OpenPlant PowerPID

How to Pass Service from Line to Run and on to Inline Component at Creation and Post Creation

Version 1.4

Initial Release December 14, 2012

How to Pass Service from Line to Run and on to Inline Component at Creation and Post Creation

Trademarks

Copyrights

Table of Contents

Document Version History

Version	Date	Author	Comments
1.0	12/14/2012	Tony DeRosa	Initial Release
1.1	12/17/2012	Tony DeRosa	Removed Value Property Template attribute – not needed for passing property only.
1.3	06/10/2013	Tony DeRosa	Made Notify for piping components class specific; Made SERVICE read only on specific components; Miscellaneous document format changes.
1.4	06/26/2013	Tony DeRosa	Added special provision for inline Instruments (Flow Elements)

Reviewed By	Date	Approved By	Date

Software Versions

Application Name	Version
OpenPlant PowerPID	08.11.10.38 (SS5 and higher)
Bentley Class Editor	From PowerPID install

Assumptions prior to starting

OpenPlant Power PID and Schema Editing

- A general understanding and use of the Class Editor.

Note: A good understanding of OpenPlant PowerPID schemas is also assumed.

Objective

This session will cover the process of using an existing **Associated Item** (In this case SERVICE) and passing this value to an inline component. Specifically you will pass this value to a valve but the same procedure could be used to pass the value to all inline components. This specific method will work with an **ASSOCIATED PROPERTY** and not a standard common property such as Description. Common properties like Description are passed from one class to another using a different method.

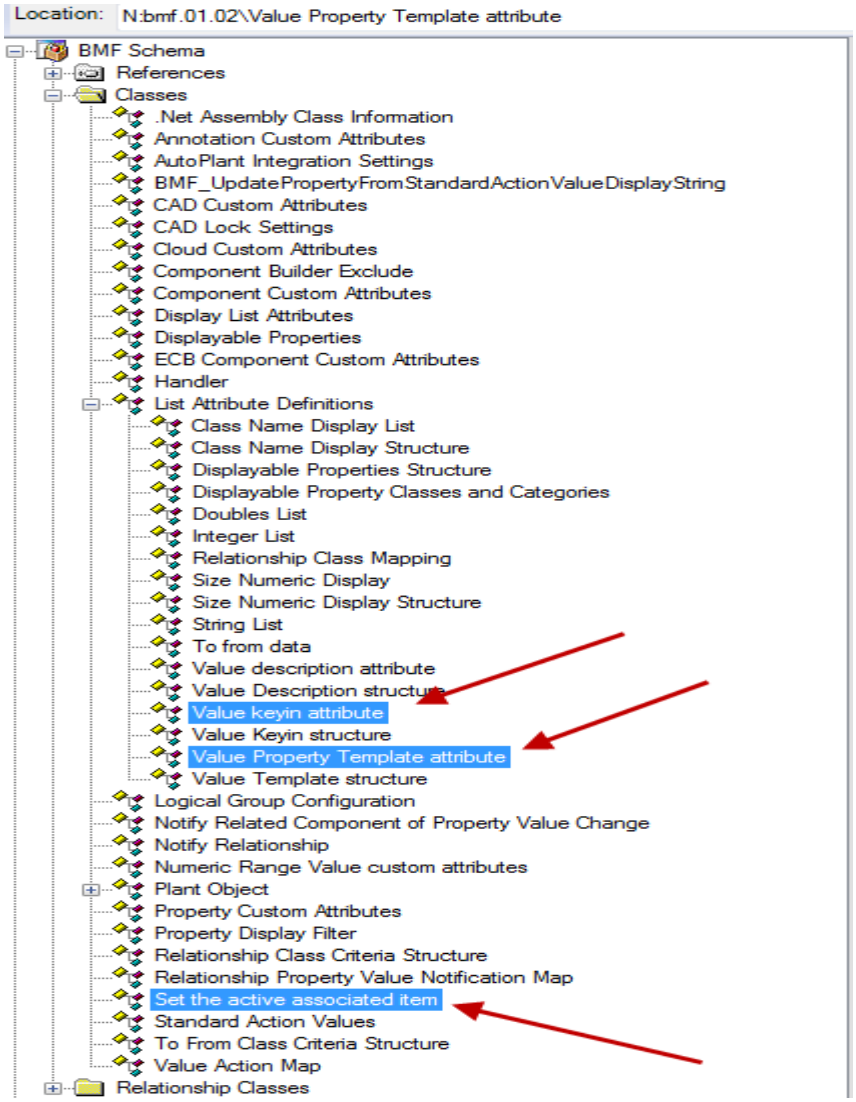
General Overview of Required Administrative Task

OpenPlant PowerPID Schemas

- Verify the BMF default schema.
- Set the class for pipeline to use the correct properties to set Service.
- Pass the Service property on Pipeline to Design Service on Pipe Run.
- Add Design Service to Pipe Run and pass this to Valve.
- Set Valve to use the inherited Service value and not the default Service value.

