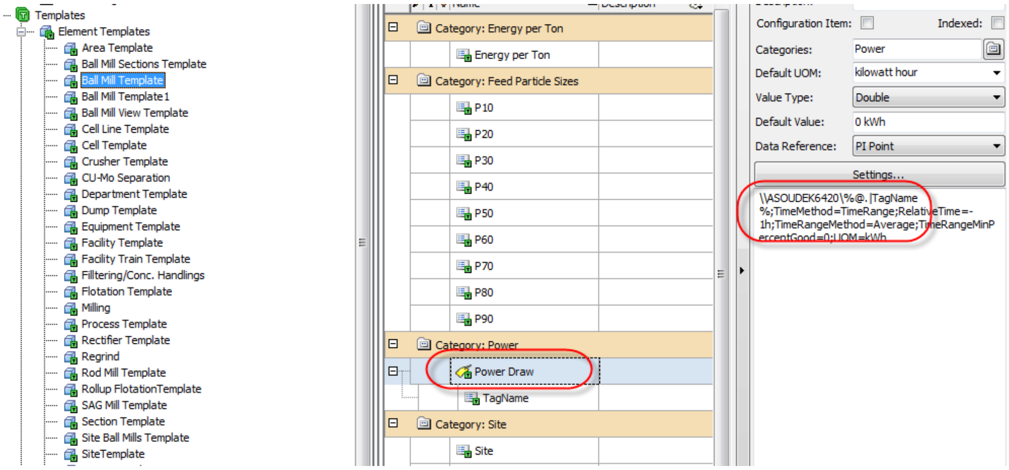
   The syntax meaning of the above substitution string is as follows. Substitute the value of what is between the **%** and **%** symbols. The @ means use the value in the Attribute named **Power Draw|TagName**, where the **Power Draw** is the parent Attribute and **TagName** is the child Attribute. See Figure 12.

2. Set the Data Reference for the child **TagName** to Table Lookup. The setting for the config string is

   ParentName = '%..\..\..\..\Element%' AND ElementName = '%Element%' AND AttributeName = '%..|Attribute%'

   The syntax meaning of the above string is as follows. Compare the ParentName column to the Element name 4 levels up the hierarchy, and the ElementName column with the Element containing the Attribute **Power Draw**, and the AttributeName column with the parent Attribute name (in this case **Power Draw** which is the parent Attribute of the **TagName** child-Attribute). See Figure 13.

3. Figure 14 shows the table details to support this structure
4. Figure 15 shows the resulting Element hierarchy and substitution results for the Attribute



Configuration Item: □ Indexed: □

Categories: Power

Default UOM: kilowatt hour

Value Type: Double

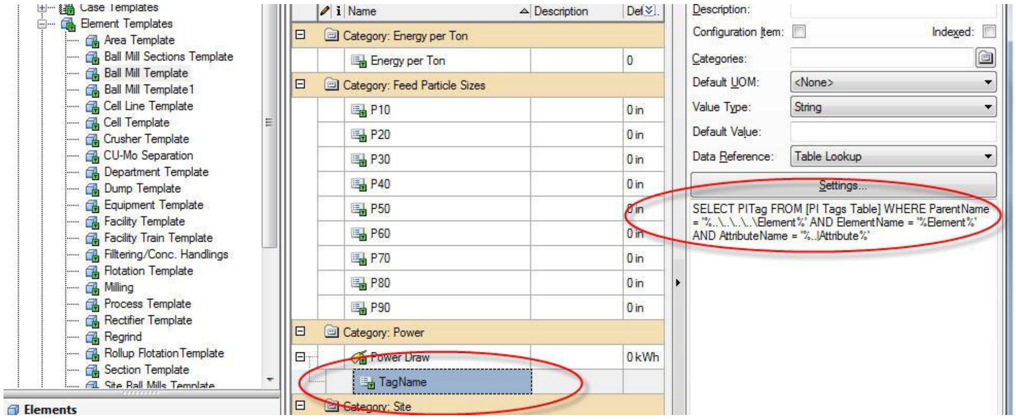
Default Value: 0 kWh

Data Reference: PI Point

Settings...

```
\\ASOUDEK6420\%@.|TagName
%;TimeMethod=TimeRange;RelativeTime=-
1h;TimeRangeMethod=Average;TimeRangeMinP
ercentGood=0;UOM=kWh
```

Figure 12



Description:

Configuration Item: □ Indexed: □

Categories:

Default UOM: <None>

Value Type: String

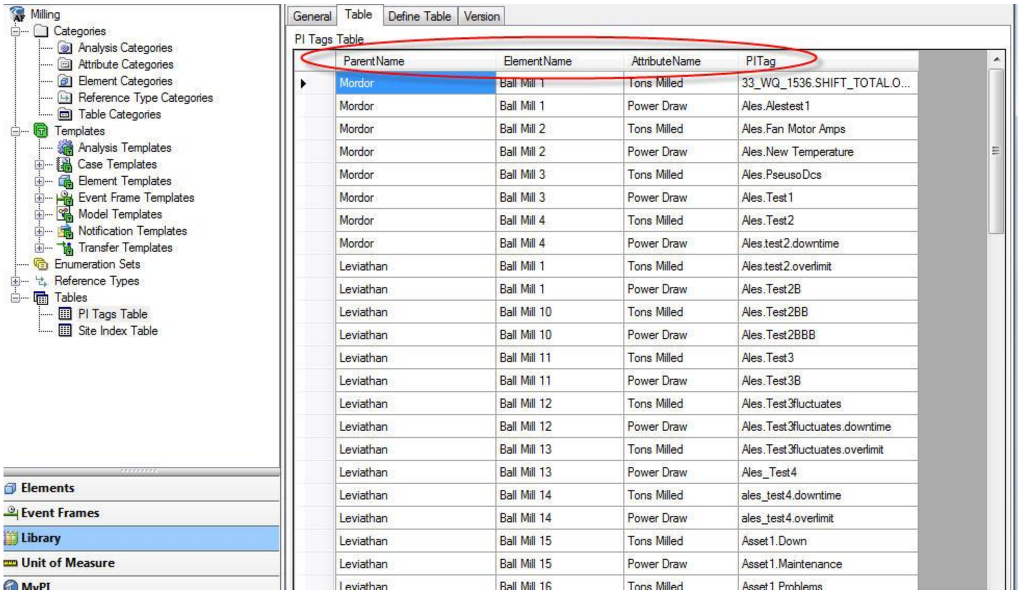
Default Value:

Data Reference: Table Lookup

Settings...

```
SELECT PITag FROM [PI Tags Table] WHERE ParentName
= '%..\..\..\..\Element%' AND ElementName = '%Element%'
AND AttributeName = '%..|Attribute%'
```

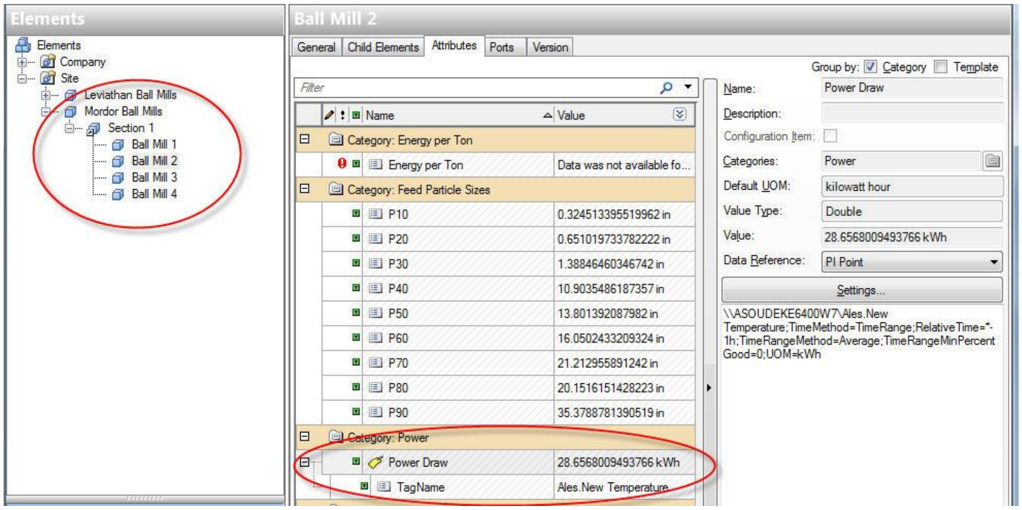
Figure 13



PI Tags Table

ParentName	ElementName	AttributeName	PITag
Mordor	Ball Mill 1	Tons Milled	33_WQ_1536.SHIFT_TOTAL.O...
Mordor	Ball Mill 1	Power Draw	Ales.Alestest1
Mordor	Ball Mill 2	Tons Milled	Ales.Fan Motor Amps
Mordor	Ball Mill 2	Power Draw	Ales.New Temperature
Mordor	Ball Mill 3	Tons Milled	Ales.PseusoDcs
Mordor	Ball Mill 3	Power Draw	Ales.Test1
Mordor	Ball Mill 4	Tons Milled	Ales.Test2
Mordor	Ball Mill 4	Power Draw	Ales.test2.downtime
Leviathan	Ball Mill 1	Tons Milled	Ales.test2.overlimit
Leviathan	Ball Mill 1	Power Draw	Ales.Test2B
Leviathan	Ball Mill 10	Tons Milled	Ales.Test2BB
Leviathan	Ball Mill 10	Power Draw	Ales.Test2BBB
Leviathan	Ball Mill 11	Tons Milled	Ales.Test3
Leviathan	Ball Mill 11	Power Draw	Ales.Test3B
Leviathan	Ball Mill 12	Tons Milled	Ales.Test3fluctuates
Leviathan	Ball Mill 12	Power Draw	Ales.Test3fluctuates.downtime
Leviathan	Ball Mill 13	Tons Milled	Ales.Test3fluctuates.overlimit
Leviathan	Ball Mill 13	Power Draw	Ales_Test4
Leviathan	Ball Mill 14	Tons Milled	ales_test4.downtime
Leviathan	Ball Mill 14	Power Draw	ales_test4.overlimit
Leviathan	Ball Mill 15	Tons Milled	Asset1.Down
Leviathan	Ball Mill 15	Power Draw	Asset1.Maintenance
Leviathan	Ball Mill 16	Tons Milled	Asset1.Problems

Figure 14



Name	Value
Category: Energy per Ton	
Energy per Ton	Data was not available fo...
Category: Feed Particle Sizes	
P10	0.324513395519962 in
P20	0.651019733782222 in
P30	1.38846460346742 in
P40	10.9035486187357 in
P50	13.801392087982 in
P60	16.0502433209324 in
P70	21.212955891242 in
P80	20.1516151428223 in
P90	35.3788781390519 in
Category: Power	
Power Draw	28.6568009493766 kWh
TagName	Ales.New Temperature

Name: Power Draw
Description:
Configuration Item: □
Categories: Power
Default UOM: kilowatt hour
Value Type: Double
Value: 28.6568009493766 kWh
Data Reference: PI Point

```
\\ASOUDEKE6400W7\Ales.New
Temperature;TimeMethod=TimeRange;RelativeTime=*-
1h;TimeRangeMethod=Average;TimeRangeMinPercent
Good=0;UOM=kWh
```

Figure 15

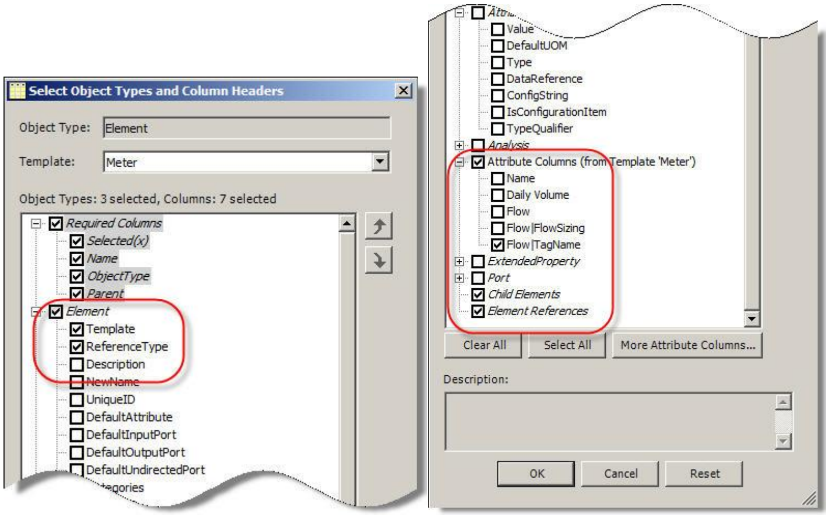
Using PI Builder

This approach is identical to the one in the Section above, Mapping Using a Table, with the exception that the child Attribute does not have a Data Reference. The values for the child Attribute are populated using Excel and the **PI Builder**. The advantage of this is that you can populate the names of the PI tags in Excel and you will not break the Element inheritance from the Template. If you were to enter the PI tag names directly into the Element Attribute **PI Point** Data Reference, then the inheritance from the Template is broken. Another one of my maxims – never break Template inheritance if you can help it. It will make any changes to configuration much easier later.

