OpenPlant PowerPID SS4 Custom Attributes

# Description

Custom attributes allow for the extension of classes and properties and how they interact with OpenPlant applications.

[1 bmf.01.02 3](#_Toc285095063)

[1.1 .Net Assembly Class Information 3](#_Toc285095064)

[1.2 Annotation Custom Attributes 4](#_Toc285095065)

[1.3 CAD Custom Attributes 6](#_Toc285095066)

[1.4 CAD Lock Settings 6](#_Toc285095067)

[1.5 Class Name Display List 8](#_Toc285095068)

[1.6 Cloud Custom Attributes 10](#_Toc285095069)

[1.7 Component Builder Exclude 11](#_Toc285095070)

[1.8 Component Custom Attributes 11](#_Toc285095071)

[1.9 Displayable Property Classes and Categories 13](#_Toc285095072)

[1.10 Doubles List 14](#_Toc285095073)

[1.11 ECB Component Custom Attributes 15](#_Toc285095074)

[1.12 Integer List 15](#_Toc285095075)

[1.13 Notify Related Component of Property Value Change 17](#_Toc285095076)

[1.14 Numeric Range Value custom attributes 18](#_Toc285095077)

[1.15 Property Custom Attributes 20](#_Toc285095078)

[1.16 Property Display Filter 21](#_Toc285095079)

[1.17 Relationship Class Mapping 21](#_Toc285095080)

[1.18 Relationship Property Value Notification Map 22](#_Toc285095081)

[1.19 Size Numeric Display 23](#_Toc285095082)

[1.20 Standard Action Values 24](#_Toc285095083)

[1.21 String List 25](#_Toc285095084)

[1.22 To from data 26](#_Toc285095085)

[1.23 Value description attribute 27](#_Toc285095086)

[1.24 Value keyin attribute 28](#_Toc285095087)

[2 Schematics.01.02 29](#_Toc285095088)

[2.1 BusinessKeyGetValueCriteria 29](#_Toc285095089)

[2.2 Data Change Property Names 30](#_Toc285095090)

[2.3 ECBusinessKeyGroup 31](#_Toc285095091)

[2.4 Edit Handle Settings 31](#_Toc285095092)

[2.5 Flow Arrow Mapping 32](#_Toc285095093)

[2.6 Insertion Settings 33](#_Toc285095094)

[2.7 Jumper Mapping 34](#_Toc285095095)

[2.8 Orientation Data Change Property Name 34](#_Toc285095096)

[2.9 Page Connector Property Name 35](#_Toc285095097)

[2.10 Pointer to spec mapping class 36](#_Toc285095098)

[2.11 Run Auto-Routing 37](#_Toc285095099)

[2.12 Run Mapping 38](#_Toc285095100)

[2.13 Run Property Names 38](#_Toc285095101)

[2.14 Schematics Bubble Class List 39](#_Toc285095102)

[2.15 Schematics CAD Custom Attributes 40](#_Toc285095103)

[2.16 Schematics Component Custom Attributes 41](#_Toc285095104)

[2.17 Schematics Document Information 41](#_Toc285095105)

[2.18 Specification Definition Mapping 42](#_Toc285095106)

[2.19 Specification Definition Mapping for OpenPlant Specs 44](#_Toc285095107)

[3 Pid.01.02 46](#_Toc285095108)

[3.1 Actuator List Class Name 46](#_Toc285095109)

[3.2 Associated Item class list pointer 47](#_Toc285095110)

[3.3 Auto Nozzle Class Name 47](#_Toc285095111)

[3.4 Browse Source Instance 48](#_Toc285095112)

[3.5 Control Valve Class Lists 48](#_Toc285095113)

[3.6 External Data source Definition Mapping 49](#_Toc285095114)

[3.7 Flow Arrow Direction 50](#_Toc285095115)

[3.8 Increment Class Name 51](#_Toc285095116)

[3.9 KKS External Data source Definition Mapping 51](#_Toc285095117)

[3.10 Nozzle Direction values Custom attribute 53](#_Toc285095118)

[3.11 P&ID Display List Attributes 54](#_Toc285095119)

[3.12 Pipe line annotation class for connectors 54](#_Toc285095120)

[3.13 Pipe Run Auto-Routing 55](#_Toc285095121)

[3.14 Related component custom attribute 56](#_Toc285095122)

[3.15 Tag Display Prompt Custom attribute 56](#_Toc285095123)

[3.16 Vessel Heads 57](#_Toc285095124)

[4 EditorCustomAttributes.01.01 59](#_Toc285095125)

[4.1 BooleanDisplay 59](#_Toc285095126)

[4.2 Category 59](#_Toc285095127)

[4.3 ExtendType 61](#_Toc285095128)

[4.4 HideProperty 65](#_Toc285095129)

[4.5 RequiresRefresh 65](#_Toc285095130)

[4.6 StandardValues 66](#_Toc285095131)

[5 Bentley Standard Custom Attributes.01.02 67](#_Toc285095132)

[5.1 Business Key Specification 67](#_Toc285095133)

[5.2 Calculated ECProperty Specification 67](#_Toc285095134)

# bmf.01.02

## .Net Assembly Class Information

|  |  |
| --- | --- |
| Class Name | DOTNET\_ASSEMBLY\_CLASS\_INFO |
| Display Label | .Net Assembly Class Information |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| COMPONENT\_HANDLER  ASSEMBLY\_PATH  CLRName | Component Handler  Assembly Path  .Net Class Name | Bmf:Handler  string  string |
| ADD\_IN\_HANDLERS  ASSEMBLY\_PATH  CLRName | Add In Handlers  Assembly Path  .Net Class Name | Bmf:Handler  string  string |

### Description

This custom attribute defines the settings for the .Net assembly functionality that is used for a component. This custom attribute provides the behavior of a component: how a component is placed and how it is manipulated.

### Properties

#### Component Handler

Component handler defines the .Net behavior used for such things as placement and manipulation. The .Net assembly path defines the location of the DLL file where the .Net assembly is contained in. The .Net Class Name defines the .Net class in the .Net assembly.

#### Add In Handlers

The add in handlers define additional tab(s) for the Tag dialog. By default the dialog only contains a Properties tab. The .Net assembly path defines the location of the DLL file where the .Net assembly is contained in. The .Net Class Name defines the .Net class in the .Net assembly.

### Example

Component for Placing Equipment Cells:

|  |  |  |
| --- | --- | --- |
| Component Handler | Assembly Path:  .Net Class Name: | %PID\_ASSEMBLY\_DIR%Bentley.Plant.Application.Pid.Catalog.dll Bentley.Plant.Application.Pid.Catalog.EquipmentCell |
| Add In Handlers | *Not used* | |

### Example2

PID Document class

|  |  |  |
| --- | --- | --- |
| Component Handler | Assembly Path:  .Net Class Name: | %BMF\_ASSEMBLY\_DIR%bentley.plant.modelingframework.  objectmodel.dll  Bentley.Plant.ObjectModel.Model |
| Add In Handlers[0] | Assembly Path:  .Net Class Name: | %PID\_ASSEMBLY\_DIR%Bentley.Plant.Application.Pid.SettingsAddin.dll  Bentley.Plant.Application.Pid.SettingsAddin.SettingsAddIn |
| Add In Handlers[1] | Assembly Path:  .Net Class Name: | %BMF\_ASSEMBLY\_DIR%Bentley.Plant.ModelingFramework.  BusinessKeyControlsAddins.dll  Bentley.Plant.ModelingFramework.BusinessKeyControls.Addins.  AddInTitleSheets |
| Add In Handlers[2] | Assembly Path:  .Net Class Name: | %BMF\_ASSEMBLY\_DIR%Bentley.Plant.ModelingFramework.  BusinessKeyControlsAddins.dll  Bentley.Plant.ModelingFramework.BusinessKeyControls.Addins.  AddInSettingsAssociatedItems |

## Annotation Custom Attributes

|  |  |
| --- | --- |
| Class Name | BMF\_ANNO\_CUSTOM\_ATTRIBUTES |
| Display Label | Annotation Custom Attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ORIGIN\_OFFSET | Origin Offset | point3d |
| DIRECTION | Direction | point3d |
| VISIBILITY | Visibility | boolean |
| TEXT\_STYLE | Text Style | string |
| PLACEMENT\_COUNT | Placement Count | string |
| ElementTemplate | Element Template | string |

### Description

Defines the settings that are used to place an annotation text string for a component

### Properties

#### Direction

Defines the rotation of an annotation related to the local coordinate system. 1,0,0 is horizontal text. 0,1,0 is vertical text. The 3rd value is for the z axis and is not used in a 2D file.

#### Element Template

Defines the element template used to place the annotation text.

#### Origin Offset

Defines the offset, in sub-units, from the component’s origin where the annotation is placed. This is only used if PLACEMENT\_COUNT=0.

#### Placement Count

Defines how many times the annotation is placed. 0 stands for automatic placement. 1 or more for user to be prompted that number of times for location of annotation.

#### Text Style

Defines a text style for the annotation

#### Visibility

Defines whether or not the annotation is visible. In class Device the visibility for the Tag Number property is turned on. For Fluid Regulators it is turned off.

### Example

SIZE property of class FLUID\_REGULATOR

|  |  |
| --- | --- |
| Origin Offset | 0,0.25 |
| Direction | 1,0 |
| Visibility | True |
| Text Style | PID |
| Placement Count | 0 |
| Element Template | *None* |

## CAD Custom Attributes

|  |  |
| --- | --- |
| Class Name | BMF\_CAD\_CUSTOM\_ATTRIBUTES |
| Display Label | CAD Custom Attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ElementTemplate | Element Template | string |

### Description

Defines CAD settings used to place the component.

### Properties

#### Element Template

Defines the element template used to place the component

### Example

|  |  |
| --- | --- |
| Element Template | Misc\Default |

## CAD Lock Settings

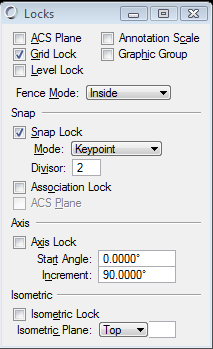
|  |  |
| --- | --- |
| Class Name | BMF\_CAD\_LOCK\_SETTINGS |
| Display Label | CAD Lock Settings |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| AXIS\_LOCK | Axis Lock | boolean |
| AXIS\_LOCK\_ANGLE | Axis Lock Angle | string |
| AXIS\_START\_ANGLE | Axis Start Angle | double |
| GRID\_LOCK | Grid Lock | boolean |
| SNAP\_DIVISOR | Snap Divisor | int |
| SNAP\_LOCK | Snap Lock | boolean |
| SNAP\_MODE | Snap Mode | int |

### Description

With the CAD Lock Settings the user is able to set the PowerDraft lock settings.



### Properties

#### Axis Lock

Enable/disable the axis lock. If set to True the Axis Lock is switched on. If set to False the Axis Lock is switched off.

#### Axis Start Angle

Set the Axis Lock angle. Only applicable if Axis Lock is on.

#### Axis Lock Angle

Set the Axis Lock increment angle. Only applicable if Axis Lock is on.

#### Grid Lock

Enable/disable the Grid Lock. If set to True the Grid Lock is switched on. If set to False the Grid Lock is switched off.

#### Snap Lock

Enable/disable the Snap Lock. If set to True the Snap Lock is switched on. If set to False the Snap Lock is switched off.

#### Snap Mode

Set the snap mode.

Available Snap Modes

|  |  |
| --- | --- |
| 2 | Connect Point |
| 4 | Key point |
| 8 | Nearest |
| 16 | Midpoint |
| 32 | Center |
| 64 | Origin |
| 128 | Bisector |
| 256 | Intersection |
| 512 | Tangent |
| 1024 | Tangent Point |
| 2048 | Perpendicular |
| 4096 | Perpendicular Point |
| 8192 | Parallel |
| 16384 | Point Through |
| 32768 | Point On |
| 65536 | Multi-snap 1 |
| 131072 | Multi-snap 2 |
| 262144 | Multi-snap 3 |

#### Snap Divisor

Set the snap lock divisor. Only applicable if Snap Lock is on.

### Example

|  |  |
| --- | --- |
| Axis Lock Angle | 0 |
| Axis Lock | True |
| Grid Lock | True |
| Snap Lock | True |

## Class Name Display List

|  |  |
| --- | --- |
| Class Name | CLASSNAME\_DISPLAY\_LIST |
| Display Label | Class Name Display List |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS  CLASS\_NAME  DISPLAY\_NAME | Items  Class Name  Display Name | bmf:Class Name Display Structure  string  string |

### Description

This custom attribute is used to associate a few component classes to a compound component.