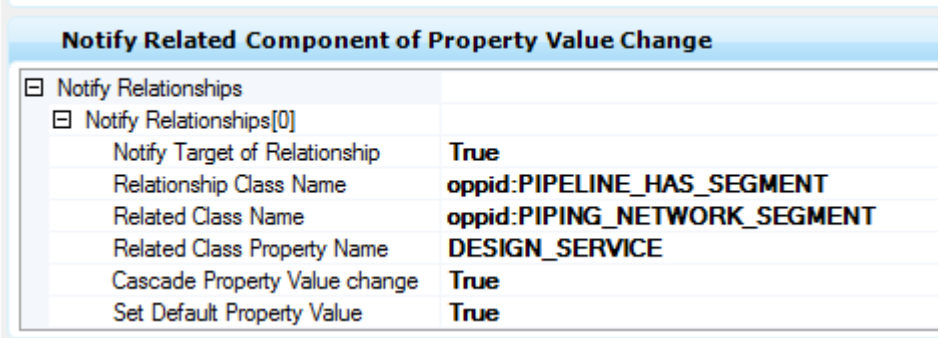
Verifying the default BMF schema

- Load the **Class Editor**
- From within the Class Editor load the **bmf.XX.XX.ecschema.xml** schema
- Expand the **Classes** node tree in the left hand pane and verify that the class names shown below are present in the schema.



Configuring Pipeline to Pass Values

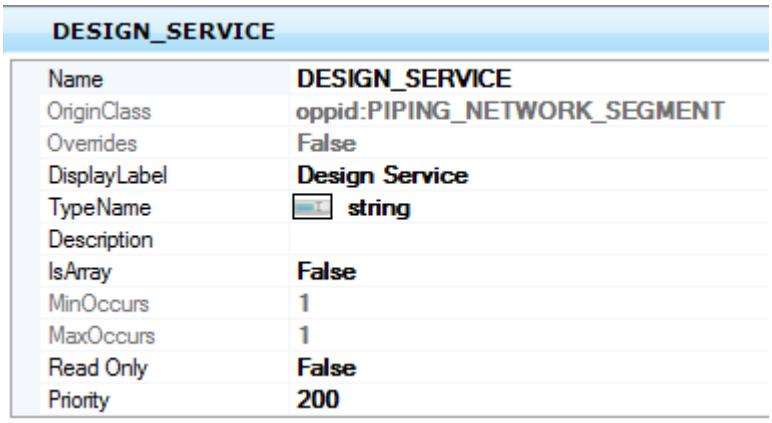
- Load the **Class Editor**
- From within the Class Editor load the **OpenPlant_PID.XX.XX.ecschema.xml** file located in the project schemas folder to be modified.
- Once loaded right click on the schema and from the context menu select **Supplement Schema…** and pick the **OpenPlant_PID_Supplemental_Imperial.XX.XX.ecschema.xml** file.
- Expand the tree in the left hand pane and pick on the **Pipeline** (Class Name: PIPING_NETWORK_SYSTEM) class. Alternatively you can use the search tool to locate the class.
- Select the **Properties** tab.
- Find the **SERVICE** property and pick the **Override** button.
- With the **SERVICE** property selected pick the **Custom Attributes...** button then select **Add/Remove…**
- Add the **Notify Related Component of Property Value Change** (Class Name: BMF_NOTIFY_RELATED_COMPONENT_OF_PROPERTY_VALUE_CHANGE)
- Fill in the custom attribute as shown below.



- **Save** the schemas.

Configuring Pipe Run to Receive and Pass Values

- Continuing in the **Class Editor**
- Select the **Pipe Run** (Class Name: PIPING_NETWORK_SEGMENT) class in the left hand pane and in the right hand pane pick on the **Properties** tab.
- On the **Properties** tab pick the **Add…** button. Add the new property, **DESIGN_SERVICE,** as shown below.



- **Save** the schema.
- Continuing in Class Editor and the on the **DESIGN_SERVICE** property add the **Category**, **Notify Related Component of Property Value Change** and **Property Display Filter** custom attributes.
- Fill in the custom attributes as shown below.

