Using Codeless Interference Rules

Introduction:

How to setup and configure:

- Refer to the "Setup" section of "***Smart3D – Interference Management Using Codeless IFC Rules - Setup & Administration Guide***" [called as GUIDE hereinafter] delivered with Smart3D-CodelessIFCRules functionality.
- Perform all the steps explained in **"*Setup*"** section of the GUIDE.
- The steps in here assume that you are working with a plant named SP3DTrain.

Verify Setup & Configuration

To verify that you have setup and configured correctly,

1. Create a log file in the %TEMP% directory with the name ***SP3DTrain***.CodelessIFCRules.log.

2. We will be testing the rules using the "Local Interference Detect". The steps to activate Local IFC are obvious, but also explained in Lab-1.

3. Create a Clashing situation involving two pumps.

4. Review the log file *SP3DTrain*.CodelessIFCRules.log in TEMP. If it has some info generated in it, it means the Codeless IFC Rules functionality is properly setup and configured for the active plant. If you do not see any info in the log file, then you need to review the steps in the "Setup" section.

Lab-1: Ignore objects for Clash Checking (IFC Preprocessing)

This lab explains how to configure various rules available to IGNORE objects for clash checking [pre-processor rule]. For this, the rules should be added under the "**IgnoreObjectsForClashChecking"** tag. Few of the allowed sub-tags are explained in below cases. Refer GUIDE for more information.

Case-A: Using ObjectType/PG/Name/Path/RangeLimit

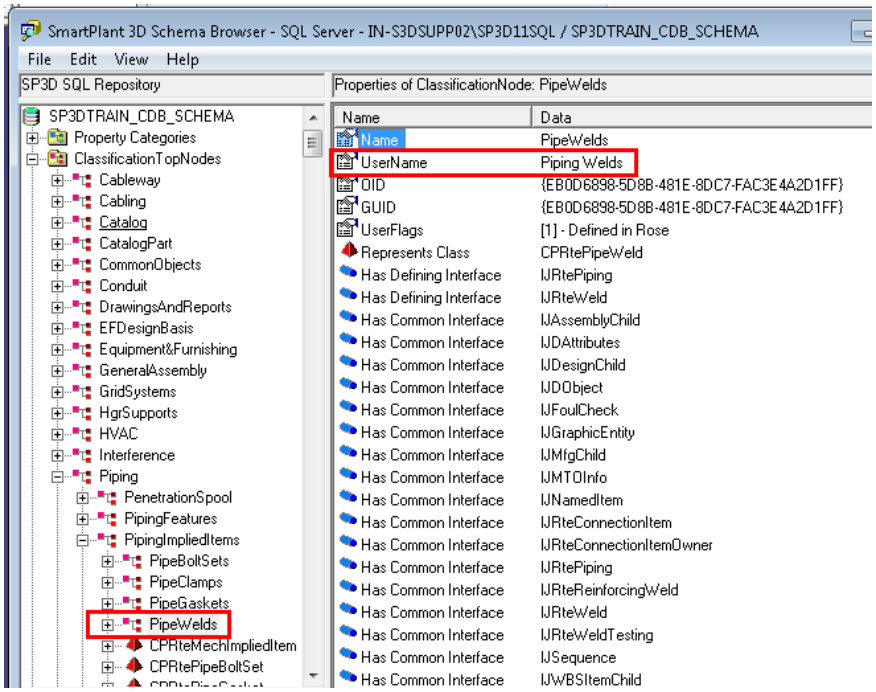
The requirements where the clashes are to be ignored and the rules to achieve those are described below.

1. Open the SP3DTrain.CodelessIFCRules.config file. Add the below XML Tags (rules) under the **IgnoreObjectsForClashChecking** node and save the config file.

2. Add below rule to suppress interference of "Piping Welds" with any object.

<*ByObjectType List="Piping Welds"/>

Tip: To know the exact string to use for the Object Types, you can refer to **"ClassificationTopNodes"** node in **MetaDataBrowser** and specify the value in the "UserName" field.



Add below rule to ignore for clash checking objects which have "Pump" in their name.

<*ByName Like="*Pump*"/>

Add below rule to suppress clashes of ANY object bigger than 100m in range.