Example: A control valve has an actuator associated that is placed together with the valve body. The user can select from a list of several available actuators.

### Properties

#### Items

This is an array of items that can provide a list of class names

##### Class Name

Class name

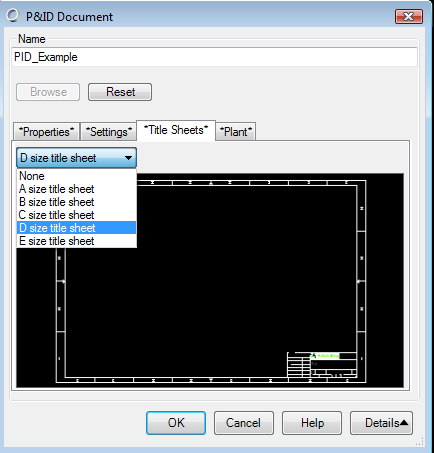
##### Display Name

Display label used with the class name. This display label is used instead of the displayLabel defined on the given class name.

### Example

An example is the Border List. For a P&ID document a number of possible borders are provided. When a new drawing is created the user can select from a list of drawing borders. The border is automatically placed in the drawing and the document properties are automatically displayed in the drawing header. The border can be changed later on.

|  |  |
| --- | --- |
| Item[0]  Display Name  Class Name | None  NONE\_BORDER |
| Item[1]  Display Name  Class Name | A size title sheet  A\_SIZE |
| Item[2]  Display Name  Class Name | B size  B\_SIZE |
| Item[3]  Display Name  Class Name | C size  C\_SIZE |
| Item[4]  Display Name  Class Name | D size  D\_SIZE |
| Item[4]  Display Name  Class Name | E size  E\_SIZE |



## Cloud Custom Attributes

|  |  |
| --- | --- |
| Class Name | BMF\_CLOUD\_CUSTOM\_ATTRIBUTES |
| Display Label | Cloud Custom Attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ARC\_LENGTH | Arc Length | double |
| ARC\_RADIUS | Arc Radius | double |
| ARC\_SWEEP\_ANGLE | Arc Sweep Angle | double |
| DEFAULT\_NUMBER\_OF\_ARCS | Default Number Of Arcs | int |
| INVERT\_CLOUD | Invert Cloud | boolean |
| MARGIN | Margin | double |

### Description

A cloud component is used to emphasize another component. This custom attribute specifies the appearance of the cloud. A cloud consists of a number of arcs that surround an object

### Properties

#### Arc Length

The length of a single arc in the cloud [sub units]

#### Arc Radius

The radius of any of the cloud arcs [sub units]

#### Arc Sweep Angle

The angle from arc start to arc end from the components origin [deg]

#### Default Number of Arcs

Number of arcs used to create cloud around a component during automatic placement when the user has not specified Arc Length and Arc Radius.

#### Invert Cloud

When set to true the arcs are drawn inverted, pointing away from the components origin

#### Margin

The gap between the actual Base Range of the component, which the cloud is being associated with and the actual rectangle range used for the cloud placement.

### Example

|  |  |
| --- | --- |
| ARC\_LENGTH | -1 |
| ARC\_RADIUS | -1 |
| ARC\_SWEEP\_ANGLE | 145 |
| DEFAULT\_NUMBER\_OF\_ARCS | 11 |
| MARGIN | 1 |
| INVERT\_CLOUD | False |

## Component Builder Exclude

|  |  |
| --- | --- |
| Class Name | BMF\_COMPONENTBUILDER\_EXCLUDE |
| Display Label | Component Builder Exclude |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ExcludeSubClass | Exclude Sub Classes | boolean |

### Description

A few components, such as parametric equipment can’t be used with the Component Manager. Classes that have this custom attribute defined won’t appear in the class list of the Component Manager

### Properties

#### Exclude Sub Classes

If set to True the class is excluded from the Component Manager

### Example

## Component Custom Attributes

|  |  |
| --- | --- |
| Class Name | BMF\_COMPONENT\_CUSTOM\_ATTRIBUTES |
| Display Label | Component Custom Attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| INSERT\_ENABLED | Insert Enabled | boolean |
| MIRROR\_ENABLED | Mirror Enabled | boolean |
| MOVE\_ENABLED | Move Enabled | boolean |
| PARENT\_DISPLAY\_ENABLED | Parent Display Enabled | boolean |
| REPEAT\_INSERT | Repeat Insert | boolean |
| REPLACE\_ENABLED | Replace Enabled | boolean |
| ROTATE\_ENABLED | Rotate Enabled | boolean |
| SCALE\_ENABLED | Scale Enabled | boolean |
| TRANSFORM\_RELATED\_ANNOTATION\_ENABLED | Transform Related Annotation Enabled | boolean |

### Description

With the Component Custom Attributes the user can toggle the manipulation behavior of a component

### Properties

#### Insert Enabled

Indicate whether a component can be inserted

#### Mirror Enabled

Indicate whether a component can be mirrored

#### Move Enabled

Indicate whether a component can be moved

#### Parent Display Enabled

Parent component is always displayed. Selected annotation will display parent component in element info, fly over and icons.

#### Repeat Insert

Allow repeat insertion of this component

#### Replace Enabled

Indicate whether a component can be replaced

#### Rotate Enabled

Indicate whether a component can be rotated

#### Scale Enabled

Indicate whether a component can be scaled

#### Transform Related Annotation Enabled

Indicate whether related annotation should be transformed with this component

### Example

The custom attribute is used in class BASE\_EQUIPMENT in the pid.01.01 ECSchema:

|  |  |
| --- | --- |
| INSERT\_ENABLED | False |
| MIRROR\_ENABLED | False |
| MOVE\_ENABLED | True |
| PARENT\_DISPLAY\_ENABLED | True |
| REPEAT\_INSERT | True |
| REPLACE\_ENABLED | True |
| ROTATE\_ENABLED | True |
| SCALE\_ENABLED | False |
| TRANSFORM\_RELATED\_ANNOTATION\_ENABLED | False |

## Displayable Property Classes and Categories

|  |  |
| --- | --- |
| Class Name | DISPLAYABLE\_PROPERTY\_CLASSES |
| Display Label | Displayable Property Classes and Categories |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| DISPLAYABLE\_CLASSES  CLASS\_NAME\_OR\_CATEGORY  ISDISPLAYABLE  ISREADONLY | Displayable Classes  Class Name or Category  Is Class or Category Displayable  Is Read only | bmf:Displayable Properties Structure  string  boolean  boolean |

### Description

In the Element Information dialog box properties are displayed grouped by a category. There are a few MicroStation and system internal properties which would be more disturbing and confusing rather than being informative. Those unwanted categories and class names can be excluded from being displayed by this custom attribute. Currently this custom attribute is only used once and globally in the bmf schema.

### Properties

#### Displayable Classes

##### Class Name or Category

Specify the class name or the category

##### Is Class or Category Displayable

Indicate whether the class or category is displayable

##### Is Read only

Indicate whether the class or category is read only

### Example

Usage in the bmf schema:

|  |  |
| --- | --- |
| Item[0]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnProp  False  True |
| Item[1]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnGraphHeader  False  True |
| Item[2]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnCellProperties  False  True |
| Item[3]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnComplex  False  True |
| Item[4]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnLockedElement  False  True |
| Item[5]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnRotation  False  True |
| Item[6]  Class Name or Category  Is Class or Category Displayable  Is Read only | MstnScale  False  True |
| Item[7]  Class Name or Category  Is Class or Category Displayable  Is Read only | Miscellaneous  False  True |
| Item[8]  Class Name or Category  Is Class or Category Displayable  Is Read only | End Conditions  False  True |
| Item[9]  Class Name or Category  Is Class or Category Displayable  Is Read only | ComponentInformation  False  True |

## Doubles List

|  |  |
| --- | --- |
| Class Name | DOUBLES\_LIST |
| Display Label | Doubles List |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS | Items | double |

### Description

This custom attribute is used to define a double value for a property.

### Properties

#### Items

A double value

### Example

With this double value the spacing between components can be specified. The definition of the pipe run spacing is defined using this custom attribute.

|  |  |
| --- | --- |
| Item[0]  Item | 0.125 |

## ECB Component Custom Attributes

|  |  |
| --- | --- |
| Class Name | BMF\_ECB\_CUSTOM\_ATTRIBUTES |
| Display Label | ECB Component Custom Attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| INSERT\_ENABLED | Insert Enabled | boolean |

### Description

Indicate whether a component can be inserted from the Item Browser (formerly ECBrowser)

### Properties

#### Insert Enabled

Switch on or off whether a component can be inserted from the Item Browser

### Example

Currently all components deriving from class DEVICE can be insterted

|  |  |
| --- | --- |
| Insert Enabled | True |

## Integer List

|  |  |
| --- | --- |
| Class Name | INTEGER\_LIST |
| Display Label | Integer List |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS | Items | int |

### Description

Similar to the Doubles List this custom attribute defines integer values

### Properties

#### Items

An integer value that is defined for a component

### Example

Currently not used

### Logical Group Configuration

|  |  |
| --- | --- |
| Class Name | LOGICAL\_GROUP\_CONFIGURATION |
| Display Label | Logical Group Configuration |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CONTAINED\_CLASS\_NAME\_FILTERS | Contained Class Name Filters | string |
| DEFAULT\_COMPONENTS\_LIST\_NAME | Default Components List Name | string |
| LOGICAL\_RELATIONSHIP\_CLASS\_NAME | Logical Relationship Class Name | string |

### Description

This custom attribute defines the components and relationships of a logical group

### Properties

#### Contained Class Name Filters

This value determines the default types that can be added to a Logical Group.

#### Default Components List Name

Name of the class containing the list of components to insert by default

#### Logical Relationship Class Name

Name of the relationship class used for logical grouping of components

### Example

One of the logical groups used in a P&ID is the Instrument Loop

|  |  |
| --- | --- |
| Default Components List Name | INSTRUMENTS\_FOR\_INSTRUMENT\_LOOP |
| Logical Relationship Class Name | LOOP\_HAS\_COMPONENT |
| Contained Class Name Filters | DEFAULT\_FILTERS\_FOR\_INSTRUMENT\_LOOP |

## Notify Related Component of Property Value Change

|  |  |
| --- | --- |
| Class Name | BMF\_NOTIFY\_RELATED\_COMPONENT\_OF\_PROPERTY\_VALUE\_CHANGE |
| Display Label | Notify Related Component of Property Value Change |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| BMF\_NOTIFY\_RELATIONSHIPS  NOTIFY\_TARGET  RELATIONSHIP\_CLASS\_NAME  RELATED\_CLASS\_NAME  RELATED\_CLASS\_PROPERTY\_NAME  CASCADE\_PROPERTY\_VALUE\_CHANGE  SET\_DEFAULT\_PROPERTY\_VALUE | Notify Relationships  Notify Target of Relationship  Relationship Class Name  Related Class Name  Related Class Property Name  Cascade Property Value change  Set Default Property Value | bmf:Notify Relationship  boolean  string  string  string  boolean  boolean |

### Description

With the Notify Relationship custom attribute a related component can be notified of a property value changed and the related property is updated

### Properties

#### Notify Relationships

##### Notify Target of Relationship

Notify target of relationship.

##### Relationship Class Name

Relationship Class Name that is used for notification

##### Related Class Name

Optional property value used to filter down the classes that get updated. If not filled out all items of the related type will be updated.