Property Display Filter

IsVisible	**True**
IsReadOnly	**True**

Notify Related Component of Property Value Change

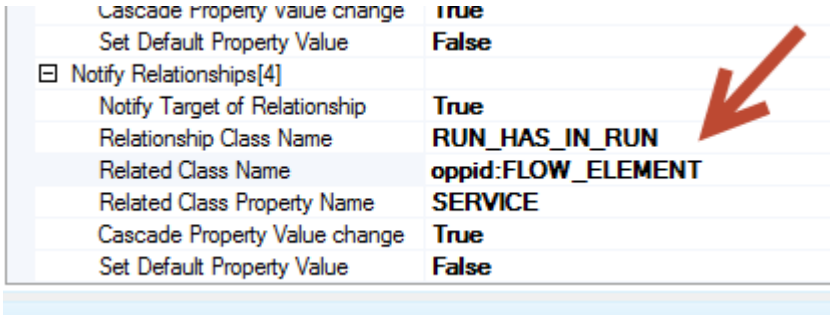
Notify Relationships	
Notify Relationships[0]	
Notify Target of Relationship	**True**
Relationship Class Name	**RUN_HAS_IN_RUN**
Related Class Name	**oppid:PIPING_COMPONENT**
Related Class Property Name	**SERVICE**
Cascade Property Value change	**True**
Set Default Property Value	**False**
Notify Relationships[1]	
Notify Target of Relationship	**False**
Relationship Class Name	**oppid:DATA_CHANGE_CONNECTS_TO_RUN**
Related Class Name	
Related Class Property Name	**SERVICE**
Cascade Property Value change	**True**
Set Default Property Value	**False**
Notify Relationships[2]	
Notify Target of Relationship	**False**
Relationship Class Name	**oppid:END_RUN_CONNECTS_TO_RUN**
Related Class Name	
Related Class Property Name	**SERVICE**
Cascade Property Value change	**True**
Set Default Property Value	**False**
Notify Relationships[3]	
Notify Target of Relationship	**True**
Relationship Class Name	**oppid:OBJECT_HAS_INSTRUMENT**
Related Class Name	
Related Class Property Name	**SERVICE**
Cascade Property Value change	**True**
Set Default Property Value	**False**

Category

Standard	**0**
Name	**SERVICE_VALUE**
DisplayLabel	**Service Value**
Description	**Service Value**
Priority	**200**
Expand	

- **Save** the schemas.

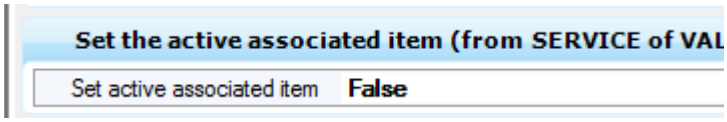
NOTE: If you need to pass SERVICE to inline Instruments such as Flow Elements you will need to add an additional Notify Relationship to the Notify Related Component of Property Value Change custom attribute. See below for this additional notification.



Configuring Valve and Instrument to Receive and Use Values

- Continuing in the **Class Editor**
- Select the **VALVE** class in the left hand panel.
- In the right hand panel pick on the **Properties** tab.
- Navigate to the **Service** property, if the property is greyed out use the **Override** button to localize the property.
- Modify it as shown below by adding the **Set the active associated item** custom attribute. Fill the custom attribute in as shown below.

NOTE: This Custom attribute when set to false, instructs the system to not use the default SERVICE but because we have passed values from Pipe Run to use those.



- **Save** the schemas.
- Repeat this process for **INSTRUMENT**.
- **Save** the schema.

Setting SERVICE Property on the Component to Read Only

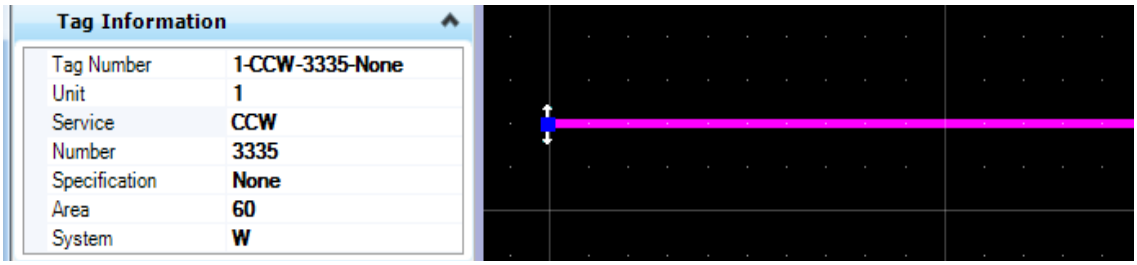
- Continuing in the **Class Editor**
- Select the **VALVE** class in the left hand panel.
- In the right hand panel pick on the **Properties** tab.
- Locate the **SERVICE** property.
- Right click on the property and from the context menu select **Custom Attribute > Add/Remove…**
- Add the **Property Display Filter** custom attribute and fill it in as shown below.