<*ByRangeDiagonalLimit Value="100m"/>

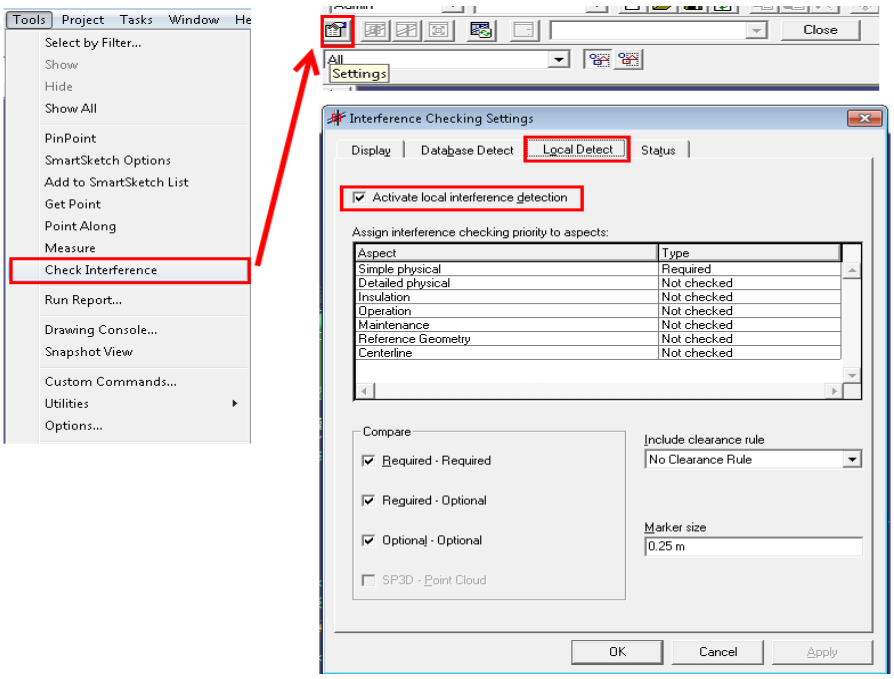
Add below rule to suppress interference of object if it contains "AdminTest" in its system hierarchy.

<*BySystemPath Like="*\AdminTest*"/>

3. Activate the Local Interference Detection:

Perform below steps to activate the Local Interference Detection.

Create a (or Open an existing) session file.
Goto Tools ->Check Interference.
Click 'Settings' on the RibbonBar.
In the "Interference Checking Settings" dialog goto "Local Detect" tab,
check the "Activate local interference detection".
Click OK.



4. Testing the rules added in the Config File:

    1. To test the ByObjectType rule, switch to Piping task and route a Piperun with a Pipe-Elbow-Pipe configuration clashing with something else existing near the elbow. Observe that the clashes with welds are not created since the welds are excluded in preprocessing stage.

    2. To test the ByName rule, switch to Equipment task. Place a pump interfering with the Piping parts modeled in above step. Name it as “IFCTestPump1”. Observe that the clashes with this pump are not created since it is excluded in preprocessing stage.

    3. To test the ByRangeDiagonalLimit rule, switch to Structure task. Place a slab of length = width about 100m. Place a Structural Member clashing with that slab. Observe that the clashes with slab are not created since the slab in question is excluded in preprocessing stage.

    4. To test the BySystemPath rule, create a new Generic System with the name “AdminTest”. Place any two objects clashing with each other under the “AdminTest”. Observe that the clashes involving those objects are not created since they are excluded in preprocessing stage.

    5. Open the SP3DTrain.CodelessIFCRules.log file in the %TEMP% directory and see the information regarding the processing of the Codeless Rules, objects processed, preprocessing results (wherever preprocessing ignored an object) objects. In each case, the log file entries indicate information needed for you to write new rules for such situations. Whenever a rule ignores an object, the rule is reported in the log file.

Limiting the Rules to a given List of Object Types:

You can also limit the rules to work on a list of ObjectTypes you specify. This also enhances the performance.

Limiting the objects for the "BySystemPath" rule is shown here. You can test for the other rules.