##### Related Class Property Name

Property Name of a Related Class that will be updated as result of this property change. Optional when the notified property name is different from the notifying property name

##### Cascade Property Value change

Indicates if the Property Value change should be cascaded to the Related Property Name

##### Set Default Property Value

When set to True the notified property is only updated on instantiation. Later property changes are not notified. When set to False the notified property is always updated

### Example

The design size property of the pipeline (piping network system) updates the size property of the pipe run (piping network segment). This happens only on pipeline creation. Later the pipeline might consist of several pipe runs with different sizes (inserted reducers). A change of the pipeline size then won’t update the pipe run sizes.

|  |  |
| --- | --- |
| Item[0]  Notify Target of Relationship  Relationship Class Name  Related Class Name  Related Class Property Name  Cascade Property Value change  Set Default Property Value | True  PIPELINE\_HAS\_PIPE\_RUN  oppid:PIPING\_NETWORK\_SEGMENT  SIZE  False  True |

## Numeric Range Value custom attributes

|  |  |
| --- | --- |
| Class Name | BMF\_NumericRangeValueCustomAttributes |
| Display Label | Numeric Range Value custom attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| UpperLimit | Upper value limit | double |
| LowerLimit | Lower value limit | double |
| KeyInUpperExceeded | Key-In if upper limit exceeded | string |
| KeyInLowerExceeded | Key-In if lower limit exceeded | string |
| KeyInInRange | Key-In numeric value in range |  |
| ExpressionParser | Expression that can be used to parse property | string |
| ResetInvalidInput | Reset Invalid input of ranges | boolean |
| DisplayMessageonInvalidInput | Display message on invalid input of ranges | boolean |
| ApplyKeyInToRelatedComponents | Apply Key In To Related Components | boolean |
| RelationshipClassNames | Relationship Class Names | string |

### Description

This custom attribute allows a rule to be set on a property that would evaluate a numeric value range.

#### Upper value limit

The lower value limit

#### Lower value limit

The lower value limit

#### Key-In if upper limit exceeded

When the limit is exceeded a key-in is executed

#### Key-In if lower limit exceeded

When the limit is exceeded a key-in is executed

#### Key-In numeric value in range

When the value is in range a key-in is executed

#### Expression that can be used to parse property

A regular expression that is used to parse a property value.

#### Reset Invalid input of ranges

If the input numeric value does not fall within the custom attributes upper and lower ranges, the value will be reset.

#### Display message on invalid input of ranges

Display a message if the input ranges are invalid

#### Apply Key In To Related Components

The command executed by the key-in is also used on components that are related to the current component. E.g. when this custom attribute is used on an instrument loop then the key-in is also used on all loop members (instruments, control valves, etc.).

#### Relationship Class Names

The relationship class name to be used for the key-in

### Example

Example of a pressure check for pipe runs (not configured in standard delivery)

|  |  |
| --- | --- |
| UpperLimit | 200 |
| LowerLimit | 45 |
| KeyInUpperExceeded | pid component assigntemplate Invalid\InvalidUpper |
| KeyInLowerExceeded | pid component assigntemplate Invalid\InvalidLower |
| KeyInInRange | pid component assigntemplate |
| ExpressionParser | (\d+) |
| ResetInvalidInput | False |
| DisplayMessageonInvalidInput | False |
| ApplyKeyInToRelatedComponents | True |
| RelationshipClassNames  Relationship Class Name [0]  Relationship Class Name [1]  Relationship Class Name [2] | RUN\_HAS\_IN\_RUN  OBJECT\_HAS\_INSTRUMENT  REDUCER\_CONNECTS\_TO\_SEGMENT |

## Property Custom Attributes

|  |  |
| --- | --- |
| Class Name | BMF\_PROP\_CUSTOM\_ATTRIBUTES |
| Display Label | Property Custom Attributes |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| DefaultValue | Default Value | string |
| IsGraphical | IsGraphical | boolean |
| FormatString | Format String | string |

### Description

With the Property Custom attribute you can apply a default value for a property

### Properties

#### DefaultValue

The default value. For boolean properties use the string “True” and “False”

#### IsGraphical

Specify whether the value influences the graphical appearance of a component

#### FormatString

A .NET format string

### Example

Setting a default value for the Device Type Code of a pump

|  |  |
| --- | --- |
| DefaultValue | PMP |
| IsGraphical | True |
| FormatString | *None* |

### Example 2

Setting a default value for the number property

|  |  |
| --- | --- |
| DefaultValue | GetNextAvailableValue |
| IsGraphical | True |
| FormatString | {0:D4} |

The reserved word “GetNextAvailablevalue” for the DefaultValue field automatically assigns the next available number to the property. The format string restricts this number to 4 digits.

## Property Display Filter

|  |  |
| --- | --- |
| Class Name | PROPERTY\_DISPLAY\_FILTERS |
| Display Label | Property Display Filter |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| Visible | IsVisible | boolean |
| ReadOnly | IsReadOnly | boolean |

### Description

With the Property Display Filter custom attribute you can set a property to read-only or hide it.

### Properties

#### IsVisible

If set to False the property won’t appear in the Element Info dialog box

#### ReadOnly

If set to True the property appears grayed-out in the Element Info dialog box and cannot be edited

### Example

This custom attribute is used for the IS\_VERTICAL property of the Tower component. Since there cannot be a horizontal tower this property is set to read-only and is also not shown in the Element Info

|  |  |
| --- | --- |
| Visible | False |
| ReadOnly | True |

## Relationship Class Mapping

|  |  |
| --- | --- |
| Class Name | RELATIONSHIP\_CLASS\_MAPPING |
| Display Label | Relationship Class Mapping |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| RELATIONSHIP\_MAPPING  SOURCE\_CONSTRAINT\_NAME  TARGET\_CONSTRAINT\_NAME  SYSTEM\_RELATIONSHIP\_NAME  INFORMATION\_RELATIONSHIP\_NAME | Relationship Mapping  Source Constraint Name  Target Constraint Name  System Relationship Class Name  Information Relationship Name | bmf:Relationship Class Criteria Structure  string  string  string  string |

### Description

These mappings are stored in the **OpenPlant\_PID\_Lists.01.01.ecschema.xml** schema. These mappings allow for addition relationships to be created without application hard coding. A relationship contains a source and a target.

. In some cases the *source* might correspond to the child end of a parent-child relationship, etc. The table below shows some how you might map other semantics onto our *source*/*target* semantics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source** | Origin | ForeignKey | Child | Detail | From |
| **Target** | Destination | Key | Parent | Master | To |

Relationship Strengths

Sometimes a relationship just represents a reference from one object to another, and sometimes it means that one object “owns” another… or at least shares ownership of it.

Referencing

“Referencing” relationships imply no ownership and no cascading deletes when the object on either end of the relationship is deleted. It can be represented in UML as a simple association. For example a Document object may have a reference to the User that last modified it.

Holding

“Holding” relationships imply shared ownership. A given object can be “held” by many different objects, and the object will not get deleted unless all of the objects holding it are first deleted (or the relationships severed.) It can be represented as “aggregation” in UML (which is an association with an open diamond on the source end of the relationship.) For example, an Equipment component may have been imported from two different documents (each document held different business properties that were extracted) and so Each document “holds” the Equipment, and if both documents go way, the Equipment goes away. You could also think of “holding” as “grouping”.

Embedding

“Embedding” relationships imply exclusive ownership and cascading deletes. An object that is the target of an “embedding” relationship may also be the target of other “referencing” relationships, but cannot be the target of any “holding” relationships (the ownership is exclusive, remember.) “Embedding” relationships can be represented by “composition” in UML, an association with a closed diamond on the source end. For example, a Folder “embeds” the Documents that it contains. You could also think of “embedding” as “containment”

**Role Labels**

The RoleLabel defined on the "Source" constraint of the ECRelationshipClass should be used when reading from source to target, e.g. if the role label on the Source constraint was "contains" and the role label on the Target constraint was "is contained by" then we would expect to see:

{source} contains {target}

{target} is contained by {source}

**Cardinality**

Relationship cardinality allows you to restrict the number of instances allowed on each endpoint of a relationship. Each relationship instance connects one source and one target instance so cardinality deals with a collection of relationships of a given type that share a source or target instance. For example let's talk about a DocumentParent relationship which connects a document and its containing folder.

### Properties

#### Relationship Mapping

The relationship mapping definition. You can add multiple mappings with this custom attribute.

##### Source Constraint Name

The name of the source class name.

##### Target Constraint Name

The name of the target class name

##### System Relationship Class Name

The name of system relationship. This denotes that the relationship is used by the system, thus is low level and should not be changed or modified.

##### Information Relationship Name

The name of information relationship. This relationship is more specific, such as Document has object or Equipment has nozzle.

### Example

|  |  |
| --- | --- |
| **Type** | **Value** |
| Source | oppid:EQUIPMENT |
| Target | oppid:NOZZLE |
| System Relationship | BMF\_CONNECT\_POINT\_CONNECTS\_TO\_CONNECT\_POINT |
| Information Relationship | oppid:EQUIPMENT\_HAS\_NOZZLE |

In this example the source is a piece of equipment. That equipment component references the nozzle thru the **EQUIPMENT\_HAS\_NOZZLE** relationship. The cardinality on the equipment is 1. Only one piece of equipment can live in this relationship. The cardinality on the nozzle is N. This indicates that many (or unlimited) number of nozzles can have a relationship with a particular piece of equipment.

<ECRelationshipClass typeName="EQUIPMENT\_HAS\_NOZZLE" displayLabel="Equipment Has Nozzle" isDomainClass="True" strength="referencing" strengthDirection="forward">

<BaseClass>ASSEMBLY\_OF\_INDIVIDUAL</BaseClass>

<Source cardinality="(0,1)" roleLabel="Equipment Has Nozzle" polymorphic="True">

<Class class="EQUIPMENT" />

</Source>

<Target cardinality="(0,N)" roleLabel="Equipment Has Nozzle (reversed)" polymorphic="True">

<Class class="NOZZLE" />

</Target>

</ECRelationshipClass>

The system relationship BMF\_CONNECT\_POINT\_CONNECTS\_TO\_CONNECT\_POINT is the relationship class used to graphically link components. It uses the class BMF\_CONNECT\_POINT. This is a one to one relationship. This relationship and class should **NEVER** be modified.

## Relationship Property Value Notification Map

|  |  |
| --- | --- |
| Class Name | RelationshipPropertyValueNotificationMap (deprecated, not used in any code) |
| Display Label | Relationship Property Value Notification Map |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| RelationshipClassName | Relationship Class Name | string |
| Direction | Direction | string |

## Size Numeric Display

|  |  |
| --- | --- |
| Class Name | SIZE\_NUMERIC\_DISPLAY |
| Display Label | Size Numeric Display |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS  DISPLAY  NOMINAL\_VALUE | Items to be displayed  Display  Nominal Value | bmf:Size Numeric Display Structure  String  String |

### Description

With this custom attribute you can create a sizes list

### Properties

#### Items to be displayed

List of sizes

##### Display

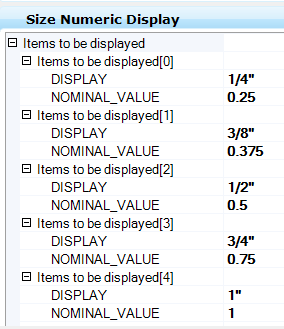
A display label for the sizes

##### Nominal Value

Nominal size value

### Example

Usage in class *Imperial Pipe Sizes*



## Standard Action Values

|  |  |
| --- | --- |
| Class Name | StandardActionValues |
| Display Label | Standard Action Values |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ValueActionMap  DisplayString  Value  KeyIn | Value Action Map  Display String  *[none]*  Key-In | bmf:Value Action Map  string  string  string |
| MustBeFromList | Must Be From List | boolean |
| SELECT\_COMPONENT\_BEFORE\_ACTION | Select Component Before Action | boolean |

### Description

Provides a list of values that are displayed as list in the element info dialog. Optionally a key-in can be executed

### Properties

#### Value Action Map

List of action values

##### Display String

Display String Value that appears in the dropdown menu

##### Value

The value that is assigned to the property

##### Key-In

Key-In associated with value

#### Must Be From List

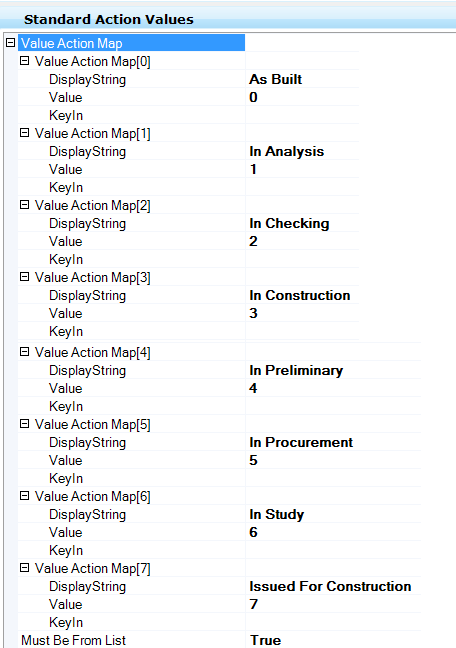
If set to false the user can enter other values.