Syntax Examples

Example:

This example uses a **Meter** Template and a parent Attribute **Flow** which has a child Attribute **TagName** to illustrate the use of the PI Builder to set the child Attribute **TagName** values for all the meter Elements. In **PI Builder**, select the Find Elements in the menu, select the **Meter** Template, and then select only the check boxes shown in Figure 16. This will provide a simple list of all meter Elements and only the child Attribute **Flow|TagName** for entering the PI tag names.



Select Object Types and Column Headers

Object Type: Element

Template: Meter

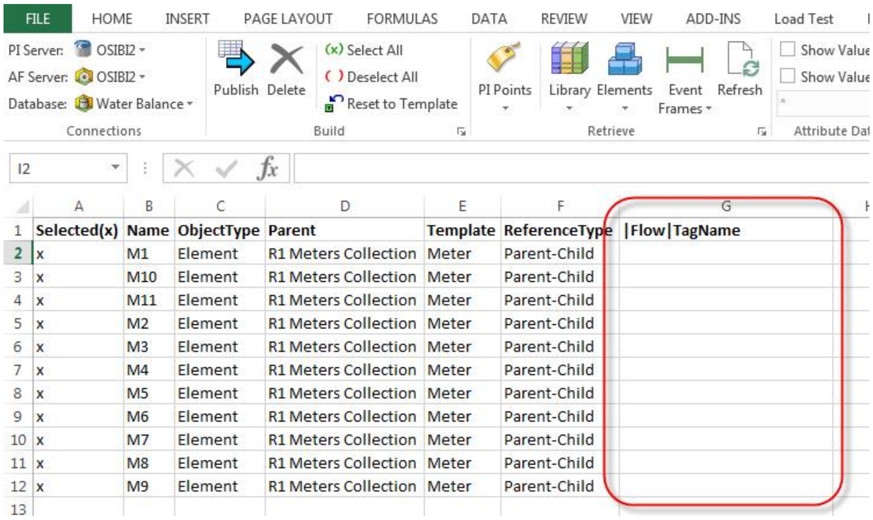
Object Types: 3 selected, Columns: 7 selected

- ☑ *Required Columns*
  - ☑ *Selected(x)*
  - ☑ *Name*
  - ☑ *ObjectType*
  - ☑ *Parent*
- ☑ *Element*
  - ☑ Template
  - ☑ ReferenceType
  - □ Description
  - □ NewName
  - □ UniqueID
  - □ DefaultAttribute
  - □ DefaultInputPort
  - □ DefaultOutputPort
  - □ DefaultUndirectedPort
  - ...tegories

- □ *Attri...*
  - □ Value
  - □ DefaultUOM
  - □ Type
  - □ DataReference
  - □ ConfigString
  - □ IsConfigurationItem
  - □ TypeQualifier
- □ *Analysis*
- ☑ Attribute Columns (from Template 'Meter')
  - □ Name
  - □ Daily Volume
  - □ Flow
  - □ Flow|FlowSizing
  - ☑ Flow|TagName
- □ *ExtendedProperty*
- □ *Port*
- ☑ *Child Elements*
- ☑ *Element References*

Figure 16

The result will be as shown in Figure 17. Then enter the PI tag names in the highlighted area and click Publish Publish button in the ribbon.



	A	B	C	D	E	F	G
1	Selected(x)	Name	ObjectType	Parent	Template	ReferenceType	|Flow|TagName
2	x	M1	Element	R1 Meters Collection	Meter	Parent-Child	
3	x	M10	Element	R1 Meters Collection	Meter	Parent-Child	
4	x	M11	Element	R1 Meters Collection	Meter	Parent-Child	
5	x	M2	Element	R1 Meters Collection	Meter	Parent-Child	
6	x	M3	Element	R1 Meters Collection	Meter	Parent-Child	
7	x	M4	Element	R1 Meters Collection	Meter	Parent-Child	
8	x	M5	Element	R1 Meters Collection	Meter	Parent-Child	
9	x	M6	Element	R1 Meters Collection	Meter	Parent-Child	
10	x	M7	Element	R1 Meters Collection	Meter	Parent-Child	
11	x	M8	Element	R1 Meters Collection	Meter	Parent-Child	
12	x	M9	Element	R1 Meters Collection	Meter	Parent-Child	
13							

Figure 17

Formula Data Reference

The Formula Data Reference gives the user the ability to create calculations based on other Attributes in the same AF Database. These calculations are performed on demand only when a PI AF client tool asks for the value of the Attribute, which has this Data Reference.

There are two possible situations:

- Referencing Attributes in the same Element
- Referencing Attributes in other Elements

General Syntax

The syntax for the Formula settings is:

A={Description};[A] where {Description} would be the string that is shown in the Figures below under the Description column.

So for example, in Figure 18 below an Attribute named ***Inherits from C*** has the Description ***|C*** (about half way down the Figure), then the settings string for the Formula would be

A=|C;[A]

Legend for Figures (18 through 24)

The screen captures in the Figures show three columns: **Name**, **Value**, and **Description**.

Name

Attribute name, indicating whether it is a value or inherited using the Formula from another Attribute. There are also three templates; **Base Template**, **A Template**, and **B Template**. The Attributes are categorized according to which template they are defined in. In Figure 18, the Element is created from the **B Template**, which inherits from the **A Template**, which in turn inherits from the **Base Template**.

Value

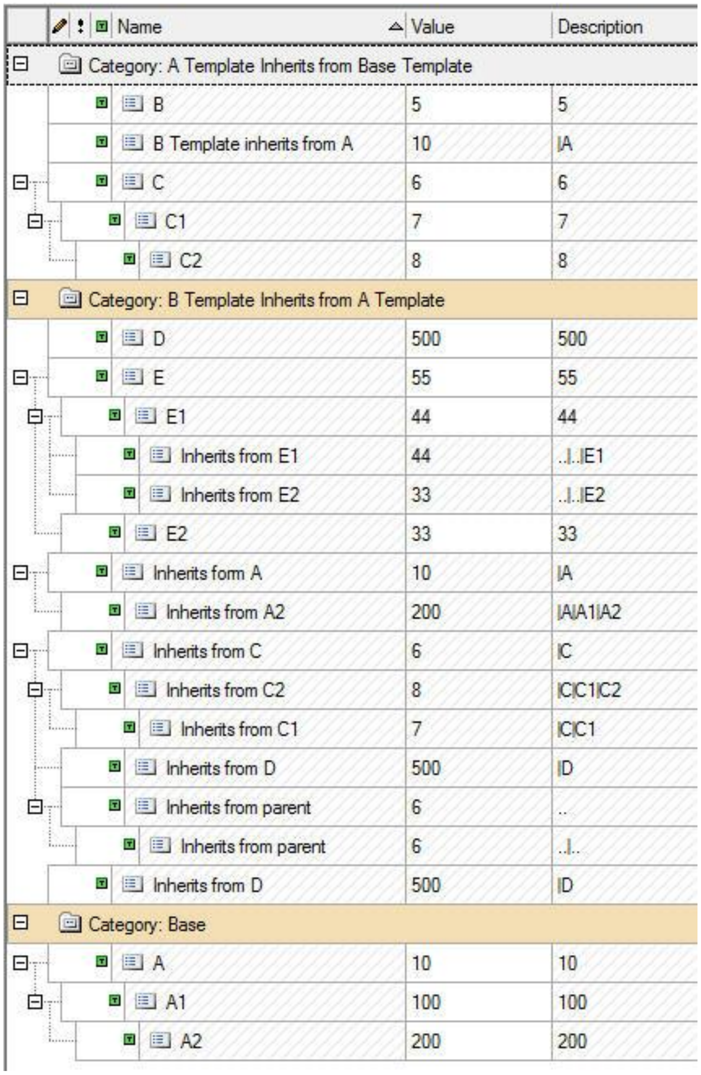
This is the value that is returned by the Formula or a manually input value.

Description

If this is a number, then this Attribute does not contain the Formula Data Reference and the value is manually input (it could be also coming from another data reference). The syntax for the Settings string for the Formula Data Reference is described in the **General Syntax** section above.

Referencing Attributes in the Same Element

Syntax Examples



<table>
<tr><th>Name</th><th>Value</th><th>Description</th></tr>
<tr><td colspan="3">Category: A Template Inherits from Base Template</td></tr>
<tr><td>B</td><td>5</td><td>5</td></tr>
<tr><td>B Template inherits from A</td><td>10</td><td>|A</td></tr>
<tr><td>C</td><td>6</td><td>6</td></tr>
<tr><td>C1</td><td>7</td><td>7</td></tr>
<tr><td>C2</td><td>8</td><td>8</td></tr>
<tr><td colspan="3">Category: B Template Inherits from A Template</td></tr>
<tr><td>D</td><td>500</td><td>500</td></tr>
<tr><td>E</td><td>55</td><td>55</td></tr>
<tr><td>E1</td><td>44</td><td>44</td></tr>
<tr><td>Inherits from E1</td><td>44</td><td>..|..|E1</td></tr>
<tr><td>Inherits from E2</td><td>33</td><td>..|..|E2</td></tr>
<tr><td>E2</td><td>33</td><td>33</td></tr>
<tr><td>Inherits form A</td><td>10</td><td>|A</td></tr>
<tr><td>Inherits from A2</td><td>200</td><td>|A|A1|A2</td></tr>
<tr><td>Inherits from C</td><td>6</td><td>|C</td></tr>
<tr><td>Inherits from C2</td><td>8</td><td>|C|C1|C2</td></tr>
<tr><td>Inherits from C1</td><td>7</td><td>|C|C1</td></tr>
<tr><td>Inherits from D</td><td>500</td><td>|D</td></tr>
<tr><td>Inherits from parent</td><td>6</td><td>..</td></tr>
<tr><td>Inherits from parent</td><td>6</td><td>..|..</td></tr>
<tr><td>Inherits from D</td><td>500</td><td>|D</td></tr>
<tr><td colspan="3">Category: Base</td></tr>
<tr><td>A</td><td>10</td><td>10</td></tr>
<tr><td>A1</td><td>100</td><td>100</td></tr>
<tr><td>A2</td><td>200</td><td>200</td></tr>
</table>

Figure 18

Referencing Attributes in the Other Elements

Syntax Examples

This example uses the Elements named and configured as shown in Figure 19.