We also learn one more feature of Codeless IFC Rules here, that you can test your changes to the config file right away, even without exiting and restarting the session. This topic is explained in "Developer / Testing Conveniences" section of the GUIDE as well. Follow the steps explained in that section and observe the changes made to the config file (to add new rules) are picked up automatically.

1. Switch to Structure task and place two members clashing with each other.
2. Switch to Equipment task and place two equipments clashing with each other.
3. Observe that no Clash is created for the above based on our earlier BySystemPath rule.
4. Now, open the config file. Modify the rule as below and save it.

```
<BySystemPath Like="*\AdminTest\*" LimitToObjectTypes="Member Part Curve, Member Part Linear "/>
```

(Refer to **"Classification TopNodes"** node in **MetaDataBrowser** for the ObjectTypes)

5. Move one of the Structure member and Equipment to clear the clash, and move them back to clashing position. Observe now that the Clash is created for the Equipment but not the member parts. Check the log file in %TEMP% for more info.

Case- B: Using ObjectPG/ Interfaces/Attribute/Attributes

1. Open the SP3DTrain.CodelessIFCRules.config file. Comment the rules we added earlier.

2. Add the below rules under the **IgnoreObjectsForClashChecking** node and save the file.

   Add below rule to suppress clashes of any object with "Misc" or "Ghost" permission groups.

   ```
   <ByObjectPG List="Misc, Ghost"/>
   ```

   Add below rule to suppress clashes of any Piping speciality object which implements *IJUASleeveThickness* interface on the CatalogPart.

   ```
   <ByInterfaces List="IJUASleeveThickness" LimitToObjectTypes="Piping Specialties"/>
   ```

   Add below rule to suppress interference of all Equipment which implements IJUAShaftDims interface and have the shaft diameter property less than 0.15m.

   ```
   <ByAttribute Interface="IJUAShaftDims" Attribute="ShaftDia" Operator="LT" Value="0.15m" Tolerance="5mm"
       LimitToObjectTypes="Equipment"/>
   ```

   Add below rule to suppress clash of objects which have *IJConstructionInfo.ConstructionType=6* (Temporary Objects) and *IJMTOInfo.ReportingType=2 (No MTO objects)* attribute values. (Note that, Numeric Codelist value should be specified with "**:**" before the value. Refer ***GUIDE*** for more information.)

   ```
   <ByAttributes Interface1="IJConstructionInfo"Attribute1="ConstructionType" Operator1="EQ" Value1=":6"
   ```

Interface2="IJMTOInfo" Attribute2="ReportingType" Operator2="EQ" Value2=":2" Comment="Ignore 'Temporary' and 'No MTO' objects" />

3. To test the ByObjectPG rule, switch to Structure task. Place any two members, and set their PG to Misc permission group and move them to a clashing situation.

4. To test the ByInterfaces rule, switch to Piping task, place any pipe specialty which implements IJUASleeveThickness interface and clash with other object.

5. To test the ByAttribute rule, switch to Equipment task and place a pump (Mechanical\Pumps\Pump\PUMP 001A-E) clashing with any other object and change its Shaft Dia attribute value to 0.15m.

6. To test the ByAttributes rule, place an Electrical Transformer (Electrical\Electrical Transformer\ ElectricalTransformer01) clashing with any other object. Observe that the clash is reported.

7. Now, set the below properties on the equipment:
    a. Reporting Requirement – Not to be Reported
    b. Reporting Type – Not to be reported, no-MTO (supplied by others)
    c. Construction Requirement – Existing
    d. Construction Type – Temporary

    Note that the clash is not reported now. For more info, refer to the log file.