#### Select Component Before Action

Selects the component that property belongs to before executing any action (Key-in).

### Example

Design State for components. The key-in is not used in the standard delivered product. If the “*Must Be From List”* flag is set to true, you can assign alternative element templates to components when selecting a design state.

****

## String List

|  |  |
| --- | --- |
| Class Name | STRING\_LIST |
| Display Label | String List |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS | Items | String (Array) |

### Description

With the string list you can define a simple list of strings. This list is used in combination with other classes, properties or custom attributes.

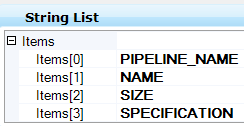
### Properties

#### Items

An array of string values

### Example

Used in the class *Reducer Merge Run Property List*. The string list defines the properties that are displayed in the merge run dialog box



## To from data

|  |  |
| --- | --- |
| Class Name | TO\_FROM\_DATA |
| Display Label | To from data |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| TO\_FROM\_MAPPING  IntermediateRelationship  IntermediateDisplayProperty  DestinationRelationship  AlternateConnectionDisplayString  ShowIntermediateTag | To From Mapping  Intermediate Relationship  Intermediate Display Property  Destination Relationship  AlternateConnectionDisplayString  Show Intermediate Tag | bmf:To From Class Criteria Structure  string  string  string  string  boolean |

### Description

This custom attribute allows for the viewing of source and destination of Pipelines. The relationships are interpreted in code and displayed in tree and node form in a modal dialog

### Properties

#### To From Mapping

##### IntermediateRelationship

Relationship between connecting components

##### IntermediateDisplayProperty

Property to be displayed for the relationship

##### DestinationRelationship

Explain in a few words usage of this property (usually Flow Direction)

##### AlternateConnectionDisplayString

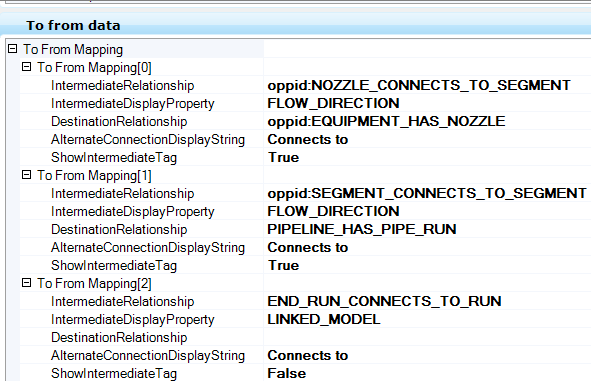
An alternate string to be displayed( e.g. “Connects to”)

##### ShowIntermediateTag

Specify whether the intermediate tag is displayed or not

### Example

To-From data for pipelines



## Value description attribute

|  |  |
| --- | --- |
| Class Name | VALUE\_DESCRIPTION\_ATTRIBUTE |
| Display Label | Value description attribute |
| Schema | Bmf.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| Column1 | Column1 | string |
| Column2 | Column2 | string |
| ITEMS  VALUE  DESCRIPTION | Items  Value  Description | bmf:Value Description structure  string  string |

### Description

User can create a simple value list to assign a property

### Properties

#### Column1

The 1st column header of the list

#### Column2

The 2nd column header of the list

#### Items

One or more items can be added that make up the list values

##### Value

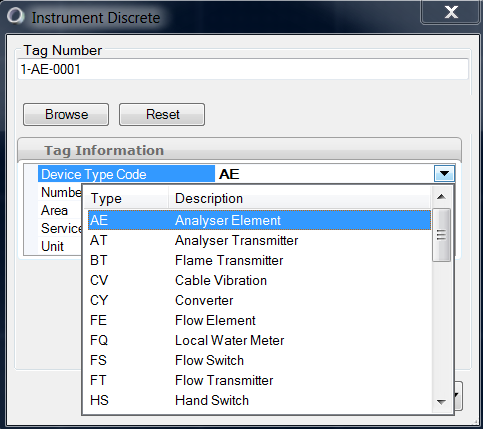
The selectable value which is to be assigned to the property

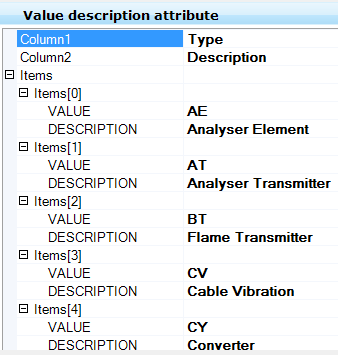
##### Description

A descriptive text of the value

### Example

The value description attribute is used to provide a list of letter codes for instruments





## Value keyin attribute

|  |  |
| --- | --- |
| Class Name | VALUE\_KEYIN\_ATTRIBUTE |
| Display Label | Value keyin attribute |
| Schema | [Schema] |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS  VALUE  KEYIN | Items  Value  Keyin | bmf:Value Keyin structure  string  string |

### Description

Allows for a key-in to be executed (run) or a template change be made, based on a property value input.

### Properties

#### Items

The values, key-ins and templates

##### Value

The matching value that would trigger a key-in or template change

##### Keyin

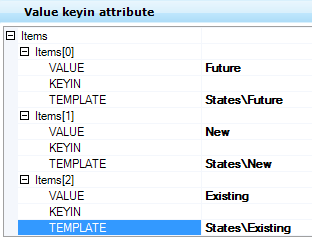
Key-in to be run when the Value is matched with the property that is typed in

##### Template

Template to change to when the Value is matched with the property that is typed in

### Example

Add this to service. So when a Service is changed to Future, the components template will be changed to Future. The templates must be created in the OPPID application (Menu->Element->Element Templates)



# Schematics.01.02

## BusinessKeyGetValueCriteria

|  |  |
| --- | --- |
| Class Name | BusinessKeyGetValueCriteria (deprecated, not used in any code calls) |
| Display Label | BusinessKeyGetValueCriteria |
| Schema | Schematics.01.01 |
| Base Class | sch:SCHEMATICS\_BUSINESSKEY\_DEFINITIONS |

## Data Change Property Names

|  |  |
| --- | --- |
| Class Name | DATA\_CHANGE\_PROPERTY\_NAMES |
| Display Label | Data Change Property Names |
| Schema | schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| MERGE\_RUN\_PROPERTY\_LIST\_NAME | Merge Run Property List Name | string |
| TEXT\_PROPERTY\_NAME\_1 | Text Property Name 1 | string |
| TEXT\_PROPERTY\_NAME\_2 | Text Property Name 2 | string |
| RUN\_TEXT\_PROPERTY\_NAME | Run Text Property Name | string |

### Description

This custom attribute defines the Merge Run Property List for a Data Change component

### Properties

#### Merge Run Property List Name

Specifies the Merge Run Property List Name

#### Text Property Name 1

Property name to store value that will be used as annotation

#### Text Property Name 2

Property name to store value that will be used as annotation

#### Run Text Property Name

Property name to get value from that will be used in annotation

#### ORIENTATION\_PROPERTY\_NAME\_1

Property name to be used in parsing a reducing size.

#### ORIENTATION\_PROPERTY\_NAME\_2

Property name to be used in parsing a reducing size.

#### RUN\_ORIENTATION\_PROPERTY\_NAME

Property name to get value from that will be used in parsing. Used with the ORIENTATION\_PROPERTY\_NAME \_1 and ORIENTATION\_PROPERTY\_NAME\_ 2 property values.

### Example

|  |  |
| --- | --- |
| MERGE\_RUN\_PROPERTY\_LIST\_NAME | REDUCER\_MERGE\_RUN\_PROPERTY\_LIST |
| TEXT\_PROPERTY\_NAME\_1 | MAIN\_SIZE |
| TEXT\_PROPERTY\_NAME\_2 | REDUCING\_SIZE |
| RUN\_TEXT\_PROPERTY\_NAME | SIZE |
| ORIENTATION\_PROPERTY\_NAME\_1 | NOMINAL\_DIAMETER |
| ORIENTATION\_PROPERTY\_NAME\_2 | NOMINAL\_DIAMETER\_REDUCING\_END |
| RUN\_ORIENTATION\_PROPERTY\_NAME | NOMINAL\_DIAMETER |

Reducer example:

SIZE (RUN\_TEXT\_PROPERTY\_NAME) values are retrieved: (after a reducer is placed in a line). There would be two SIZE values.

NOMINAL\_DIAMETER (RUN\_ORIENTATION\_PROPERTY\_NAME) values are retrieved: (after a reducer is placed in a line). There would be two NOMINAL\_DIAMETER values.

The SIZE and NOMINAL\_DIAMETER values are evaluated. The values are they placed in the MAIN\_SIZE, REDUCING\_SIZE, NOMINAL\_DIAMETER and NOMINAL\_DIAMETER\_REDUCING\_END properties.

## ECBusinessKeyGroup

|  |  |
| --- | --- |
| Class Name | ECBusinessKeyGroup |
| Display Label | ECBusinessKeyGroup |
| Schema | Schematics.01.01 |
| Base Class | sch:SCHEMATICS\_BUSINESSKEY\_DEFINITIONS |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| GROUP\_DISPLAY\_NAME | *[none]* | string |
| GROUP\_NAME | *[none]* | string |
| ICON\_NAME | *[none]* | string |

### Description

Used for grouping like class types in various lists. Also used setting icon values. Icon value names would be defined in dgnlibs files.

### Properties

#### GROUP\_DISPLAY\_NAME

The name that is displayed in the Item Browser for the group

#### GROUP\_NAME

The group’s name

#### ICON\_NAME

Name of an icon that is displayed with the group. The icon must exist in one of the referenced dgn libraries.

### Example

|  |  |
| --- | --- |
| GROUP\_DISPLAY\_NAME | Instruments |
| GROUP\_NAME | INSTRUMENT |
| ICON\_NAME | instruments |

## Edit Handle Settings

|  |  |
| --- | --- |
| Class Name | EDIT\_HANDLE\_SETTINGS |
| Display Label | Edit Handle Settings |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| BY\_CONNECT\_POINT | Drag By Connect Point | boolean |
| BY\_RANGE\_CORNERS | Drag By Range Corners | boolean |
| BY\_RANGE | Drag By Range | boolean |

### Description

Sets the manipulation behavior of a component

### Properties

#### Drag By Connect Point

Allows dragging via pre-defined connect points

#### Drag By Range Corners

Allows dragging via range corners

#### Drag By Range

Allows dragging via the eight range points

### Example

Definition of the Schematics Base Component

|  |  |
| --- | --- |
| Drag By Connect Point | True |
| Drag By Range Corners | False |
| Drag By Range | True |

## Flow Arrow Mapping

|  |  |
| --- | --- |
| Class Name | FLOW\_ARROW\_MAPPING |
| Display Label | Flow Arrow Mapping |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| DIRECTIONAL\_CLASSNAME | Directional Class Name | string |
| BI\_DIRECTIONAL\_CLASSNAME | Bi-Directional Class Name | string |

### Description

Determines directional or bi-directional flow arrow to be placed on a pipe run

### Properties

#### Directional Class Name

Directional flow arrow class name

#### Bi-Directional Class Name

Bi-directional flow arrow class name

### Example

Flow Arrow definition in the Imperial supplemental schema

|  |  |
| --- | --- |
| Directional Class Name | oppid:PIPE\_FLOW\_ARROW\_BREAK |
| Bi-Directional Class Name | oppid:PIPE\_FLOW\_ARROW\_BIDIR |

## Insertion Settings

|  |  |
| --- | --- |
| Class Name | INSERTION\_SETTINGS |
| Display Label | Insertion Settings |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| BY\_CONNECT\_POINT | Insert By Connect Point | boolean |
| BY\_RANGE\_CORNERS | Insert By Range Corners | boolean |
| BY\_RANGE | Insert By Range | boolean |
| BY\_DEFAULT | Insert By Default | string |

### Description

Component insertion options.

### Properties

#### Insert By Connect Point

Allows placement via pre-defined connect points

#### Insert By Range Corners

Allows placement via range corners

#### Insert By Range

Allows placement via 8 range points

#### Insert By Default

Default insertion point: origin, connect point: name, range: name

### Example

Settings for the Schematics Base Component

|  |  |
| --- | --- |
| Insert By Connect Point | True |
| Insert By Range Corners | False |
| Insert By Range | False |
| Insert By Default | Origin |

## Jumper Mapping

|  |  |
| --- | --- |
| Class Name | JUMPER\_MAPPING |
| Display Label | Jumper Mapping |
| Schema | Schematics.01.01 |
| Base Class | bmf:bmf\_display\_list\_attributes |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASS\_NAME | Class Name | string |
| ORDER | Priority | int |

### Description

Class priority and jumper to use

### Properties

#### Class Name

Defines the jumper class

#### Priority

Sets the priority of which line type (Major/Minor/Existing Pipeline, Equipment) will be gapped.