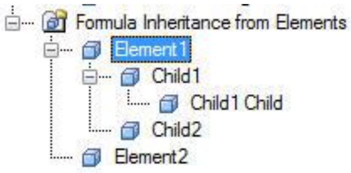
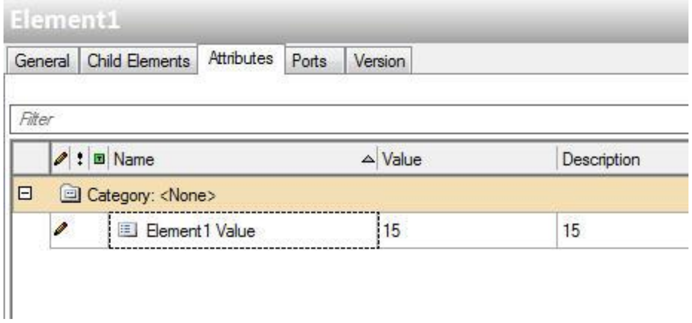
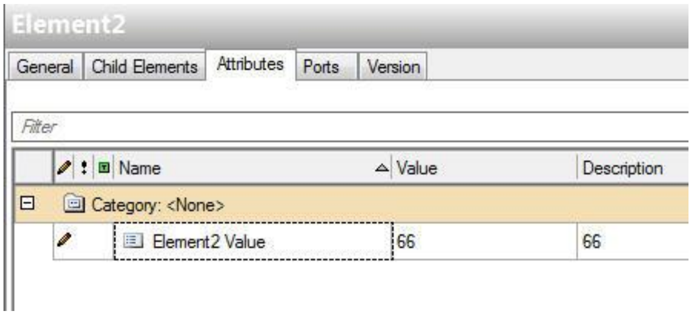


Figure 19



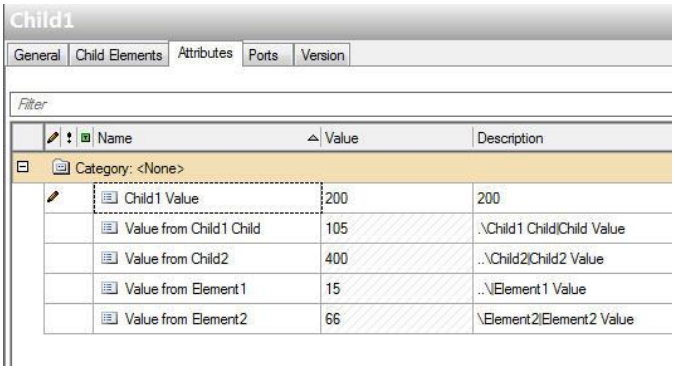
Name	Value	Description
Category: <None>		
Element1 Value	15	15

Figure 20



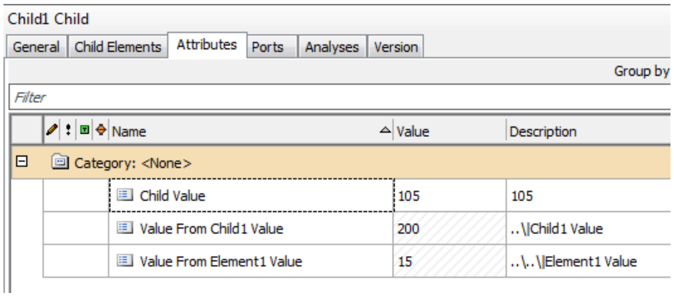
Name	Value	Description
Category: <None>		
Element2 Value	66	66

Figure 21



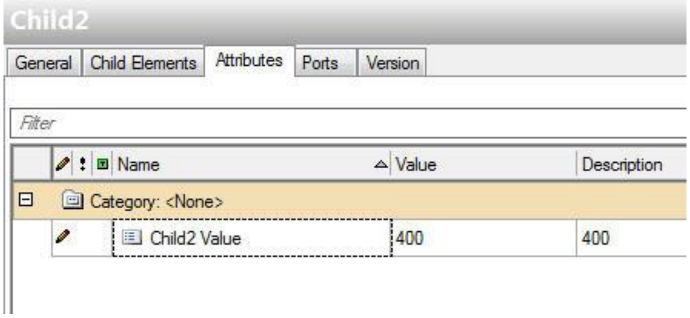
Name	Value	Description
Category: <None>		
Child1 Value	200	200
Value from Child1 Child	105	.\Child1 Child|Child Value
Value from Child2	400	..\Child2|Child2 Value
Value from Element1	15	..|Element1 Value
Value from Element2	66	\Element2|Element2 Value

Figure 22



Name	Value	Description
Category: <None>		
Child Value	105	105
Value From Child1 Value	200	..|Child1 Value
Value From Element1 Value	15	..\..|Element1 Value

Figure 23



Name	Value	Description
Category: <None>		
Child2 Value	400	400

Figure 24

Table Lookup Data Reference

The Table Lookup Data Reference gives the user the ability to create views of data from external databases for use in the PI AF Database. Typically, these would be things like equipment specifications, material properties, etc. The values in these PI AF Tables can then be retrieved from an Element Attribute using a SQL **Select** statement. The syntax is as follows:

SELECT <*Table Field Name1*> FROM <*Table Name*> WHERE <*Table Field Name2*> = %Element% AND <*Table Field Name3*> = %Attribute%

The SQL statement can also use the **OR** and **ORDER BY** clauses.

Substitution Syntax

Example:

This example shows how to use the look up to return the value of the **PITag** field in the PI AF Table **Table1** by comparing the value of an Attribute **PlantCode** of a grand-parent Element in the hierarchy to the **AttributeValue** field value in the Table. The Element **GrandParent1** has an Attribute **PlantCode** (Figure 25). The child Element **Element1** has the **Table Lookup** data reference. The **Select** statement is

SELECT PITag FROM Table1 WHERE ElementName = '%Element%' AND AttributeValue = @[..\..\|PlantCode]

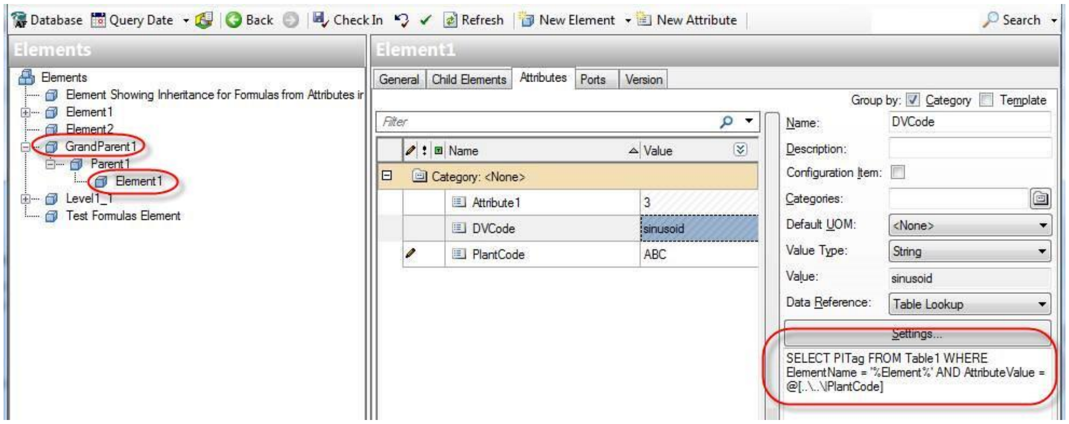
There are two substitutions in the **Select** statement above: **%Element%** and @**[..\..\|PlantCode]**, the explanation of these is below.

ElementName = %Element%

Means the name of the Element is substituted into the comparison (%Element%) and then compared to the value in the **ElementName** field in the PI AF Table **Table1**, see Figure 27.

AttributeValue =@[..\..\|PlantCode]

The @ sign means get the value of an Attribute not the name of it. In this Data Reference the Attribute name to locate must be enclosed inside **[ ]**. (If the Attribute **PlantCode** existed in the same Element, then you do not need to use the **[ ]** symbols, and all you would write is @**|PlantCode**). The **..\..** signifies go up two levels of Elements in the hierarchy, so in this example starting at the **Element1** go two levels up to Element **GrandParent1**. Then the last part of the syntax is **|PlantCode** , the | signifies that the text following is an Attribute name, in this case **PlantCode**. In this case, the value is **DEF** as shown in Figure 26.



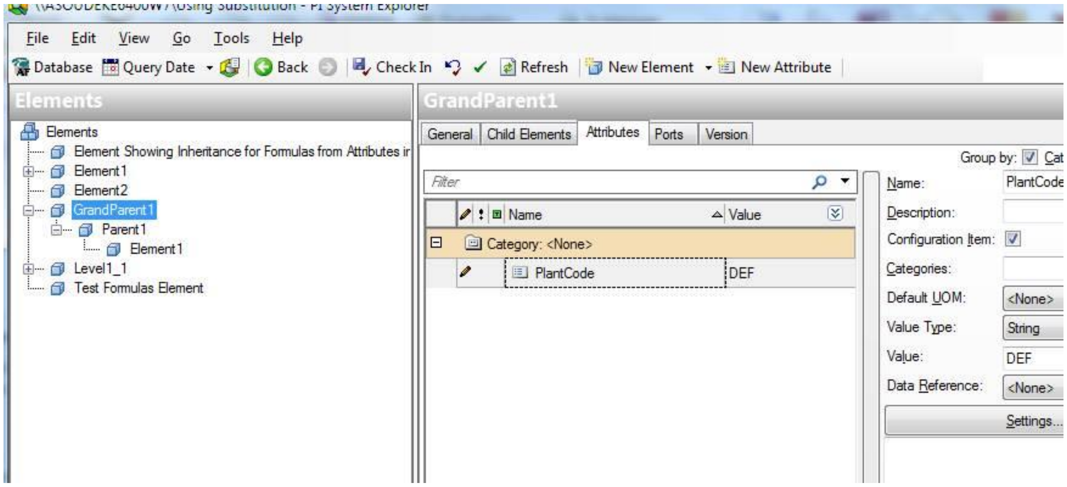
Name	Value
Category: <None>	
Attribute1	3
DVCode	sinusoid
PlantCode	ABC

Group by: ☑ Category □ Template

Name: DVCode
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: sinusoid
Data Reference: Table Lookup
Settings...

```
SELECT PITag FROM Table1 WHERE
ElementName = '%Element%' AND AttributeValue =
@[..\..\|PlantCode]
```

Figure 25

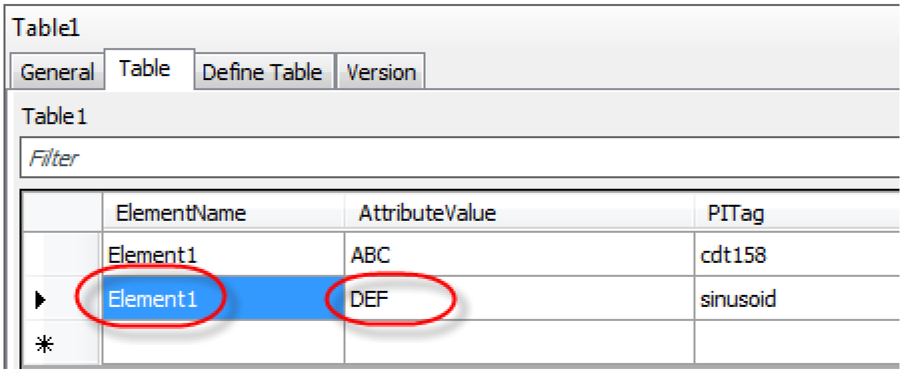


Name	Value
Category: <None>	
PlantCode	DEF

Group by: ☑ Cat

Name: PlantCode
Description:
Configuration Item: ☑
Categories:
Default UOM: <None>
Value Type: String
Value: DEF
Data Reference: <None>
Settings...