NOTE: Add this to every class listed in the Notify Related Component of Property Value Change custom attribute – VALVE, INSTRUMENT, PIPING_COMPONENT, FLOW_ELEMENT etc.

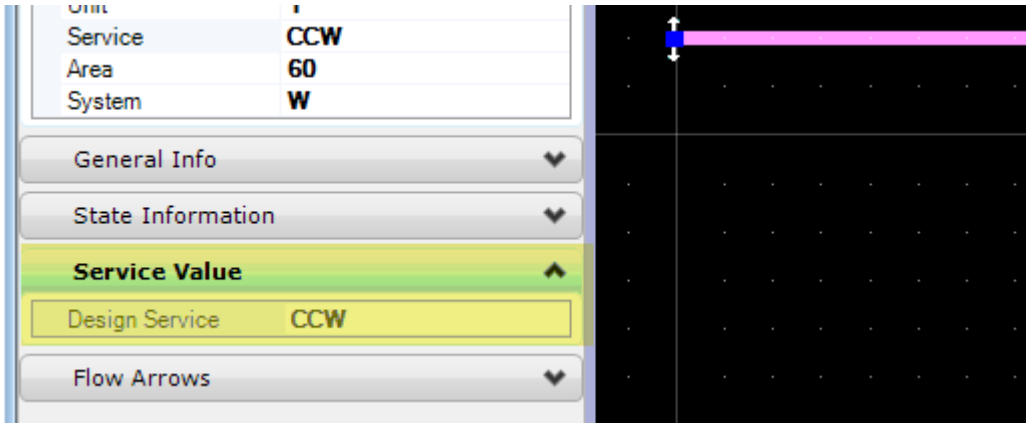
- **Save** the schema.
- Now after placing the valve the **SERVICE** property will no longer be editable on the valve and can only be edited on the **PIPELINE** and then passed to **PIPE RUN** and onto the **VALVE**. You could repeat this process on instruments and other inline components where you do not want the SERVICE property modified on the specific component.

Testing the Results

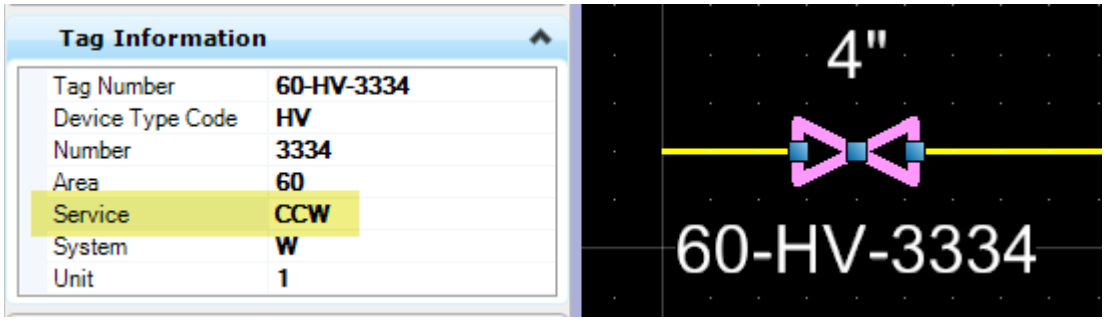
- Launch **OpenPlant PowerPID**.
- Create a new PID file.
- Add several new services to the file. Add "**CW**", "**CCW**", **CWR**". Set the **default** service to **CW**.
- Draw a pipeline and set the service for this new line to **CCW**. Look at the properties for the line and then the run.



- While on the run properties look at the **Service Value** category and verify it matches the **Service** from the Line.