### Example

|  |  |
| --- | --- |
| Class Name | PID\_JUMPER |
| Priority | 6 |

## Orientation Data Change Property Name

|  |  |
| --- | --- |
| Class Name | ORIENT\_DATA\_CHANGE\_PROPERTY\_NAMES |
| Display Label | Orientation Data Change Property Name |
| Schema | Schematics |
| Base Classes | sch:DATA\_CHANGE\_PROPERTY\_NAMES |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| *MERGE\_RUN\_PROPERTY\_LIST\_NAME* | *Merge Run Property List Name* | string |
| ORIENTATION\_PROPERTY\_NAME\_1 | Orientation Property Name 1 | string |
| ORIENTATION\_PROPERTY\_NAME\_2 | Orientation Property Name 2 | string |
| RUN\_ORIENTATION\_PROPERTY\_NAME | Run Orientation Property Name | string |
| *TEXT\_PROPERTY\_NAME\_1* | *Text Property Name 1* | string |
| *TEXT\_PROPERTY\_NAME\_2* | *Text Property Name 2* | string |
| *RUN\_TEXT\_PROPERTY\_NAME* | *Run Text Property Name* | string |

(Properties inherited from base class in italics)

### Description

This custom attribute defines the Merge Run Property List for a Data Change component

### Properties

#### Merge Run Property List Name

Specifies the Merge Run Property List Name

#### Text Property Name 1

Property name to store value that will be used as annotation

#### Text Property Name 2

Property name to store value that will be used as annotation

#### Run Text Property Name

Property name to get value from that will be used in annotation

#### ORIENTATION\_PROPERTY\_NAME\_1

Property name to be used in parsing a reducing size.

#### ORIENTATION\_PROPERTY\_NAME\_2

Property name to be used in parsing a reducing size.

#### RUN\_ORIENTATION\_PROPERTY\_NAME

Property name to get value from that will be used in parsing. Used with the ORIENTATION\_PROPERTY\_NAME \_1 and ORIENTATION\_PROPERTY\_NAME\_ 2 property values.

### Example

|  |  |
| --- | --- |
| MERGE\_RUN\_PROPERTY\_LIST\_NAME | REDUCER\_MERGE\_RUN\_PROPERTY\_LIST |
| TEXT\_PROPERTY\_NAME\_1 | MAIN\_SIZE |
| TEXT\_PROPERTY\_NAME\_2 | REDUCING\_SIZE |
| RUN\_TEXT\_PROPERTY\_NAME | SIZE |
| ORIENTATION\_PROPERTY\_NAME\_1 | NOMINAL\_DIAMETER |
| ORIENTATION\_PROPERTY\_NAME\_2 | NOMINAL\_DIAMETER\_REDUCING\_END |
| RUN\_ORIENTATION\_PROPERTY\_NAME | NOMINAL\_DIAMETER |

Reducer example:

SIZE (RUN\_TEXT\_PROPERTY\_NAME) values are retrieved: (after a reducer is placed in a line). There would be two SIZE values.

NOMINAL\_DIAMETER (RUN\_ORIENTATION\_PROPERTY\_NAME) values are retrieved: (after a reducer is placed in a line). There would be two NOMINAL\_DIAMETER values.

## Page Connector Property Name

|  |  |
| --- | --- |
| Class Name | PAGE\_CONNECTOR\_PROPERTY\_NAMES |
| Display Label | Page Connector Property Name |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| TYPE\_PROPERTY\_NAME | Type Property Name | string |
| PIPE\_LINE\_PROPERTY\_NAME | Pipe Line Property Name | string |
| PIPE\_RUN\_PROPERTY\_NAME | Pipe Run Property Name | string |
| RUN\_SIZE\_PROPERTY\_NAME | Run Size Property Name | string |
| PIPE\_SPEC\_PROPERTY\_NAME | Pipe Spec Property Name | string |
| DOCUMENT\_PROPERTY\_NAME | Document Property Name | string |

### Description

Used to display the property names in the offpage connector dialog

### Properties

#### Type Property Name

The property value of the offpage connector type (On or Off)

#### Pipe Line Property Name

The property value of the of the Pipeline (Tag number) Property

#### Pipe Run Property Name

The property value of the of the Pipe Run Property

#### Run Size Property Name

The property value of the of the Size Property

#### Pipe Spec Property Name

The property value of the of the Pipe spec Property

#### Document Property Name

The property value of the of the Document name Property

### Example

Definition in the OffPage Connector class

|  |  |
| --- | --- |
| Type Property Name | OFFPAGE\_CONNECTOR|TYPE|Type |
| Pipe Line Property Name | oppid:PIPING\_NETWORK\_SYSTEM|NAME|Pipeline |
| Pipe Run Property Name | oppid:PIPING\_NETWORK\_SEGMENT|NAME|Piperun |
| Run Size Property Name | oppid:PIPING\_NETWORK\_SEGMENT|SIZE|Size |
| Pipe Spec Property Name | oppid:PIPING\_NETWORK\_SEGMENT|SPECIFICATION|Pipe Spec |
| Document Property Name | oppid:PID\_DOCUMENT|FILE\_NAME|Document |

## Pointer to spec mapping class

|  |  |
| --- | --- |
| Class Name | SPEC\_POINTER |
| Display Label | Pointer to spec mapping class |
| Schema | Schematics.01.01 |
| Base Class | sch:SPEC\_DEFINITION\_CONTAINER |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| SPEC\_MAPPING\_CLASS | Spec mapping class | string |
| SPEC\_MAPPING\_CLASS\_FOR\_OPENPLANT\_SPECS | Spec mapping class for OpenPlant specs | string |
| USE\_ECCLASS\_FOR\_OPENPLANT\_SPECS | Alternate lookup class for OpenPlant specs | string |

### Description

This custom attributes provides pointers to spec mapping classes

### Properties

#### Spec mapping class

Pointer to AutoPlant specs

#### Spec mapping class for OpenPlant specs

Pointer to OpenPlant specs

#### Alternate lookup class for OpenPlant specs

Alternate pointer to OpenPlant spaces

### Example

Spec mapping pointers for valves

|  |  |
| --- | --- |
| Spec mapping class | SPEC\_VALVE\_MAPPING |
| Spec mapping class for OpenPlant specs | SPEC\_VALVE\_MAPPING\_FOR\_OPENPLANT\_SPECS |
| Alternate lookup class for OpenPlant specs | *[not defined]* |

## Run Auto-Routing

|  |  |
| --- | --- |
| Class Name | RUN\_AUTO\_ROUTING |
| Display Label | Run Auto-Routing |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| END\_DISTANCE | Auto-routing End Distance | double |
| PROMPT\_FOR\_DIRECTION\_FROM\_RUN | Prompt For Direction From Run | boolean |

### Description

Properties for auto-routing a Pipe Run

### Properties

#### Auto-routing End Distance

Distance from end of run for auto-routing

#### Prompt For Direction From Run

Flag set to specify whether the user is prompted tom specify the run flow direction

### Example

This custom attribute class is base class of pid:PIPE\_RUN\_AUTO\_ROUTING

## Run Mapping

|  |  |
| --- | --- |
| Class Name | RUN\_MAPPING |
| Display Label | Run Mapping |
| Schema | Schematics.01.01 |
| Base Class | bmf:bmf\_display\_list\_attributes |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASS\_NAME | Class Name | string |
| [Property2] | [DisplayLabel2] | [DataType2] |

### Description

Pointer to the pipe run class

### Class Name

Specifies the Run Mapping class Name

### Example

Usage for the off page connector

|  |  |
| --- | --- |
| Class Name | oppid:PIPING\_NETWORK\_SEGMENT |

## Run Property Names

|  |  |
| --- | --- |
| Class Name | RUN\_PROPERTY\_NAMES |
| Display Label | Run Property Names |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| TEXT\_SIZE\_PROPERTY\_NAME | Run Text Size Property Name | [DataType1] |
| DESIGN\_SIZE\_PROPERTY\_NAME | Run Design Size Property Name | [DataType2] |
| NOMINAL\_DIAMETER\_PROPERTY\_NAME | Run Nominal Diameter Property Name |  |

### Description

Used to defined size property

### Properties

#### Run Text Size Property Name

Specifies the size property name of the pipe run class

#### Run Design Size Property Name

Specifies the design size property name of the pipe run class

#### Run Nominal Diameter Property Name

Specifies the nominal diameter property name of the pipe run class

### Example

Definition in class PIPING\_NETWORK\_SYSTEM

|  |  |
| --- | --- |
| Run Text Size Property Name | SIZE |
| Run Design Size Property Name | DESIGN\_SIZE |
| Run Nominal Diameter Property Name | NOMINAL\_DIAMETER |

## Schematics Bubble Class List

|  |  |
| --- | --- |
| Class Name | SCHEMATICS\_BUBBLE\_CLASS\_LISTS |
| Display Label | Schematics Bubble Class List |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| BODY\_LIST\_NAME | Body List Name | string |
| DIVIDER\_LIST\_NAME | Divider List Name | string |

### Description

Pointers to List Classes for Annotation Bubbles and Instruments.

### Properties

#### Body List Name

Pointer to instrument bubble list

#### Divider List Name

Pointer to instrument divider list

### Example

Definition for Instrument Annotation class

|  |  |
| --- | --- |
| Body List Name | CONTROL\_VALVE\_ANNOTATION\_BODY\_LIST |
| Divider List Name | CONTROL\_VALVE\_ANNOTATION\_DIVIDER\_LIST |

## Schematics CAD Custom Attributes

|  |  |
| --- | --- |
| Class Name | SCHEMATICS\_CAD\_CUSTOM\_ATTRIBUTES |
| Display Label | Schematics CAD Custom Attributes |
| Schema | Schematics.01.01 |
| Base Classes | bmf:BMF\_CAD\_CUSTOM\_ATTRIBUTES |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ElementTemplate | Element Template | string |
| CellName | *[none]* | string |
| CellLibrary | *[none]* | string |
| CELL\_TEXT\_ORIENTATION | Cell Text Orientation | int |

### Description

Schematics CAD General Default Settings

### Properties

#### ElementTemplate

Specifies the Element Template that is assigned to the component

#### CellName

Specifies the cell name of the component (to be omitted for linear components)

#### CellLibrary

Specifies the cell library the cell is contained in (to be omitted for linear components)

#### [Property2]

This custom attribute sets the orientation of text elements within the cell. Possible values: 0 ignore, 1 left to right for horizontal, 2 left to right horizontal bottom to top for vertical

### Example

An instrument function cell component

|  |  |
| --- | --- |
| Element Template | Instrumentation\Instruments |
| CellName | Inst\_Function\_Difference |
| CellLibrary | Instruments.cel |
| Cell Text Orientation | *[not specified]* |

## Schematics Component Custom Attributes

|  |  |
| --- | --- |
| Class Name | SCHEMATICS\_COMPONENT\_CUSTOM\_ATTRIBUTES |
| Display Label | Schematics Component Custom Attributes |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| USE\_PREDEFINED\_POINTS\_FOR\_RANGE | Use Predefined Points for Range | boolean |
| [Property2] | [DisplayLabel2] | [DataType2] |

### Description

Indicate whether to use predefined connect points to determine the component's

### Properties

#### Use Predefined Points for Range

Flag to set whether to use predefined connect points to determine the component's range or not

### Example

Setting for valves

|  |  |
| --- | --- |
| Use Predefined Points for Range | True |

## Schematics Document Information

|  |  |
| --- | --- |
| Class Name | SCHEMATICS\_DOCUMENT\_INFORMATION(Legacy) |
| Display Label | Schematics Document Information |
| Schema | Schematics.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASS | Class | string |
| DESCRIPTION | Description | string |
| INSTANCE | Instance | string |
| SCHEMA | Schema | string |

|  |  |
| --- | --- |
| Description | P&ID Document |
| Instance | Internal |
| Schema | OpenPlant\_PID.01.0X.ecschema.xml |

## Specification Definition Mapping

|  |  |
| --- | --- |
| Class Name | SPEC\_DEFINITION\_MAPPING |
| Display Label | Specification Definition Mapping |
| Schema | Schematics.01.01 |
| Base Class | sch:SPEC\_DEFINITION\_CONTAINER |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| TABLES\_LIST | Tables List | string (array) |
| SQL\_SELECT | Sql Select Where Statement | string |
| PROPERTY\_MAPPING  SOURCE\_FIELDNAME  DESTINATION\_FIELDNAME  SOURCE\_DISPLAYNAME  PORT\_NAME | Property Mapping  Source Field Name  Destination Field Name  Source Display Name  Port Name | sch:Specification Definition Structure  string  string  string  string |
| QUERY\_MAPPING  EC\_QUERY\_FIELDNAME  EC\_QUERY\_FIELDNAME\_SUFFIX\_VALUE  REMOVE\_SUFFIX | Query Mapping fields  EC Property Query Field Name  EC Property Query Field name suffix Value  Remove suffix value | sch:Specification Definition Query Structure  string  string  boolean |
| ALIAS\_MAPPING  PROPERTY  SOURCE\_VALUE  DESTINATION\_VALUE | Alias Mappings  Property Alias Name  Source value to match with destination value  Value to populate in EC Property | sch:Specification Alias Structure  string  string  string |

### Description

Custom attribute for defining how a component reads for an external data source (spec)

### Properties

#### Property Mapping

##### Source Field Name

Source Field Name in which to populate destination field value.