Figure 26



	ElementName	AttributeValue	PITag
	Element1	ABC	cdt158
▶	Element1	DEF	sinusoid
✳			

Figure 27

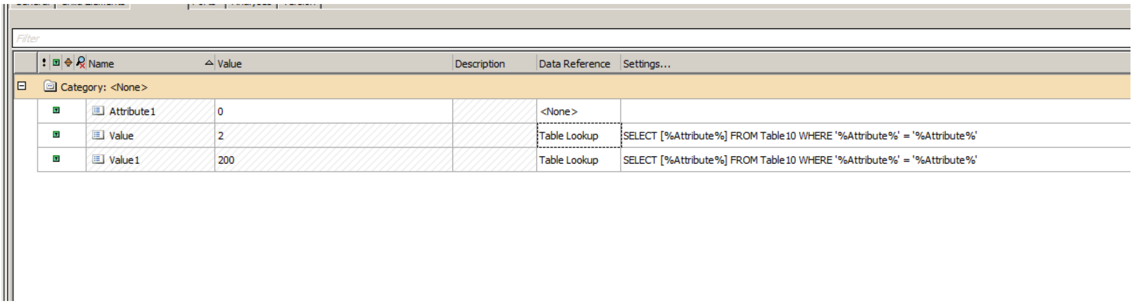
Other useful substitution parameters:

Syntax	Meaning
%Element%	Returns the name of the Element of the current Attribute
%\Element%	Returns the name of the root Element from the hierarchy of the Element of the current Attribute
%..\Element%	Returns the name of the parent Element of the Element of the current Attribute
%ElementDescription%	Returns the description of the Element of the current Attribute
%Attribute%	Returns the name of the Attribute
%..|Atribute%	Returns the name of the parent Attribute of the current Attribute
%System%	Returns the name of the PI AF Server
%Database%	Returns the name of the PI AF database

Example:

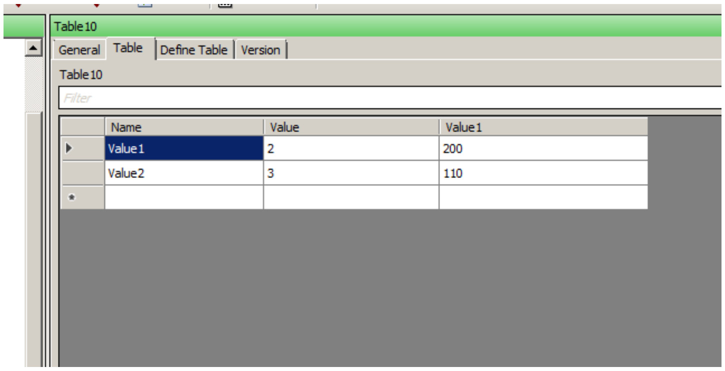
This example shows a judicious naming of the Attributes in the Element Template and the column names of the Table so that the query can be made very generic. In this example the Table has three columns; Name, Value, and Value1. The Attributes in the Element Template are named Value and Value1. Then the same query is used in both Attributes with the syntax as follows.

Select [%Attribute%] From <*Table Name*> Where '%Attribute%' = 'Attribute%'



Name	Value	Description	Data Reference	Settings...
Attribute1	0		<None>	
Value	2		Table Lookup	SELECT [%Attribute%] FROM Table10 WHERE '%Attribute%' = '%Attribute%'
Value1	200		Table Lookup	SELECT [%Attribute%] FROM Table10 WHERE '%Attribute%' = '%Attribute%'

Figure 28



Name	Value	Value1
Value1	2	200
Value2	3	110

Figure 29

String Builder Data Reference (PI AF 2.6 or later)

Its purpose is to allow the user to combine strings together, get values from Attributes that are strings or numeric, and use substitution parameters to get things like the Element name. This can be very useful for obtaining system parameters like the PI system name or values from parent Elements.

Benefits:

- This is useful for getting a string or a numeric value type from other Element Attributes, where the Attributes do not need to have a PI Point Data Reference.
- Create a string by concatenating multiple values from referenced Element Attributes and other text.
- Can get the name of the Element (and its parent Elements). This is a useful technique for making reporting using SQL statements easier and more efficient. I recommend using this approach.

Limitations:

- Does not perform Unit of Measure conversions
- Does not perform any calculations, but there are some basic formatting and string manipulation functions

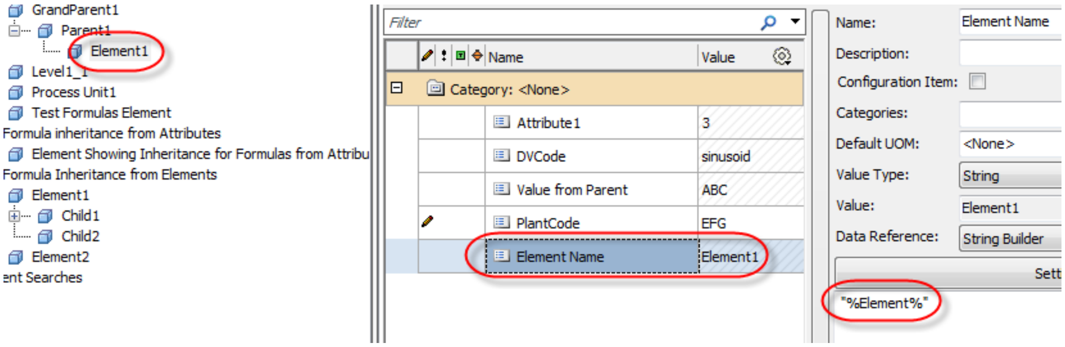
Syntax Examples

Return the name of Element

“%Element%”

Example:

Returns the name of the Element of the current Attribute.



GrandParent1
Parent1
Element1
Level1_1
Process Unit1
Test Formulas Element
Formula inheritance from Attributes
Element Showing Inheritance for Formulas from Attribu
Formula Inheritance from Elements
Element1
Child1
Child2
Element2
ent Searches

Category: <None>

Name	Value
Attribute1	3
DVCode	sinusoid
Value from Parent	ABC
PlantCode	EFG
Element Name	Element1

Name:	Element Name
Description:	
Configuration Item:	□
Categories:	
Default UOM:	<None>
Value Type:	String
Value:	Element1
Data Reference:	String Builder

"%Element%"

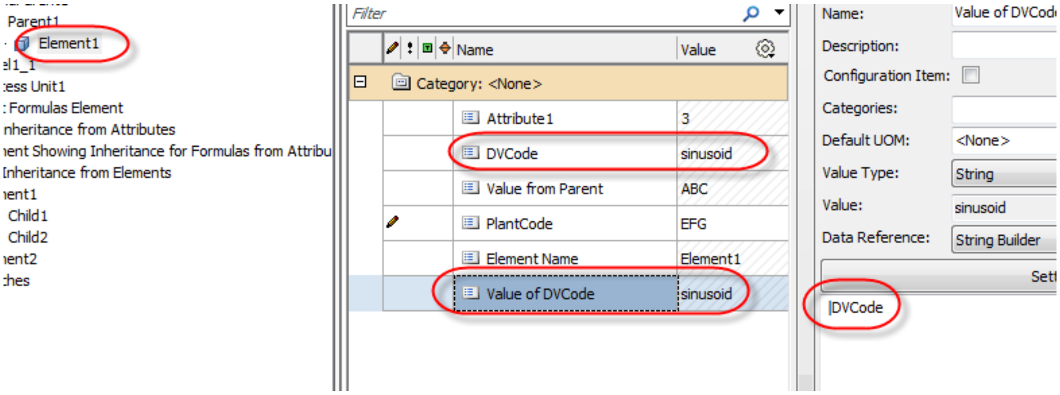
Figure 30

Returning a value from an Element Attribute

|<Attribute Name>

Example:

Returns the value of Attribute **DVCode** in the same Element.



Name	Value
Category: <None>	
Attribute1	3
DVCode	sinusoid
Value from Parent	ABC
PlantCode	EFG
Element Name	Element1
Value of DVCode	sinusoid

Name: Value of DVCod
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: sinusoid
Data Reference: String Builder

|DVCode

Parent1
Element1
el1_1
ess Unit1
Formulas Element
nheritance from Attributes
ent Showing Inheritance for Formulas from Attribu
Inheritance from Elements
ent1
Child1
Child2
ent2
hes
Filter
Sett

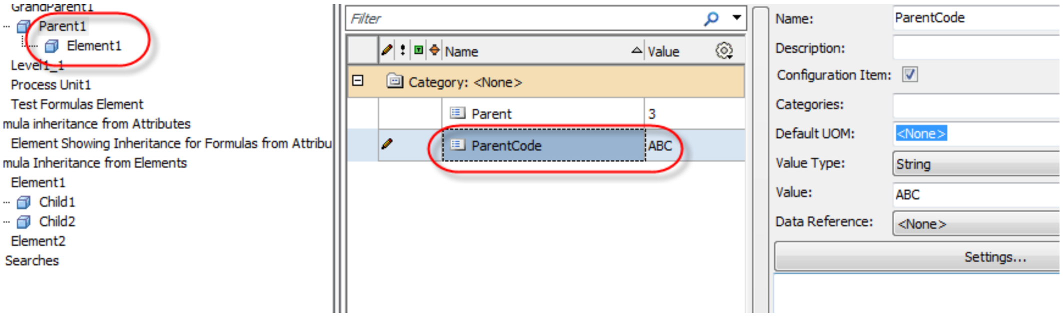
Figure 31

Returning a value from a parent Element Attribute

..\|<Attribute Name>

Example:

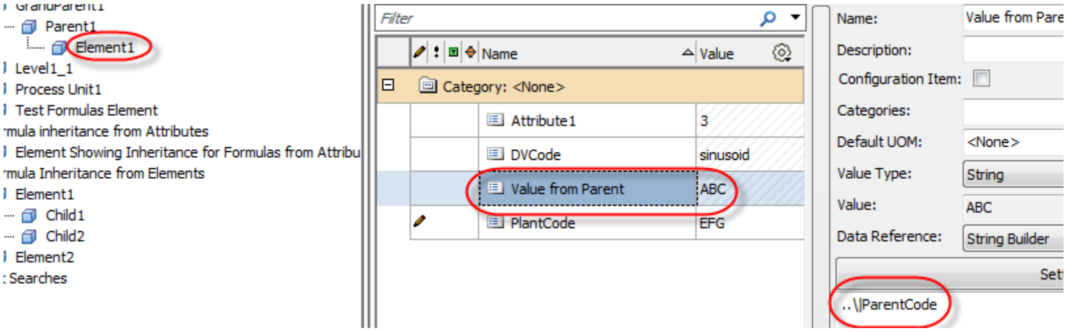
Returning the value of Attribute **PlantCode** that belongs to the parent Element.



Name	Value
Category: <None>	
Parent	3
ParentCode	ABC

Name: ParentCode
Description:
Configuration Item: ☑
Categories:
Default UOM: <None>
Value Type: String
Value: ABC
Data Reference: <None>

Figure 32


GrandParent1
Parent1
Element1
Level1_1
Process Unit1
Test Formulas Element
rmula inheritance from Attributes
Element Showing Inheritance for Formulas from Attribu
rmula Inheritance from Elements
Element1
Child1
Child2
Element2
: Searches
Filter
Set

Name	Value
Category: <None>	
Attribute1	3
DVCode	sinusoid
Value from Parent	ABC
PlantCode	EFG

Name: Value from Pare
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: ABC
Data Reference: String Builder
..\|ParentCode

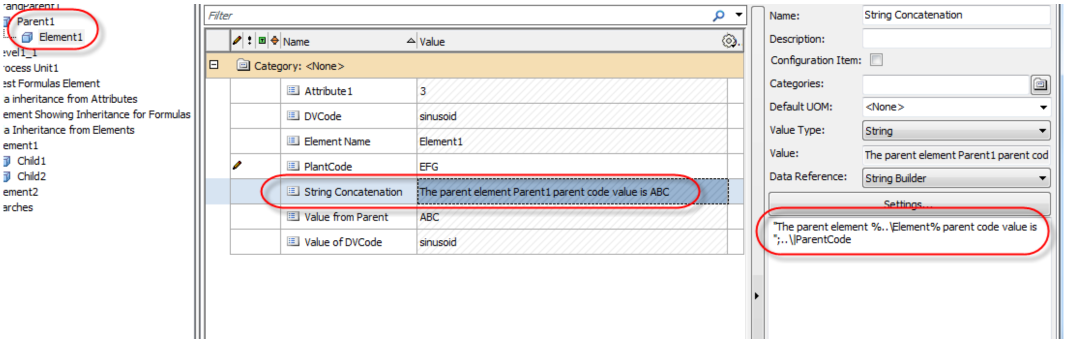
Figure 33

Concatenating values and strings

“The parent element %..\Element% parent code value “;..\|*<Attribute Name>*

Example:

Ccreating a string by concatenating the name of the parent Element and the value of the Attribute **PlantCode** of the parent Element. The **;** is the concatenation symbol.