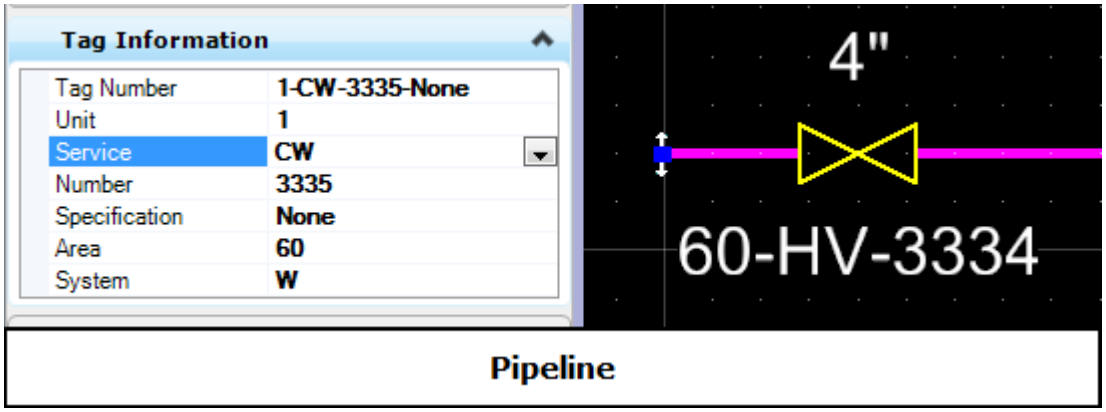


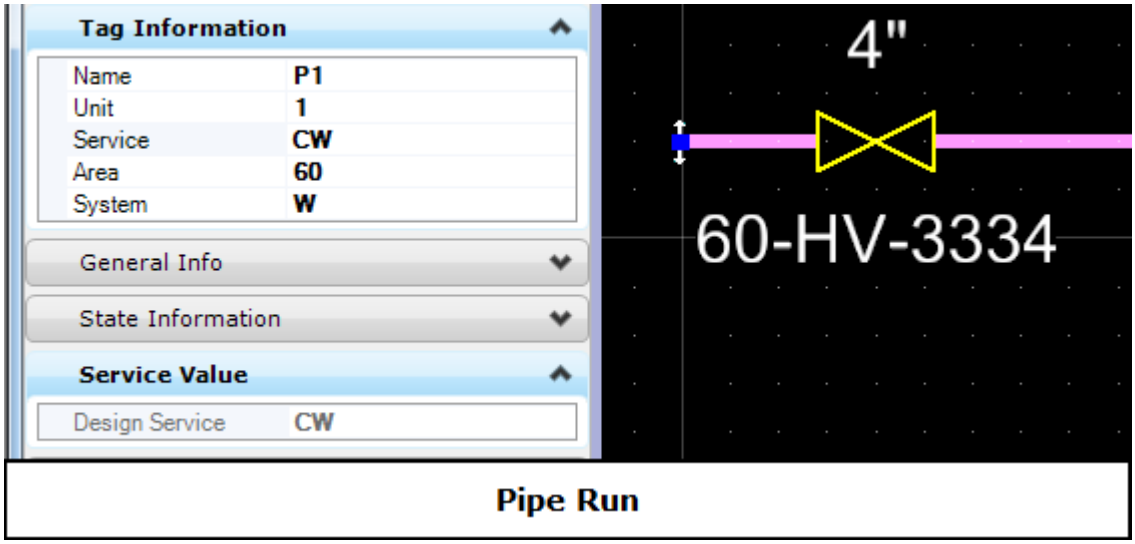
- Insert a valve in the line and note that the valve **SERVICE** is also **CCW** and not the system default of **CW**.

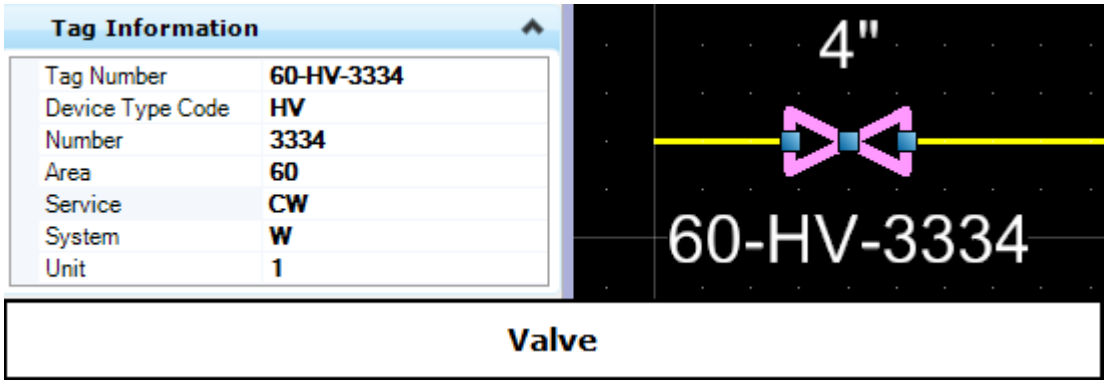


- Edit the **LINE** and change the service to CW.

- Note the change passes from line to run to valve.







- Repeat this test process by adding an instrument to the line. Note that the SERVICE value is populated based on the pipe run.

Notes