##### Destination Field Name

Destination Field Name from spec database

##### Source Display Name

Source Display Name to be displayed in record selection dialog

##### Port Name

3D Port Name property field name

#### Tables List

List of tables within the spec databases to search for components

#### Sql Select Where Statement

Select statement to query for the spec

#### Query Mapping fields

##### EC\_QUERY\_FIELDNAME

EC Field Name inwhich to get value to query by.

##### VALUE\_FIELDNAME

##### VALUE\_SUFFIX

##### REMOVE\_SUFFIX

#### Alias Mappings

Alternate or alias map name

### Example

Spec mapping for valves

|  |  |
| --- | --- |
| Tables List | valvbw  valve  valvfl  valvscsw |
| Sql Select Where Statement | Where MAIN\_SIZE ='SIZE' and STYPE ='SPECIFIC\_TYPE' and GTYPE = 'GENERAL\_TYPE' |
| Query Mapping fields[0]  EC Property Query Field Name  EC Property Query Field name suffix Value  Remove suffix value  Query Mapping fields[1]  EC Property Query Field Name  EC Property Query Field name suffix Value  Remove suffix value  Query Mapping fields[2]  EC Property Query Field Name  EC Property Query Field name suffix Value  Remove suffix value | SIZE  "  True  SPECIFIC\_TYPE  *[none]*  *[none]*  GENERAL\_TYPE  *[none]*  *[none]* |
| Alias Mappings[0]  Property Alias Name  Source value to match with destination value  Value to populate in EC Property  Alias Mappings[1]  Property Alias Name  Source value to match with destination value  Value to populate in EC Property  Alias Mappings[2]  Property Alias Name  Source value to match with destination value  Value to populate in EC Property  … | END\_CONDITION\_DISPLAY  FL  End\_Condition\_Flanged  END\_CONDITION\_DISPLAY  BW  End\_Condition\_Buttweld  END\_CONDITION\_DISPLAY  SWF  End\_Condition\_Threaded |
| Property Mapping  Source Field Name  Destination Field Name  Source Display Name  Port Name | [not used] |

## Specification Definition Mapping for OpenPlant Specs

|  |  |
| --- | --- |
| Class Name | SPEC\_DEFINITION\_MAPPING\_FOR\_OPENPLANT\_SPECS |
| Display Label | Specification Definition Mapping for OpenPlant Specs |
| Schema | Schematics.01.01 |
| Base Class | sch:SPEC\_DEFINITION\_CONTAINER |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| PROPERTY\_MAPPING  SOURCE\_FIELDNAME  DESTINATION\_FIELDNAME  SOURCE\_DISPLAYNAME  PORT\_NAME | Property Mapping  Source Field Name  Destination Field Name  Source Display Name  Port Name | sch:Specification Definition Structure  string  string  string  string |
| CONDITION\_MAPPING  EC\_QUERY\_FIELDNAME  VALUE\_FIELDNAME  VALUE\_SUFFIX  REMOVE\_SUFFIX | Condition Mapping  Source Field Name  Destination Field Name  Condition Field suffix Value  Remove suffix value | sch:Specification Condition Structure  string  string  string  boolean |
| ALIAS\_MAPPING  PROPERTY  SOURCE\_VALUE  DESTINATION\_VALUE | Alias Mappings  Property Alias Name  Source value to match with destination value  Value to populate in EC Property | sch:Specification Alias Structure  string  string  string |

### Description

Allow a data source to be read, values queried, value return and populated to matching EC properties

### Properties

#### Property Mapping

##### Source Field Name

Source Field Name in which to populate destination field value.

##### Destination Field Name

Destination Field Name from spec database

##### Source Display Name

Source Display Name to be displayed in record selection dialog

##### Port Name

3D Port Name property field name

#### Condition Mapping

##### Source Field Name

##### Destination Field Name

##### Condition Field suffix Value

##### Remove suffix value

#### Alias Mappings

Alternate or alias mapping value

### Example

Spec Valve Mapping for OpenPlant Specs

|  |  |
| --- | --- |
| Property Mapping[0]  Source Field Name  Destination Field Name  Source Display Name  Port Name  Property Mapping[1]  Source Field Name  Destination Field Name  Source Display Name  Port Name  Property Mapping[2]  Source Field Name  Destination Field Name  Source Display Name  Port Name  … | NOMINAL\_DIAMETER  NOMINAL\_SIZE  Size  DESCRIPTION  DESCRIPTION  Short Description  CATALOG\_NAME  Catalog |
| Condition Mapping[0]  Source Field Name  Destination Field Name  Condition Field suffix Value  Remove suffix value | NOMINAL\_DIAMETER  NOMINAL\_DIAMETER |
| Alias Mappings[0]  Property Alias Name  Source value to match with destination value  Value to populate in EC Property  Alias Mappings[1]  Property Alias Name  Source value to match with destination value  Value to populate in EC Property  Alias Mappings[2]  Property Alias Name  Source value to match with destination value  Value to populate in EC Property  … | END\_CONDITION\_DISPLAY  FLANGED  End\_Condition\_Flanged  END\_CONDITION\_DISPLAY  BUTT\_WELD  End\_Condition\_Buttweld  END\_CONDITION\_DISPLAY  SOCKET\_WELD\_FEMALE  End\_Condition\_Socketweld |

# Pid.01.02

## Actuator List Class Name

|  |  |
| --- | --- |
| Class Name | ACTUATOR\_CLASS\_NAME\_LIST |
| Display Label | Actuator List Class Name |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASSNAME | Class Name | string |

### Description

Specifies the list name of actuators for control valves

### Properties

#### Class Name

Pointer to the actuator list

### Example

None available

## Associated Item class list pointer

|  |  |
| --- | --- |
| Class Name | ASSOCIATED\_ITEM\_LIST\_POINTER |
| Display Label | Associated Item class list pointer |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ASSOCIATED\_ITEM\_CLASS | Associated Item class list | string |

### Description

Provides a pointer to the associated item class

### Properties

#### Associated Item class list

Pointer to the associated item class

### Example

Associated item list pointer for valves

|  |  |
| --- | --- |
| Associated Item class list | ASSOCIATED\_ITEMS\_LIST\_FOR\_VALVES |

## Auto Nozzle Class Name

|  |  |
| --- | --- |
| Class Name | AUTO\_NOZZLE\_CLASS |
| Display Label | Auto Nozzle Class Name |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASSNAME | Class Name | string |

### Description

Provides a pointer to a nozzle class

### Properties

#### Class Name

Pointer to the nozzle class

### Example

Auto Nozzle for vessels

|  |  |
| --- | --- |
| Class Name | SINGLE\_NOZZLE |

## Browse Source Instance

|  |  |
| --- | --- |
| Class Name | BROWSE\_SOURCE\_INSTANCE (not used in code) |
| Display Label | Browse Source Instance |
| Schema | Pid.01.01 |
| Base Class | pid:PROP\_DISPLAY\_LISTS |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| SEARCH\_CLASSES | Classes to search | String (array) |

### Description

This is currently not used or utilized in code.

### Properties

#### Classes to search

### Example

## Control Valve Class Lists

|  |  |
| --- | --- |
| Class Name | CONTROL\_VALVE\_CLASS\_LISTS |
| Display Label | Control Valve Class Lists |
| Schema | Pid.01.01 |
| Base Class | pid:PROP\_DISPLAY\_LISTS |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| BODY\_LIST\_NAME | Body List Name | string |
| ACTUATOR\_LIST\_NAME | Actuator List Name | string |
| FAIL\_MODE\_LIST\_NAME | Fail Mode List Name | string |

### Description

This custom attribute contains pointers to list classes for control valves

### Properties

#### Body List Name

Pointer to a list class for control valve body types

#### Actuator List Name

Pointer to a list class for control valve actuators

#### Fail Mode List Name

Pointer to a list class for control valve fail mode symbols

### Example

Control Valve Class Lists for angle control valves in the OpenPlant\_PID\_Supplemental\_ISO.01.01.ecschema

|  |  |
| --- | --- |
| Body List Name | ANGLE\_CONTROL\_VALVE\_LIST |
| Actuator List Name | ANGLE\_CONTROL\_VALVE\_ACTUATOR\_LIST |
| Fail Mode List Name | CONTROL\_VALVE\_FAIL\_MODE\_LIST |

## External Data source Definition Mapping

|  |  |
| --- | --- |
| Class Name | EXTERNAL\_DATSOURCE\_DEFINITION\_MAPPING |
| Display Label | External Data source Definition Mapping |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| DATA\_CONNECTION\_STRING | Data connection string | string |
| FILENAME | File Name of database (can be blank) | string |
| PATH\_LOCATION | Path location of database (can be blank) | string |
| SQL\_SELECT | Sql Select Statement | string |
| PROPERTY\_MAPPING  ECPROPERTY  TABLE\_FIELDNAME | Property Mapping  ECProperty name  Table Field name | pid:Definition Structure  string  string |

### Description

With this custom attribute you can create a dropdown to assign a value to a property list based on a database query

### Properties

#### Data connection string

Data connection string to the database

#### File Name of database

File Name of database (can be blank, e.g. when using a database server)

#### Path location of database

Path location of database (can be blank, e.g. when using a database server)

#### Sql Select Statement

SQL query to create the list

#### Property Mapping

To map database fields with the EC Properties

##### ECProperty name

The name of the EC property

##### Table Field name

The name of the field in the database table

### Example

## <ECProperty propertyName="DEVICE\_TYPE\_CODE" typeName="string" description="code uses to describe the type of component." displayLabel="Device Type Code"> <ECCustomAttributes> <EXTERNAL\_DATSOURCE\_DEFINITION\_MAPPING xmlns="pid.01.01"> <PROPERTY\_MAPPING> <UPDATE\_DEFINITION\_STRUCTURE> <TABLE\_FIELDNAME>Type</TABLE\_FIELDNAME> <ECPROPERTY>DEVICE\_TYPE\_CODE</ECPROPERTY> </UPDATE\_DEFINITION\_STRUCTURE> <UPDATE\_DEFINITION\_STRUCTURE> <TABLE\_FIELDNAME>Description</TABLE\_FIELDNAME> <ECPROPERTY>DESCRIPTION</ECPROPERTY> </UPDATE\_DEFINITION\_STRUCTURE> </PROPERTY\_MAPPING> <DATA\_CONNECTION\_STRING>provider=Microsoft.ACE.OLEDB.12.0; data source = </DATA\_CONNECTION\_STRING> <FILENAME>PickLists.accdb</FILENAME> <PATH\_LOCATION /> <SQL\_SELECT>SELECT Type,Description from EquipmentTypes where Key='DEVICE\_TYPE\_CODE'</SQL\_SELECT> </EXTERNAL\_DATSOURCE\_DEFINITION\_MAPPING> <ExtendType xmlns="EditorCustomAttributes.01.00"> <Name>ShowPickListviewDataBase</Name> </ExtendType> </ECCustomAttributes> </ECProperty>

## Flow Arrow Direction

|  |  |
| --- | --- |
| Class Name | FLOW\_ARROW\_DIRECTION |
| Display Label | Flow Arrow Direction |
| Schema | [Schema] |
| BaseClass | pid:PROCESS\_LIST\_CONTAINER |
| Custom Attributes | EditorCustomAttributes.01.00: StandardValues |

### Standard Values

|  |  |
| --- | --- |
| Name | Value |
| MustBefromList | *not set* |
| ValueMap[0]  DisplayString  Value  ValueMap[1]  DisplayString  Value  ValueMap[2]  DisplayString  Value | Forward  0  Backward  1  Bidirectional  2 |

### Data Structure

No properties defined. The standard values custom attribute directly provides a dropdown list with values defined in the custom attribute. The display string is displayed in the element info property. The value is the actual value stored on the instance.