Name	Value
Category: <None>	
Attribute1	3
DVCode	sinusoid
Element Name	Element1
PlantCode	EFG
String Concatenation	The parent element Parent1 parent code value is ABC
Value from Parent	ABC
Value of DVCode	sinusoid

Name: String Concatenation
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: The parent element Parent1 parent cod
Data Reference: String Builder
"The parent element %..\Element% parent code value is ";..\|ParentCode

Figure 34

Other useful substitution parameters:

Syntax	Meaning
%Element%	Returns the name of the Element of the current Attribute
%\Element%	Returns the name of the root Element from the hierarchy of the Element of the current Attribute
%..\Element%	Returns the name of the parent Element of the Element of the current Attribute

%ElementPath%	Returns the path from the top level Element to the referenced Element, e.g. Asset1\Asset2\Asset3
%ElementDescription%	Returns the description of the Element of the current Attribute
%Attribute%	Returns the name of the Attribute
%..|Atribute%	Returns the name of the parent Attribute of the current Attribute
%Description%	Returns the description of the current Attribute
%System%	Returns the name of the PI AF Server
%Database%	Returns the name of the PI AF database

PI Event Frames

Event Frames can have Attributes that get data from other Event Frames, when the Event Frames are associated with each other as parent/child. The Attributes can also get data from the referenced Elements.

Data from Elements

There are three ways of getting data from a referenced Element: PI Point, String Builder, and Formula Data References. Benefits and limitations of each as well as examples of the syntax are described in the sections below.

PI Point Data Reference

Benefits:

- This is useful for getting data at different aggregations from the referenced Element's Attribute. (Like maximum, minimum, average, etc.)
- Will perform Unit of Measure conversions. The Attributes in the Element and the Event Frame must be of the same class.
- Passes time context

Limitations:

- The Attribute in the Element must be a PI Point Data Reference.

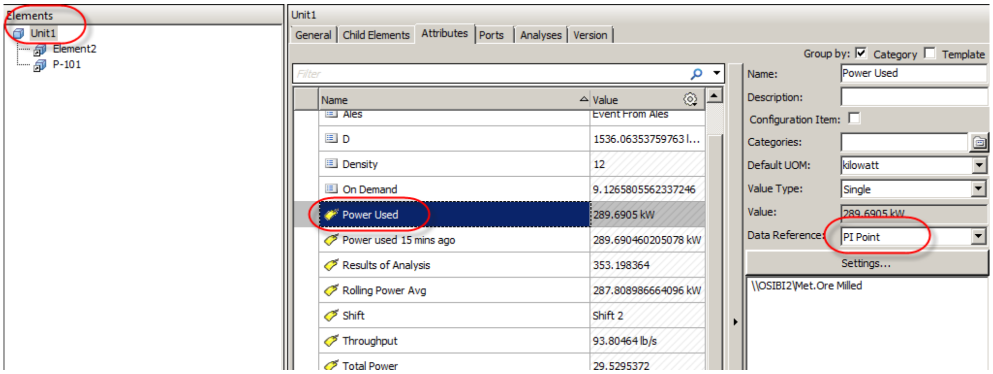
Syntax Examples

An Event Frame returning a value from referenced Element

.\Elements[.]|*<Attribute Name>*

Example:

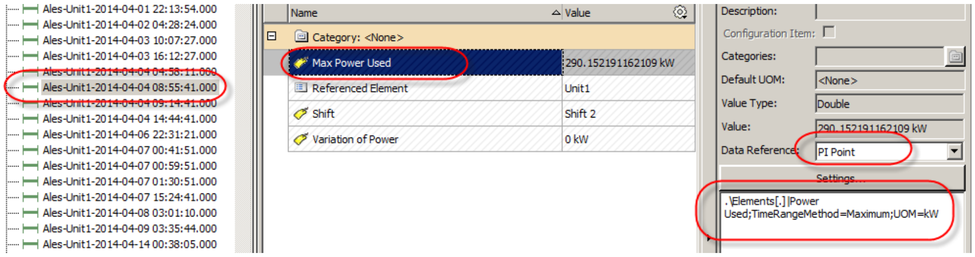
Event Frame returning the maximum value of the Element **Unit1** for the Attribute **Power Used**.



Name	Value
Ales	Event From Ales
D	1536.06353759763 l...
Density	12
On Demand	9.1265805562337246
Power Used	289.6905 kW
Power used 15 mins ago	289.690460205078 kW
Results of Analysis	353.198364
Rolling Power Avg	287.808986664096 kW
Shift	Shift 2
Throughput	93.80464 lb/s
Total Power	29.5295372

Name: Power Used
Description:
Configuration Item: □
Categories:
Default UOM: kilowatt
Value Type: Single
Value: 289.6905 kW
Data Reference: PI Point
\\OSIBI2\Met.Ore Milled

Figure 35



Ales-Unit1-2014-04-01 22:13:54.000
Ales-Unit1-2014-04-02 04:28:24.000
Ales-Unit1-2014-04-03 10:07:27.000
Ales-Unit1-2014-04-03 16:12:27.000
Ales-Unit1-2014-04-04 04:58:11.000
Ales-Unit1-2014-04-04 08:55:41.000
Ales-Unit1-2014-04-04 09:14:41.000
Ales-Unit1-2014-04-04 14:44:41.000
Ales-Unit1-2014-04-06 22:31:21.000
Ales-Unit1-2014-04-07 00:41:51.000
Ales-Unit1-2014-04-07 00:59:51.000
Ales-Unit1-2014-04-07 01:30:51.000
Ales-Unit1-2014-04-07 15:24:41.000
Ales-Unit1-2014-04-08 03:01:10.000
Ales-Unit1-2014-04-09 03:35:44.000
Ales-Unit1-2014-04-14 00:38:05.000

Name	Value
Category: <None>	
Max Power Used	290.152191162109 kW
Referenced Element	Unit1
Shift	Shift 2
Variation of Power	0 kW

Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 290.152191162109 kW
Data Reference: PI Point
Settings...
.\Elements[.]|Power Used;TimeRangeMethod=Maximum;UOM=kW

Figure 36

Use the syntax in the above example when there is only one referenced Element. It is possible and desirable in some situations to have more than one referenced Element for an Event Frame. In such cases, different search criteria need to be used to get data from the desired Element. The most useful criteria are:

- Index (1 based)
- Name (can use wild cards)
- Template (cannot use wild cards, must be explicit)
- Category (cannot use wild cards, must be explicit)

Example syntax for more than one referenced Element:

Syntax	Meaning
.\Elements[@Index = 3]|Attr1	Get the value Attribute **Attr1** from the 3rd referenced Element
.\Elements[@Name = Pump*]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose name starts with **Pump**
.\Elements[@Template = Process Units]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose Template is **Process Units**
.\Elements[@Category = Rotating Equip]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose Category is **Rotating Equip**
.\Elements[@Category=Rotating Equip][@Name = Pump*]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose Category is **Rotating** Equip and name starts with **Pump**

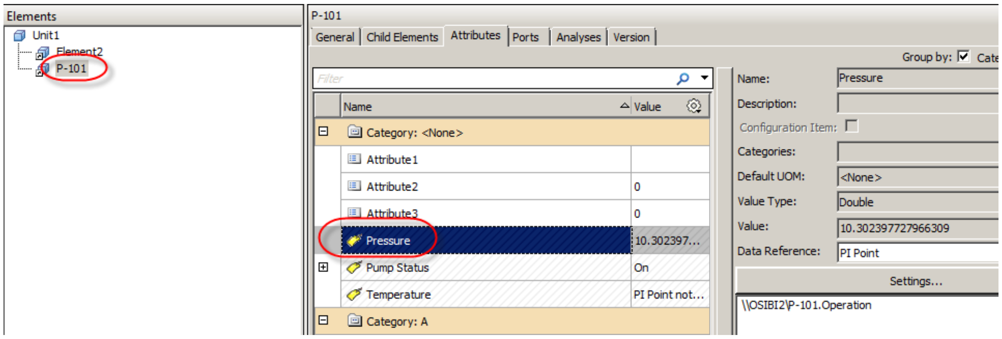
Note: - You can combine several search criteria together in one call as shown by the last row in the table above.

An Event Frame returning a value from referenced Element's child Element

.\Elements[.]\[@Name=P-1*|*<Attribute Name>*

Note: - The syntax above is using the Name as the search criteria for the child Element starting with **P-1**, but you can also use the Index, Template, and Category.

Example:
Event Frame returning the maximum value of the **Unit1** child Element whose name starts with **P-1** for the Attribute **Pressure**. The Event Frame has **Unit1** as the referenced Element.

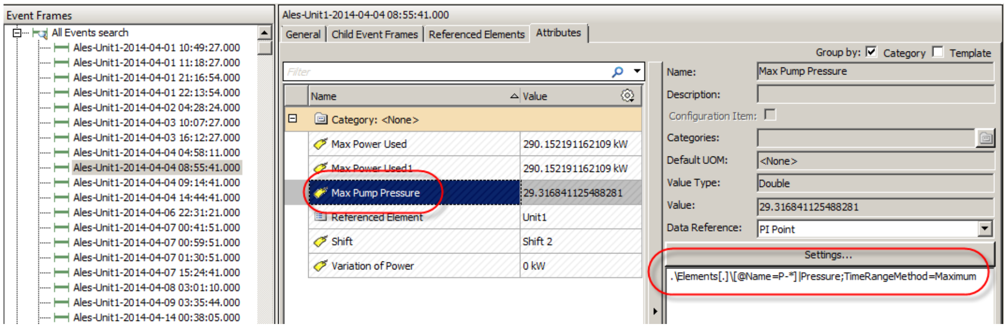


Name	Value
Category: <None>	
Attribute1	
Attribute2	0
Attribute3	0
Pressure	10.302397...
Pump Status	On
Temperature	PI Point not...
Category: A	

Name: Pressure
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 10.302397727966309
Data Reference: PI Point

\\OSIBI2\P-101.Operation

Figure 37


Event Frames
All Events search
Ales-Unit1-2014-04-01 10:49:27.000
Ales-Unit1-2014-04-01 11:18:27.000
Ales-Unit1-2014-04-01 21:16:54.000
Ales-Unit1-2014-04-01 22:13:54.000
Ales-Unit1-2014-04-02 04:28:24.000
Ales-Unit1-2014-04-03 10:07:27.000
Ales-Unit1-2014-04-03 16:12:27.000
Ales-Unit1-2014-04-04 04:58:11.000
Ales-Unit1-2014-04-04 08:55:41.000
Ales-Unit1-2014-04-04 09:14:41.000
Ales-Unit1-2014-04-04 14:44:41.000
Ales-Unit1-2014-04-06 22:31:21.000
Ales-Unit1-2014-04-07 00:41:51.000
Ales-Unit1-2014-04-07 00:59:51.000
Ales-Unit1-2014-04-07 01:30:51.000
Ales-Unit1-2014-04-07 15:24:41.000
Ales-Unit1-2014-04-08 03:01:10.000
Ales-Unit1-2014-04-09 03:35:44.000
Ales-Unit1-2014-04-14 00:38:05.000
Ales-Unit1-2014-04-04 08:55:41.000
General
Child Event Frames
Referenced Elements
Attributes
Group by: ☑ Category □ Template
Filter
Settings...

Name	Value
Category: <None>	
Max Power Used	290.15219162109 kW
Max Power Used1	290.15219162109 kW
Max Pump Pressure	29.316841125488281
Referenced Element	Unit1
Shift	Shift 2
Variation of Power	0 kW

Name: Max Pump Pressure
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 29.316841125488281
Data Reference: PI Point

```
.\Elements[.]\[@Name=P-*]|Pressure;TimeRangeMethod=Maximum
```

Figure 38

String Builder Data Reference

Benefits:

- This is useful for getting a string or a numeric value type from the referenced Element's Attribute, where the referenced Element's Attribute does not need to have a PI Point Data Reference.
- Create a string by concatenating multiple values from referenced Element Attributes and other text.
- Passes time context – the end time of the Event Frame

- Can get the name of the referenced Element (and its parent Elements). This is a useful technique for making reporting using SQL statements easier and more efficient. I recommend using this approach.

Limitations:

- Does not perform Unit of Measure conversions
- Does not perform any aggregation or calculations

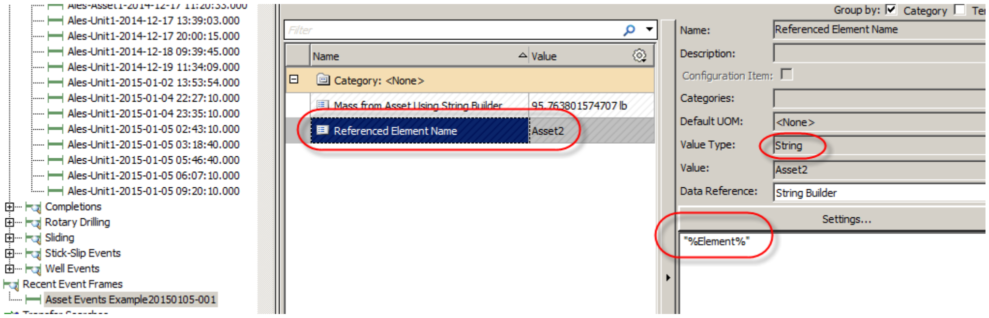
Syntax Examples

An Event Frame returning the name of a referenced Element

“%Element%”

Example:

Event Frame returning the name of the referenced Element.


Ales-Asset1-2014-12-17 11:20:33.000
Ales-Unit1-2014-12-17 13:39:03.000
Ales-Unit1-2014-12-17 20:00:15.000
Ales-Unit1-2014-12-18 09:39:45.000
Ales-Unit1-2014-12-19 11:34:09.000
Ales-Unit1-2015-01-02 13:53:54.000
Ales-Unit1-2015-01-04 22:27:10.000
Ales-Unit1-2015-01-04 23:35:10.000
Ales-Unit1-2015-01-05 02:43:10.000
Ales-Unit1-2015-01-05 03:18:40.000
Ales-Unit1-2015-01-05 05:46:40.000
Ales-Unit1-2015-01-05 06:07:10.000
Ales-Unit1-2015-01-05 09:20:10.000
Completions
Rotary Drilling
Sliding
Stick-Slip Events
Well Events
Recent Event Frames
Asset Events Example20150105-001
Group by: ☑ Category □ Te
Filter
Settings...

Name	Value
Category: <None>	
Mass from Asset Using String Builder	95.763801574707 lb
Referenced Element Name	Asset2

Name: Referenced Element Name
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: Asset2
Data Reference: String Builder

"%Element%"

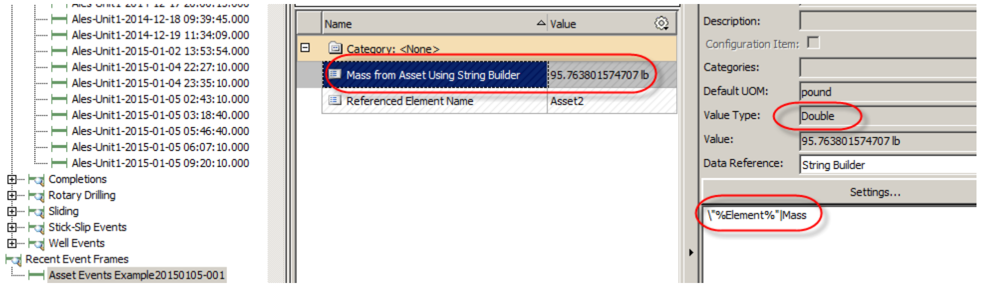
Figure 39

An Event Frame returning a value from a referenced Element

\”%Element%”|*<Attribute Name>* **(AF 2.6 or earlier)**

Example:

Event Frame returning the value of Attribute **Mass** from the referenced Element.


Ales-Unit1-2014-12-17 20:00:15.000
Ales-Unit1-2014-12-18 09:39:45.000
Ales-Unit1-2014-12-19 11:34:09.000
Ales-Unit1-2015-01-02 13:53:54.000
Ales-Unit1-2015-01-04 22:27:10.000
Ales-Unit1-2015-01-04 23:35:10.000
Ales-Unit1-2015-01-05 02:43:10.000
Ales-Unit1-2015-01-05 03:18:40.000
Ales-Unit1-2015-01-05 05:46:40.000
Ales-Unit1-2015-01-05 06:07:10.000
Ales-Unit1-2015-01-05 09:20:10.000
Completions
Rotary Drilling
Sliding
Stick-Slip Events
Well Events
Recent Event Frames
Asset Events Example20150105-001
Settings...

Name	Value
Category: <None>	
Mass from Asset Using String Builder	95.763801574707 lb
Referenced Element Name	Asset2

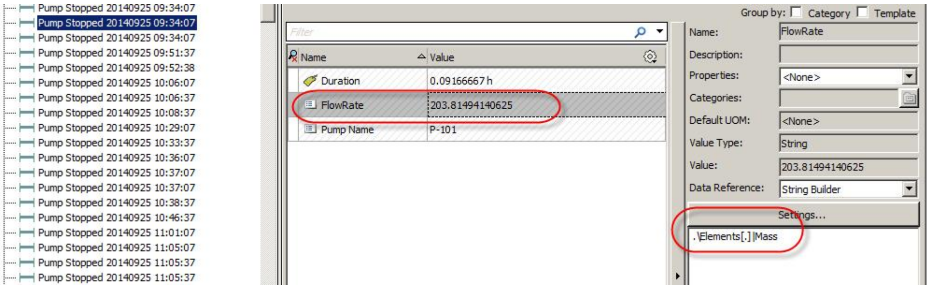
Description:
Configuration Item: □
Categories:
Default UOM: pound
Value Type: Double
Value: 95.763801574707 lb
Data Reference: String Builder

\"%Element%"|Mass

Figure 40

\Elements[.]|*<Attribute Name>* **(AF 2.7 or later)**

Example:
Event Frame returning the value of Attribute **Mass** from the referenced Element.



Pump Stopped 20140925 09:34:07
Pump Stopped 20140925 09:34:07
Pump Stopped 20140925 09:34:07
Pump Stopped 20140925 09:51:37
Pump Stopped 20140925 09:52:38
Pump Stopped 20140925 10:06:07
Pump Stopped 20140925 10:06:37
Pump Stopped 20140925 10:08:37
Pump Stopped 20140925 10:29:07
Pump Stopped 20140925 10:33:37
Pump Stopped 20140925 10:36:07
Pump Stopped 20140925 10:37:07
Pump Stopped 20140925 10:37:07
Pump Stopped 20140925 10:38:37
Pump Stopped 20140925 10:46:37
Pump Stopped 20140925 11:01:07
Pump Stopped 20140925 11:05:07
Pump Stopped 20140925 11:05:37
Pump Stopped 20140925 11:05:37

Name	Value
Duration	0.09166667 h
FlowRate	203.81494140625
Pump Name	P-101

Group by: □ Category □ Template

Name: FlowRate
Description:
Properties: <None>
Categories:
Default UOM: <None>
Value Type: String
Value: 203.81494140625
Data Reference: String Builder

Settings...

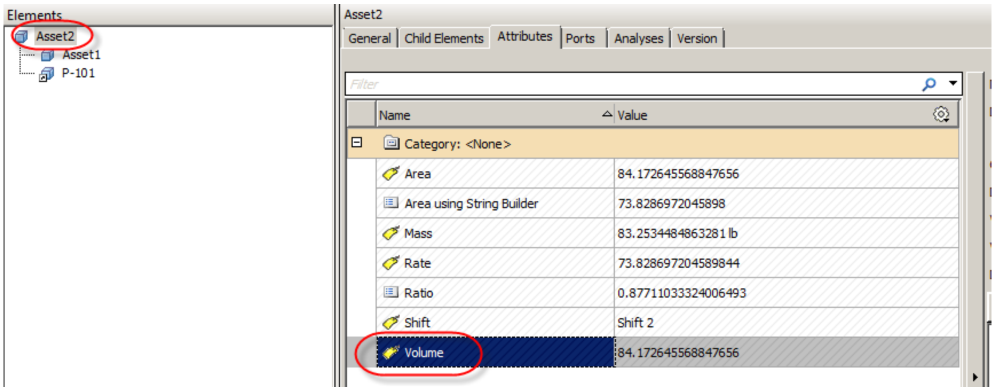
.\Elements[.]|Mass

Figure 41

An Event Frame returning a value from a referenced Element where the Event Frame Attribute has the same name as the Attribute in the referenced Element

\”%Element%”|%Attribute%   (AF 2.6 or earlier)

Example:
Event Frame returning the value of Attribute **Volume** from the referenced Element. The Attribute in the Event Frame is also name **Volume**.

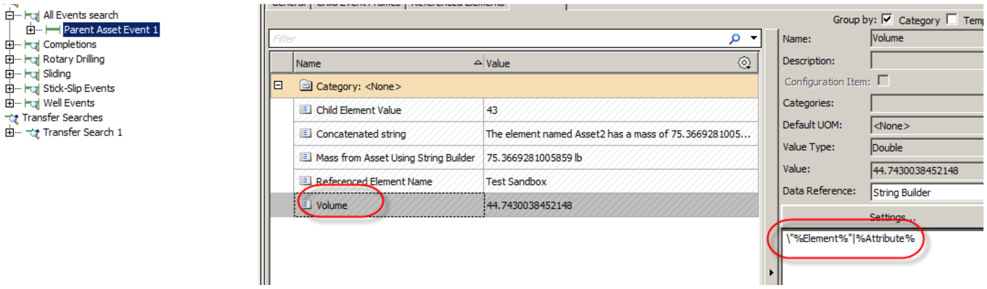


Elements
Asset2
Asset1
P-101

Asset2

Name	Value
Category: <None>	
Area	84.172645568847656
Area using String Builder	73.8286972045898
Mass	83.2534484863281 lb
Rate	73.828697204589844
Ratio	0.87711033324006493
Shift	Shift 2
Volume	84.172645568847656

Figure 42



All Events search
Parent Asset Event 1
Completions
Rotary Drilling
Sliding
Stick-Slip Events
Well Events
Transfer Searches
Transfer Search 1

Name	Value
Category: <None>	
Child Element Value	43
Concatenated string	The element named Asset2 has a mass of 75.3669281005...
Mass from Asset Using String Builder	75.3669281005859 lb
Referenced Element Name	Test Sandbox
Volume	44.7430038452148

Group by: ☑ Category □ Temp

Name: Volume
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 44.7430038452148
Data Reference: String Builder

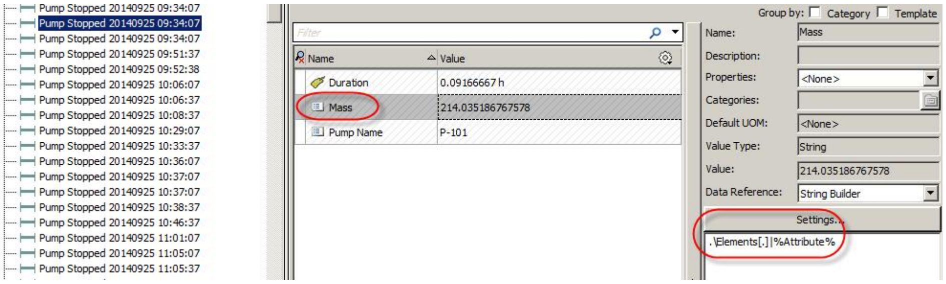
Settings...