### Description

This custom attribute maps the flow arrow direction with the flow direction value [???]

## Increment Class Name

|  |  |
| --- | --- |
| Class Name | INCREMENT\_NUMBER\_VALUES |
| Display Label | Increment Class Name |
| Schema | Pid.01.01 |
| Base Class | pid:PROP\_DISPLAY\_LISTS |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| RelationshipClassName | Relationship Class Name | string |
| IncrementPropertyName | Increment Property name | string |
| StringFormat | String Number Format | string |

### Description

This custom attribute allows a number property to be increment based on a relationship to another component

### Properties

#### Relationship Class Name

The relationship to the other component

#### Increment Property name

The property that is to be incremented

#### String Number Format

A number format string

### Example

This custom attribute is used to increment the nozzle number per equipment

|  |  |
| --- | --- |
| Relationship Class Name | EQUIPMENT\_HAS\_NOZZLE |
| Increment Property name | NUMBER |
| String Number Format | {0:D1} |

## KKS External Data source Definition Mapping

|  |  |
| --- | --- |
| Class Name | KKS\_EXTERNAL\_DATSOURCE\_DEFINITION\_MAPPING |
| Display Label | KKS External Data source Definition Mapping |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| DATA\_CONNECTION\_STRING | Data connection string | string |
| FILENAME | File Name of database (can be blank) | string |
| PATH\_LOCATION | Path location of database (can be blank) | string |
| SQL\_SELECT\_1 | Sql Select Statement 1 | string |
| SQL\_SELECT\_2 | Sql Select Statement 2 | string |
| SQL\_SELECT\_3 | Sql Select Statement 3 | string |
| PROPERTY\_MAPPING  ECPROPERTY  TABLE\_FIELDNAME | Property Mapping  ECProperty name  Table Field name | pid:Definition Structure  string  string |
| KEYFIELDNAME | Key field name to be used in sql statements | string |
| FIELD\_DISPLAY\_LIST | Fields to be displayed | string (array) |

### Description

This custom attribute is a special enhancement of the External Data source Definition Mapping custom attribute. It is designed to query for the KKS System Code triplets (KKS Kraftwerks-Kennzeichnungs-System = Power Plant Designation System)

### Properties

#### Data connection string

Data connection string to the database

#### File Name of database

File Name of database (can be blank, e.g. when using a database server)

#### Path location of database

Path location of database (can be blank, e.g. when using a database server)

#### Sql Select Statement 1

SQL query for 1st letter of System Code triplet

#### Sql Select Statement 2

SQL query for 2nd letter of System Code triplet

#### Sql Select Statement 3

SQL query for 3rd letter of System Code triplet

#### Property Mapping

To map database fields with the EC Properties

##### ECProperty name

The name of the EC property

##### Table Field name

The name of the field in the database table

#### Key field name to be used in sql statements

#### Fields to be displayed

### Example

The usage of this custom attribute in the KKS workspace to query the System Calssification

|  |  |
| --- | --- |
| Data connection string | provider=Microsoft.ACE.OLEDB.12.0; data source = |
| File Name of database | KKS\_Codes.accdb |
| Path location of database |  |
| Property Mapping[0]  Table Field name  ECProperty name  Property Mapping[1]  Table Field name  ECProperty name  Property Mapping[2]  Table Field name  ECProperty name | KKS\_Code  SYSTEM\_CLASSIFICATION  TEXT  KKS\_CODE\_DESCRIPTION  REMARK  DESCRIPTION |
| Key field name to be used in sql statements | KKS\_Code |
| Sql Select Statement 1 | SELECT \* from system\_code where LEVEL = 1 |
| Sql Select Statement 2 | SELECT \* from system\_code where LEVEL = 2 AND KKS\_Code LIKE 'KKS\_Code|TOKEN%' |
| Sql Select Statement 3 | SELECT \* from system\_code where LEVEL = 3 AND KKS\_Code LIKE 'KKS\_Code|TOKEN%' |
| Fields to be displayed[0]  Fields to be displayed[1] | KKS\_CODE  TEXT |

## Nozzle Direction values Custom attribute

|  |  |
| --- | --- |
| Class Name | NOZZLE\_DIRECTION\_OPTIONS |
| Display Label | Nozzle Direction values Custom attribute |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| NozzleDirectionValues  StartString  EndString  FieldName | NozzleDirectionOptions  Display  Display  Display | pid:Nozzle Direction structure  string  string  string |

### Description

These customs attribute lets clients define the string value used for direction to be placed in the specified property name. When a pipe run is connected to a nozzle or a piece of equipment (auto nozzle), the direction of the flow (usually To or From) can be set.

### Properties

#### NozzleDirectionOptions

##### StartString

The string value to be used for the start or first point picked

##### EndString

The string value to be used for the end or last point picked

##### FieldName

Property Field name to use to store the StartString and EndString

### Example

This custom attribute in on the NOZZLE class

|  |
| --- |
| StartString To  EndString From  FieldName FLOW\_DIRECTION |

## P&ID Display List Attributes

|  |  |
| --- | --- |
| Class Name | pid\_display\_list\_attributes |
| Display Label | P&ID Display List Attributes |
| Schema | Pid.01.01 |
| Base Class | bmf:bmf\_display\_list\_attributes |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASS\_NAME | Class Name | string |

### Description

This custom attributes is e.g. used to assign one of the available pipe sizes lists (imperial, metric) to the active sizes list.

### Properties

#### Class Name

Defines the display list class

### Example

Assign a sizes list to the active sizes list

|  |  |
| --- | --- |
| Class Name | IMPERIAL\_PIPE\_SIZES |

## Pipe line annotation class for connectors

|  |  |
| --- | --- |
| Class Name | PIPE\_LINE\_ANNOTATION\_CONNECTOR\_CLASS |
| Display Label | Pipe line annotation class for connectors |
| Schema | Pid.01.01 |
| Base Class | pid:PROP\_DISPLAY\_LISTS |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| CLASSNAME | Class Name | string |

### Description

Assigns a pipeline annotation class as annotation for the off page connector

### Properties

#### Class Name

Class name of the pipe annotation class

### Example

|  |  |
| --- | --- |
| Class Name | PIPE\_LINE\_ANNOTATION |

## Pipe Run Auto-Routing

|  |  |
| --- | --- |
| Class Name | PIPE\_RUN\_AUTO\_ROUTING |
| Display Label | Pipe Run Auto-Routing |
| Schema | Pid.01.01 |
| Base Class | sch:RUN\_AUTO\_ROUTING |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| END\_DISTANCE | Auto-routing End Distance | double |
| PROMPT\_FOR\_DIRECTION\_FROM\_RUN | Prompt For Direction From Run | boolean |
| PERPENDICULAR\_DIRECTION\_OUT\_OF\_VESSEL\_EDGE | Use Perpendicular Direction Out of Vessel Edge | boolean |

### Description

These variables are used to define the direction of auto Pipe Run line placement.

### Properties

#### Auto-routing End Distance

Distance from end of run for auto-routing

#### Prompt For Direction From Run

Flag set to specify whether the user is prompted tom specify the run flow direction

#### Use Perpendicular Direction Out of Vessel Edge

Force perpendicular direction of pipe for auto routing

### Example

This custom attribute is used on the Piping Network Segment class

|  |  |
| --- | --- |
| Auto-routing End Distance | 0.5 |
| Prompt For Direction From Run | False |
| Use Perpendicular Direction Out of Vessel Edge | True |

## Related component custom attribute

|  |  |
| --- | --- |
| Class Name | RELATED\_COMPONENT\_CUSTOM\_ATTRIBUTES |
| Display Label | Related component custom attribute |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| RelationShipName |  | string |

### Description

Provides a brows button besides the property name where all components that are related to the actual object via the specified relationship are displayed

### Properties

#### RelationShipName

Specify the relationship name

### Example

Usage in class Piping Network Segment on the Related Inrun Components porperty

|  |  |
| --- | --- |
| RelationShipName | RUN\_HAS\_IN\_RUN |

## Tag Display Prompt Custom attribute

|  |  |
| --- | --- |
| Class Name | TAG\_DISPLAY\_PROMPT\_CUSTOM\_ATTRIBUTE |
| Display Label | Tag Display Prompt Custom attribute |
| Schema | [Schema] |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| Tag\_Display\_Options  AlwaysOn  AlwaysOff  SyncWithTagDisplayVariable | Tag Display Options | pid:Tag display structure options  boolean  boolean  boolean |

### Description

Allows tag prompting to be set On or Off or to Sync with the settings variable (In Setting dialog). This could be set per component

### Properties

#### Tag Display Options

##### AlwaysOn

Tag prompting always on for this component type

##### AlwaysOff

Tag prompting always off for this component type

##### SyncWithTagDisplayVariable

Sync with the settings variable in the setting Dialog ( Tag Prompt)

### Example

This is used on nozzles

|  |  |
| --- | --- |
| AlwaysOn | False |
| AlwaysOff | True |
| SyncWithTagDisplayVariable | False |

## Vessel Heads

|  |  |
| --- | --- |
| Class Name | VESSEL\_HEADS |
| Display Label | Vessel Heads |
| Schema | Pid.01.01 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ITEMS  DISPLAY  ENUM\_VALUE  HORIZONTAL\_ENABLED  VERTICAL\_ENABLED  TOPHEAD  BOTTOMHEAD | Items  Display  Enum value  Horizontal Enabled  Vertical Enabled  Display in Top Head List  Display in Bottom Head List | pid:Vessel Head Structure  string  int  boolean  boolean  boolean  boolean |

### Description

This custom attribute controls whether and where a certain vessel head is available and can be configured.

### Properties

#### Items

##### Display

Display label for the vessel head

##### Enum value

Internal type number (doubles for each new item)

##### Horizontal Enabled

Flag set to enable or disable horizontal type

##### Vertical Enabled

Flag set to enable or disable vertical type

##### Display in Top Head List

Flag set to enable or disable whether the vessel head is available as top head

##### Display in Bottom Head List

Flag set to enable or disable whether the vessel head is available as bottom head

### Example

Usage in the VESSEL\_HEADS class

|  |  |
| --- | --- |
| Items[0]  Display  Enum value  Horizontal Enabled  Vertical Enabled  Display in Top Head List  Display in Bottom Head List | None  1  True  True  True  True |
| Items[1]  Display  Enum value  Horizontal Enabled  Vertical Enabled  Display in Top Head List  Display in Bottom Head List | Hemispherical  2  True  True  True  True |
| ⁞ | ⁞ |
| Items[4]  Display  Enum value  Horizontal Enabled  Vertical Enabled  Display in Top Head List  Display in Bottom Head List | Floating Roof  16  False  True  True  False |

# EditorCustomAttributes.01.01

## BooleanDisplay

|  |  |
| --- | --- |
| Class Name | BooleanDisplay |
| Display Label | BooleanDisplay |
| Schema | EditorCustomAttributes.01.01 |

### Data Structure

|  |  |
| --- | --- |
| Name | Data Type |
| TrueString | string |
| FalseString | string |

### Description

Custom attribute to set a Boolean property to True or False, however the strings displayed for these two values are different

### Properties

#### TrueString

String to be displayed for “True”

#### FalseString

String to be displayed for “False”

### Example

Boolean Display custom attribute used to switch the instrument bubble leader line on and off

|  |  |
| --- | --- |
| TrueString | Leader On |
| FalseString | Leader Off |

## Category

|  |  |
| --- | --- |
| Class Name | Category |
| Display Label | Category |
| Schema | EditorCustomAttributes.01.01 |

### Data Structure

|  |  |
| --- | --- |
| Name | Data Type |
| Standard | int |
| Name | String |
| DisplayLabel | String |
| Description | String |
| Priority | Int |
| Expand | Boolean |

### Description

Defines the category of a property in the Element info and item browser

### Properties

#### Standard

This value should be left empty.