\"%Element%"|%Attribute%

Figure 43

\Elements[.]|*%Attribute%* (AF 2.7 or later)

Example:
Event Frame returning the value of Attribute **Mass** from the referenced Element.



- Pump Stopped 20140925 09:34:07
- Pump Stopped 20140925 09:34:07
- Pump Stopped 20140925 09:34:07
- Pump Stopped 20140925 09:51:37
- Pump Stopped 20140925 09:52:38
- Pump Stopped 20140925 10:06:07
- Pump Stopped 20140925 10:06:37
- Pump Stopped 20140925 10:08:37
- Pump Stopped 20140925 10:29:07
- Pump Stopped 20140925 10:33:37
- Pump Stopped 20140925 10:36:07
- Pump Stopped 20140925 10:37:07
- Pump Stopped 20140925 10:37:07
- Pump Stopped 20140925 10:38:37
- Pump Stopped 20140925 10:46:37
- Pump Stopped 20140925 11:01:07
- Pump Stopped 20140925 11:05:07
- Pump Stopped 20140925 11:05:37

Name	Value
Duration	0.09166667 h
Mass	214.035186767578
Pump Name	P-101

Group by: ☐ Category ☐ Template

Name: Mass
Description:
Properties: <None>
Categories:
Default UOM: <None>
Value Type: String
Value: 214.035186767578
Data Reference: String Builder
Settings...
.\Elements[.]|%Attribute%

Figure 44

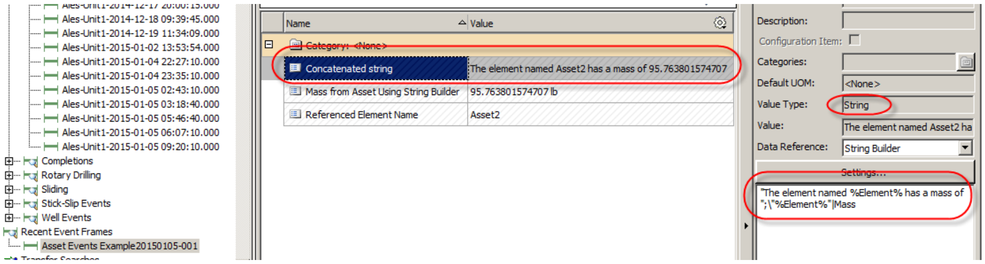
An Event Frame returning a value and a name from a referenced Element and creating a concatenated string

"The element named %Element% has a mass of ";\"%Element%"|*<Attribute Name>* (AF 2.6 or earlier)

"The element named %Element% has a mass of ";.\Elements[.]|*<Attribute Name>* (AF 2.7 or later)

Example:

Event Frame creating a string by concatenating the name and description of the referenced Element and the value of the Attribute **Mass** of the referenced Element. The **;** is the concatenation symbol.



- Ales-Unit1-2014-12-17 20:00:15.000
- Ales-Unit1-2014-12-18 09:39:45.000
- Ales-Unit1-2014-12-19 11:34:09.000
- Ales-Unit1-2015-01-02 13:53:54.000
- Ales-Unit1-2015-01-04 22:27:10.000
- Ales-Unit1-2015-01-04 23:35:10.000
- Ales-Unit1-2015-01-05 02:43:10.000
- Ales-Unit1-2015-01-05 03:18:40.000
- Ales-Unit1-2015-01-05 05:46:40.000
- Ales-Unit1-2015-01-05 06:07:10.000
- Ales-Unit1-2015-01-05 09:20:10.000
- Completions
- Rotary Drilling
- Sliding
- Stick-Slip Events
- Well Events
- Recent Event Frames
- Asset Events Example20150105-001

Category: <None>

Name	Value
Concatenated string	The element named Asset2 has a mass of 95.763801574707
Mass from Asset Using String Builder	95.763801574707 lb
Referenced Element Name	Asset2

Description:
Configuration Item: ☐
Categories:
Default UOM: <None>
Value Type: String
Value: The element named Asset2 ha
Data Reference: String Builder
Settings...
"The element named %Element% has a mass of ";\"%Element%"|Mass

Figure 45

Other useful substitution parameters:

Syntax	Meaning
%ElementPath%	Returns the path from the top level Element to the referenced Element, e.g. Asset1\Asset2\Asset3

%ElementDescription%	Returns the description of the referenced Element
%..\Element%	Returns the name of the referenced Element parent Element. If there is no parent Element a blank is returned.
%StartTime%	Returns the start time of the Event Frame
%EndTime%	Returns the end time of the Event Frame
%System%	Returns the name of the PI AF Server
%Database%	Returns the name of the PI AF database

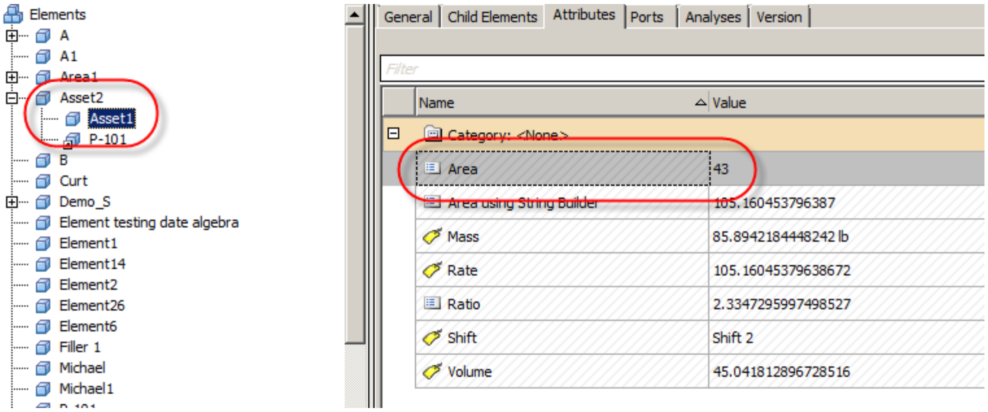
An Event Frame returning a value from referenced Element’s child Element

.\Elements[.]\[@Name=A*|*<Attribute Name>*

Note: - The syntax above is using the Name as the search criteria for the child Element starting with the letter **A**, but you can also use the Index, Template, and Category.

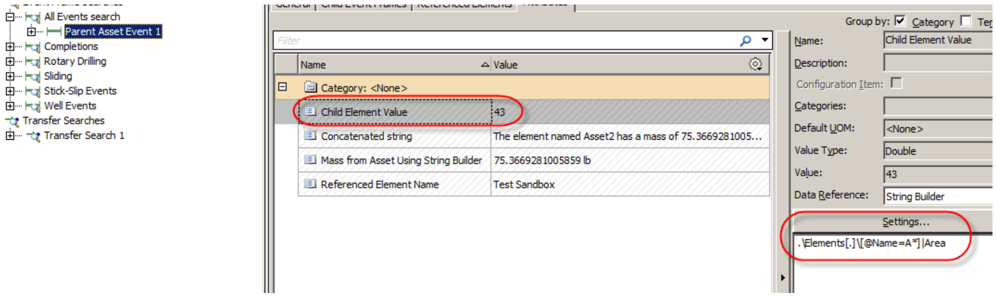
Example:

Event Frame returning the value of the Attribute **Mass** from the child Element **Asset1** of the referenced Element **Asset2**.



Name	Value
Category: <None>	
Area	43
Area using String Builder	105.160453796387
Mass	85.8942184448242 lb
Rate	105.16045379638672
Ratio	2.3347295997498527
Shift	Shift 2
Volume	45.041812896728516

Figure 46



- All Events search
  - Parent Asset Event 1
- Completions
- Rotary Drilling
- Sliding
- Stick-Slip Events
- Well Events
- Transfer Searches
  - Transfer Search 1

Group by: ☑ Category □ Te

Name	Value
Category: <None>	
Child Element Value	43
Concatenated string	The element named Asset2 has a mass of 75.3669281005...
Mass from Asset Using String Builder	75.3669281005859 lb
Referenced Element Name	Test Sandbox

Name: Child Element Value
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 43
Data Reference: String Builder
Settings...
.\Elements[.]\[@Name=A*]|Area

Figure 47

Formula Data Reference

Benefits:

- This is useful for getting a numeric value type from the referenced Element’s Attribute, where the referenced Element’s Attribute does not need to have a PI Point Data Reference.
- Passes time context – the end time of the Event Frame

Limitations:

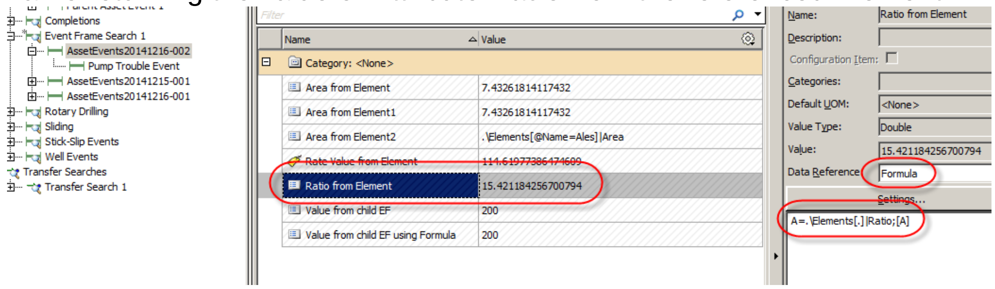
- Only works for numeric values

Syntax Examples

An Event Frame returning a value from a referenced Element

A=.\Elements[.]|*<Attribute>*;[A]

Example:
Event Frame returning the value of Attribute **Ratio** from the referenced Element.



- Completions
- Event Frame Search 1
  - AssetEvents20141216-002
    - Pump Trouble Event
  - AssetEvents20141215-001
  - AssetEvents20141216-001
- Rotary Drilling
- Sliding
- Stick-Slip Events
- Well Events
- Transfer Searches
  - Transfer Search 1

Name	Value
Category: <None>	
Area from Element	7.43261814117432
Area from Element1	7.43261814117432
Area from Element2	.\Elements[@Name=Ales]|Area
~~Rate Value from Element~~	~~114.61977386474609~~
Ratio from Element	15.421184256700794
Value from child EF	200
Value from child EF using Formula	200

Name: Ratio from Element
Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 15.421184256700794
Data Reference: Formula
Settings...
A=.\Elements[.]|Ratio;[A]

Figure 48

Use the syntax in the above example when there is only one referenced Element. It is possible and desirable in some situations to have more than one referenced Element for an Event Frame. In such cases, different search criteria need to be used to get data from the desired Element. The most useful criteria are:

- Index (1 based)
- Name (can use wild cards)
- Template (cannot use wild cards, must be explicit)
- Category (cannot use wild cards, must be explicit)

Example syntax for more than one referenced Element:

Syntax	Meaning
A=.\Elements[@Index = 3]|Attr1	Get the value Attribute **Attr1** from the 3rd referenced Element
A=.\Elements[@Name = Pump*]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose name starts with **Pump**
A=.\Elements[@Template = Process Units]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose Template is **Process Units**
A=.\Elements[@Category = Rotating Equip]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose Category is **Rotating Equip**
A=.\Elements[@Category=Rotating Equip][@Name = Pump*]|Attr1	Get the value Attribute **Attr1** from the first referenced Element whose Category is **Rotating** Equip and name starts with **Pump**

Note: - You can combine several search criteria together in one call as shown by the last row in the table above.