#### Name

The name of the property category

#### DisplayLabel

The display label of the category (this is translated for I18N)

#### Description

A description of the category (this is translated for I18N)

#### Priority

Defines the position and order of the category in the property pane

#### Expand

Display the category expanded or collapsed on startup

### Example

The category “Operating Conditions”

|  |  |
| --- | --- |
| Standard | 0 |
| Name | OPERATING\_CONDITIONS |
| DisplayLabel | Operating Conditions |
| Description | Operating Condition Properties |
| Priority | 100 |
| Expand | *not set* |

☞ Note: The property category is defined or each property that is to be displayed for this category separately, however, only the first instance of this custom attribute that is encountered is used to set the display label and priority. Make sure that you define one category equal for all instances.

☞ Note: The property category is stored in the preferences or OpenPlant PowerPID. Any change of a category priority or introduction of a new category is not correctly displayed until the corresponding GroupPanels.xml file is deleted and newly created. For each project a separate xml file is maintained.

## ExtendType

|  |  |
| --- | --- |
| Class Name | ExtendType |
| Display Label | ExtendType |
| Schema | EditorCustomAttributes.01.01 |

### Data Structure

|  |  |
| --- | --- |
| Name | Data Type |
| Standard | int |
| Name | string |

### Description

With the “ExtendType” custom attribute you can add an extended functionality to a property.

### Properties

#### Standard

*Leave empty*

#### Name

Name of the extension

### Example

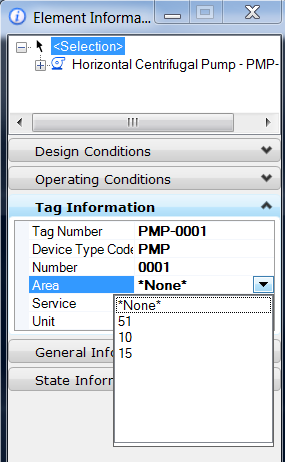
Usage on the PLANT\_AREA property to assign associated items

|  |  |
| --- | --- |
| Standard |  |
| Name | AssociatedItems |

### Available ExtendTypes

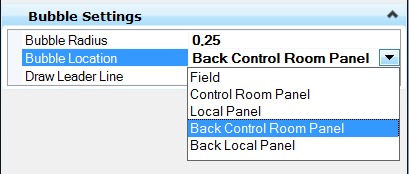
#### AssociatedItems

This extended type provides a dropdown list with the corresponding available associated item.



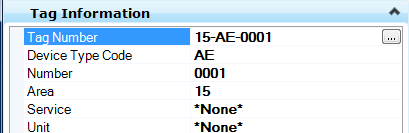
#### BubbleLocation

This extended type provides a dropdown list with the available bubble locations. Selecting a bubble location changes the bubble graphics.



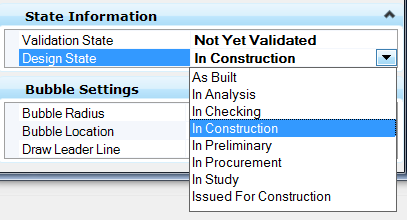
#### BusinessKey

This extended type provides a browse butten on the property that opens the Business Key dialog box.



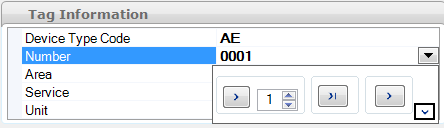
#### DesignStates

This extended type provides a dropdown list with the available design states.



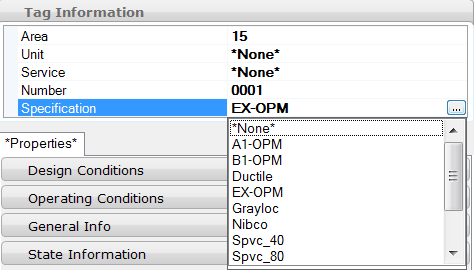
#### GetNextValue

This extended type provides a dropdown dialog box to select the next available number.



#### PipingSpecSelection

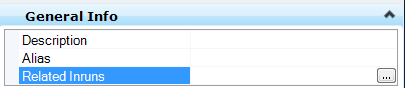
This extended type provides a dropdown list with all the available pipe specifications. The list is generated from the available spec files.

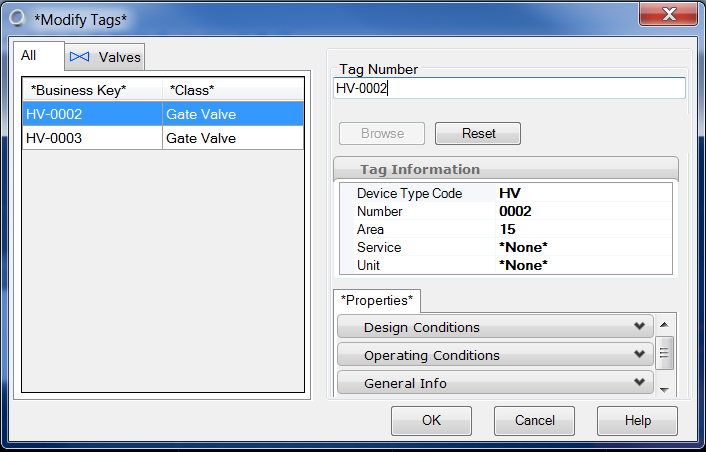


#### RelatedComponent

This extended type provides a browse button that opens a dialog box with all related components.

Example: For a Pipe Run all related inline components (Valves, Control valve, Flow elements, any component that is in the Pipe Run) can be displayed





#### ShowPickListview

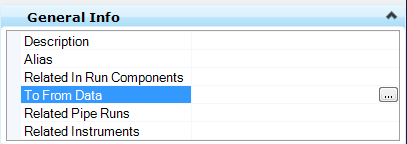
Provides a dropdown list with a defined list of values. The selected value is assigned to the property. The value list must be defined via the Value description attribute custom attribute

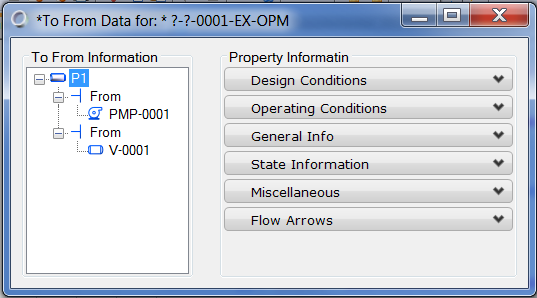
#### ShowPipeSizes

Provides a dropdown list displaying the active sizes list

#### ToFromData

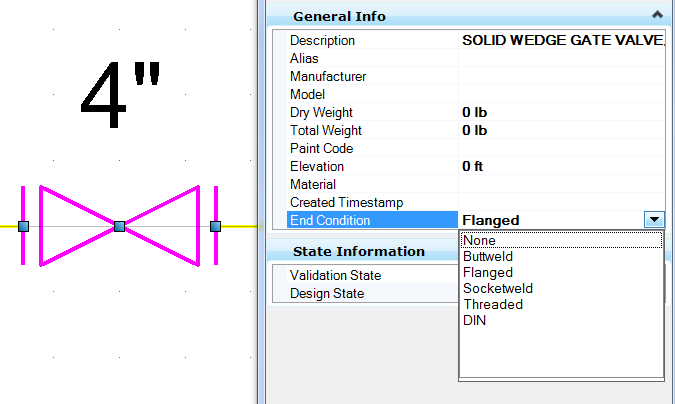
Provides a browse button that opens the “To From” data dialog box showing the components a pipeline is connected to





#### ValveEndConditions

Provides with a dropdown list of the available valve end conditions. The valve end conditions are displayed graphically by symbols.



## HideProperty

|  |  |
| --- | --- |
| Class Name | HideProperty |
| Display Label | HideProperty |
| Schema | EditorCustomAttributes.01.01 |

### Data Structure

|  |  |
| --- | --- |
| Name | Data Type |
| If2D | Boolean |
| If3D | Boolean |

### Description

Hides a property from the property pane in the element information and item browser

### Properties

#### If2D

If set to True it hides the property in an 2D application (e.g. OPPID)

#### If3D

If set to True it hides the property in an 3D application (e.g. OPM)

### Example

In the ISO standard instrument bubbles can be elongated to fit longer annotation strings. This is not the case for imperial standards. The property is therefore hidden

|  |  |
| --- | --- |
| If2D | True |
| If3D | True |

## RequiresRefresh

|  |  |
| --- | --- |
| Class Name | RequiresRefresh |
| Display Label | RequiresRefresh |
| Schema | EditorCustomAttributes.01.01 |

### Data Structure

There are no properties defined for this custom attribute

### Description

This custom attribute is used to trigger property refresh.

### Example

The custom attribute is used on the SIZE property of class PIPE

## StandardValues

|  |  |
| --- | --- |
| Class Name | StandardValues |
| Display Label | StandardValues |
| Schema | EditorCustomAttributes.01.01 |

### Data Structure

|  |  |
| --- | --- |
| Name | Data Type |
| MustBeFromList | Boolean |
| ValueMap  DisplayString  Value | string  string |

### Description

With the StandardValue custom attribute you can create a list of predefined values

### Properties

#### MustBeFrom List

Force to use a value from the list. A user defined key-in is not possible

#### ValueMap

##### DisplayString

A string that is displayed in the dropdown menu

##### Value

The value that is assigned to the property when selected

### Example

Set the vessel internals type.

|  |  |
| --- | --- |
| MustBeFromList | *not set* |
| ValueMap[0]  DisplayString  Value  ValueMap[1]  DisplayString  Value  ValueMap[2]  DisplayString  Value  ValueMap[3]  DisplayString  Value | None  1  Packed  2  Trayed  4  Plate  8 |

The corresponding binary values (0001, 0010, 0100 and 1000) illustrate the internals types that are switched on and off.

# Bentley Standard Custom Attributes.01.02

## Business Key Specification

|  |  |
| --- | --- |
| Class Name | BusinessKeySpecification |
| Display Label | Business Key Specification |
| Schema | Bentley Standard Custom Attributes.01.02 |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| PropertyName | Property Name | [DataType1] |

### Description

Specifies the property that is used for the BusinessKey

### Properties

#### Property Name

Name of Property to use for displaying ECBusinessKey.StringValue

### Example

Business Key Specification

|  |  |
| --- | --- |
| Property Name | NAME |

## Calculated ECProperty Specification

|  |  |
| --- | --- |
| Class Name | CalculatedECPropertySpecification |
| Display Label | Calculated ECProperty Specification |
| Schema | [Schema] |

### Data Structure

|  |  |  |
| --- | --- | --- |
| Name | Display Label | Data Type |
| ECExpression | ECExpression | string |
| FailureValue | Failure value | string |
| ParserRegularExpression | Parser regular expression | string |
| IsDefaultValueOnly | Is calculated as default value only | boolean |
| RequiredSymbolSets | Required ECExpression SymbolSets | string |

### Description

See Bentley.ECObjects.Standards.CalculatedECPropertySpecification. ECCustomAttributeClass for specifying an ECExpression to be used for calculating the value of a given ECProperty.

### Properties

#### ECExpression

The ECExpression used to calculate an ECProperty's value.

#### Failure value

The string representation of the value to use when the value cannot be calculated due to missing dependents or any other reason.

#### Parser regular expression

Will be used when converting from the property value back into its constituent properties. This may be null, and is mostly applicable when the expression has been used to concatentate multiple other property values.

#### Is calculated as default value only

Indicates that the value will only be calculated once on the initial get.

#### Required ECExpression SymbolSets

ECExpressions SymbolSets required for evaluation of the ECExpression

### Example

|  |  |
| --- | --- |
| ECExpression | this.GetRelatedInstance("PLANT\_AREA\_HAS\_NAMED\_ITEM:1:PLANT\_AREA, NAME:?").NAME &amp; "-" &amp; this.DEVICE\_TYPE\_CODE &amp; "-" &amp; this.NUMBER |
| FailureValue | Device |
| ParserRegularExpression | ^[?\w]\*-(?&lt;DEVICE\_TYPE\_CODE&gt;[?\w]\*)-(?&lt;NUMBER&gt;.\*)|(?&lt;NUMBER&gt;.\*) |
| IsDefaultValueOnly | False |
| RequiredSymbolSets | False |
| UseLastValidValueOnFailure | True |