Data from other Event Frames

There are two ways of getting data from child Event Frame: String Builder, and Formula Data References. You can only get data from Event Frames that are in a parent-child relationship. Benefits and limitations of each as well as examples of the syntax are described in the sections below.

String Builder Data Reference

Benefits:

- This is useful for getting a string or a numeric value type from the referenced Element's Attribute, where the referenced Element's Attribute does not need to have a PI Point Data Reference.
- Create a string by concatenating multiple values from referenced Element Attributes and other text.
- Passes time context – the end time of the Event Frame

- Can get the name of the referenced Element (and its parent Elements). This is a useful technique for making reporting using SQL statements easier and more efficient. I recommend using this approach.

Limitations:

- Does not perform Unit of Measure conversions
- Does not perform any aggregation or calculations

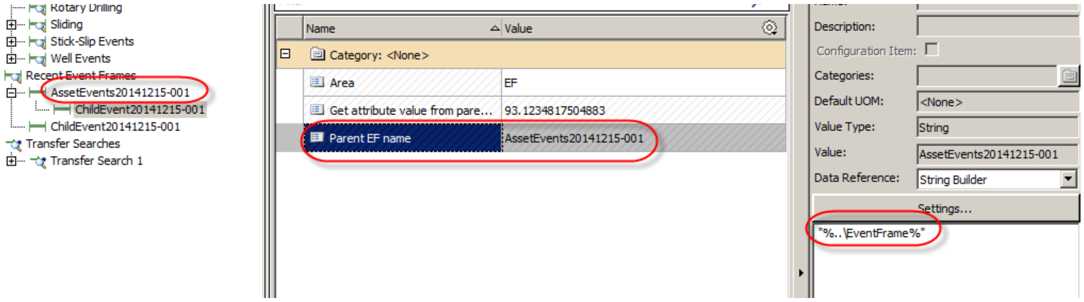
Syntax Examples

A child Event Frame returning a parent Event Frame name

"%..\EventFrame%"

Example:

Event Frame returning the name of the parent Event Frame.



Name	Value
Category: <None>	
Area	EF
Get attribute value from pare...	93.1234817504883
Parent EF name	AssetEvents20141215-001

Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: AssetEvents20141215-001
Data Reference: String Builder

"%..\EventFrame%"

Figure 49

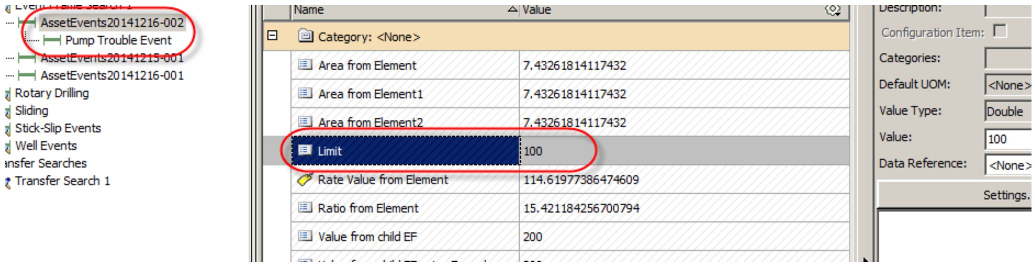
Other useful substitution parameters:

Syntax	Meaning
%Template%	Returns the name of the Event Frame Template
%EventFrame%	Returns the name of the Event Frame
%..\EventFrame%	Returns the name of the parent Event Frame of the current Event Frame. If there is no parent Event Frame a blank is returned.
%EventFrameDescription%	Returns the Event Frame description
%StartTime%	Returns the start time of the Event Frame
%EndTime%	Returns the end time of the Event Frame
%System%	Returns the name of the PI AF Server
%Database%	Returns the name of the PI AF database

A child Event Frame reading a parent Event Frame value

..\|<Attribute>

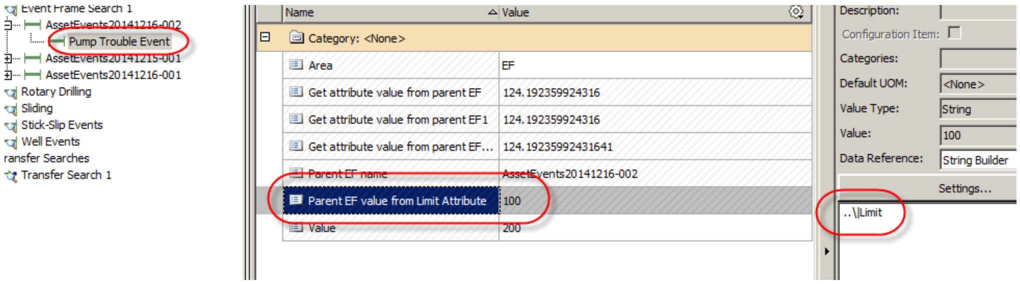
Example:
Event Frame returning the value of Attribute **Limit** from the parent Event Frame.



Name	Value
Category: <None>	
Area from Element	7.43261814117432
Area from Element1	7.43261814117432
Area from Element2	7.43261814117432
Limit	100
Rate Value from Element	114.61977386474609
Ratio from Element	15.421184256700794
Value from child EF	200

Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: Double
Value: 100
Data Reference: <None>
Settings.

Figure 50



Name	Value
Category: <None>	
Area	EF
Get attribute value from parent EF	124.192359924316
Get attribute value from parent EF1	124.192359924316
Get attribute value from parent EF…	124.19235992431641
Parent EF name	AssetEvents20141216-002
Parent EF value from Limit Attribute	100
Value	200

Description:
Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: 100
Data Reference: String Builder
Settings...
..\|Limit

Figure 51

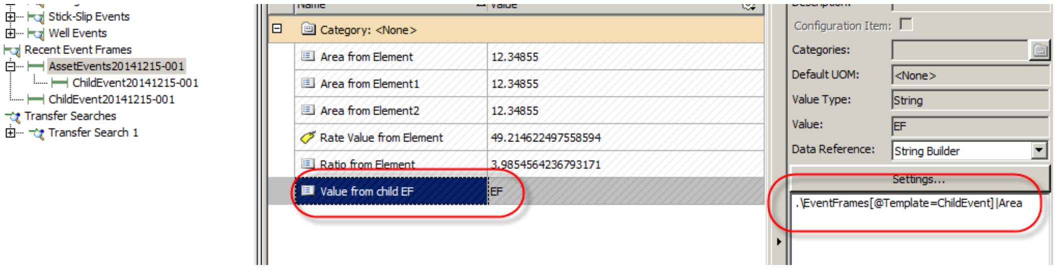
A parent Event Frame reading a child Event Frame Attribute value

.\EventFrames[@Template=<ChildEFName>]|<Attribute>

Note: - The syntax above is using the Template as the search criteria for the child Element, but you can also use the Index, Name, and Category.

Note: - You must name the child Event Frame to prevent getting an error message.

Example:
Event Frame returning the value of Attribute **Area** from the child Event Frame whose Template is **ChildEvent**.



Name	Value
Category: <None>	
Area from Element	12.34855
Area from Element1	12.34855
Area from Element2	12.34855
Rate Value from Element	49.214622497558594
Ratio from Element	3.9854564236793171
Value from child EF	EF

Configuration Item: □
Categories:
Default UOM: <None>
Value Type: String
Value: EF
Data Reference: String Builder

```
.\EventFrames[@Template=ChildEvent]|Area
```

Figure 52

Formula Data Reference

Benefits:

- This is useful for getting a numeric value type from the child Event Frame's Attribute, where the child Event Frame's Attribute does not need to have a PI Point Data Reference.
- Passes time context – the end time of the Event Frame

Limitations:

- Only works for numeric values

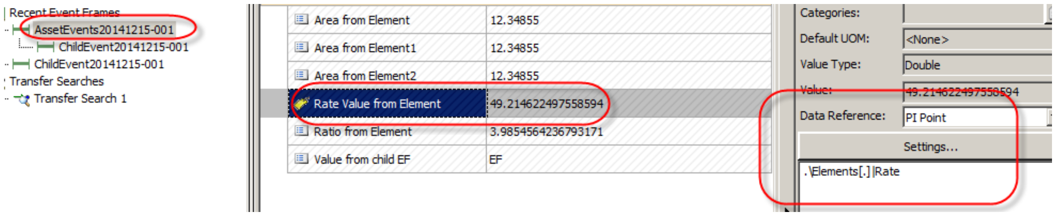
Syntax Examples

A child Event Frame reading a parent Event Frame Attribute value

A=..\|<Attribute>;[A]

Example:

Event Frame returning the value of Attribute **Rate** from the parent Event Frame.

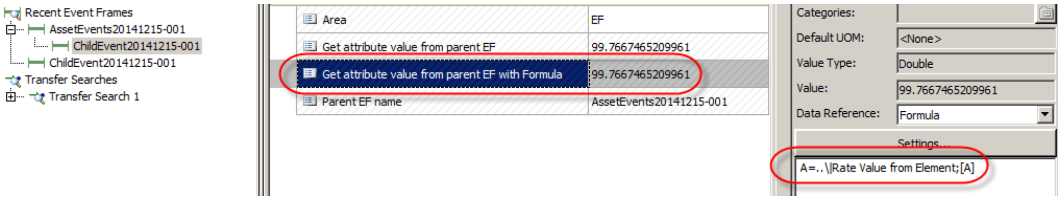


Area from Element	12.34855
Area from Element1	12.34855
Area from Element2	12.34855
Rate Value from Element	49.214622497558594
Ratio from Element	3.9854564236793171
Value from child EF	EF

Categories:
Default UOM: <None>
Value Type: Double
Value: 49.214622497558594
Data Reference: PI Point

```
.\Elements[.]|Rate
```

Figure 53



Area	EF
Get attribute value from parent EF	99.7667465209961
Get attribute value from parent EF with Formula	99.7667465209961
Parent EF name	AssetEvents20141215-001

Categories:
Default UOM: <None>
Value Type: Double
Value: 99.7667465209961
Data Reference: Formula

A=..\|Rate Value from Element;[A]

Figure 54

Note: - In this example the value in the attributes for the parent and child Event Frames are not the same. The reason is that in the parent Event Frame the attribute is configured to get the End Time value from the referenced Element's **Rate** Attribute, and the End Times for the parent and child Event Frames are different. Hence, the End time of the Event Frame is passed to the call for the data.

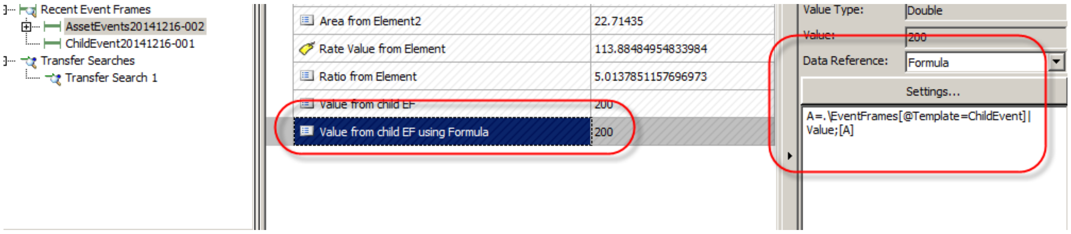
A parent Event Frame reading a child Event Frame Attribute value

.\EventFrames[@Template=*<ChildEFName>*]|*<Attribute>*

Note: - The syntax above is using the Template as the search criteria for the child Element, but you can also use the Index, Name, and Category.

Note: - You must name the child Event Frame to prevent getting an error message.

Example:
Event Frame returning the value of Attribute **Value** from the child Event Frame.



Area from Element2	22.71435
Rate Value from Element	113.88484954833984
Ratio from Element	5.0137851157696973
Value from child EF	200
Value from child EF using Formula	200

Value Type: Double
Value: 200
Data Reference: Formula

A=.\EventFrames[@Template=ChildEvent]|Value;[A]

Figure 55