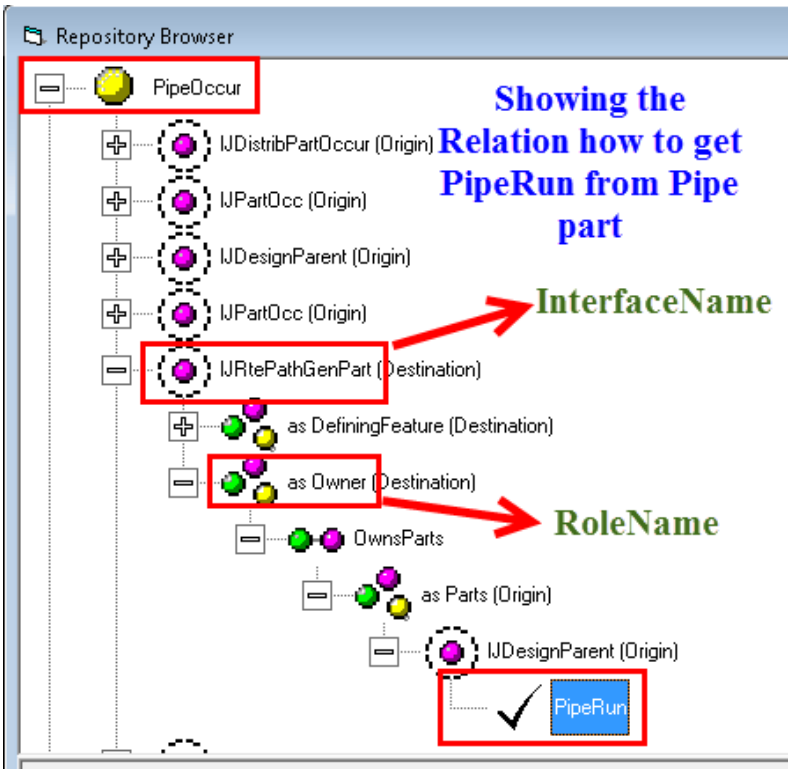
Refer GUIDE to know the allowable values for "Operator" attribute and the syntax for specifying the values for unitted and Codelisted properties in the "Value" attribute.

Using "RelationNavigationPath" option:

We have an advanced option "**RelationNavigationPath**", using which you can also perform the above checks on an object related to the object being evaluated.

For example, to ignore all Parts of Pipe Runs smaller than 1in (25mm) add the below rule to the config file and save it. This rule uses Relationship navigation feature to go and examine the property of related object (PipeRun) while the clash in question involves a Piping part.

```
<ByAttributes LimitToObjectTypes="Piping Components, Pipes, Pipe Instruments, Pipe Specialties"
        RelationNavigationPath="IJRtePathGenPart%Owner[1]"
        Interface1="IJRtePipeRun" Attribute1="NPD" Operator1="LE" Value1="25"
        Interface2="IJRtePipeRun" Attribute2="NPDUnitType" Operator2="EQ" Value2="mm"
            Comment="Ignore Parts on Pipe Runs smaller than 25mm"/>
```



Testing the rules added in the Config File:

1. Switch to Piping task, route a Piperun of '1 in' NPD (any NPD <= 1 in) using any allowable spec such that it clash with any other object.
2. Route another Piperun of '4 in' NPD (any NPD > 1 in) in such a way that it clash with any other object (but not with the Piperun routed in the above step).
3. Observe that the clash is created for the pipes routed in step 3 but not for the one routed in step 2.
4. Review the Log file in TEMP – Also observe that changes made to the config file (to add new rules) are picked up automatically.

Case-C: Using Rules combination and FilesPath

If combinations of any of the above rules have to be used, then use the "ByRulesCombination" tag. Note that all rules included under this tag must satisfy.

Below example rule eliminates underground Pipes & components with 'Existing to be reused in place' construction type.

```
<ByRulesCombination LimitToObjectTypes="Pipes, Piping Components">
    <ByObjectPG List="Underground Piping"/>
    <ByAttribute Interface="IJConstructionInfo" Attribute="ConstructionType" Operator="EQ"  Value=":17" />
</ByRulesCombination>
```

Open the config file, Comment the rules added earlier. Add the above rule and save the file.

To test this rule:

1. Switch to Piping task and route a Piperun such that two pipes are clashing with each other. Set their PG to "Underground Piping". Note that the clash is generated for the pipes.
2. Now, modify the "ConstructionType" of the pipes to 'Existing to be reused in place' and see that the clash is not generated.

To ignore Foreign Files (DGN, R3D etc) to participate in the clash, use the "ByFilePath" tag. This tag must define the following for each attribute it wants to qualify:

```
<ByFilePath Like="*\Roads\*" Type="DGN" />
<ByFilePath Equals="RefFiles\Grade\Ground.dgn" Type="DGN"/>
```

If you are not sure of the type of the file, check the log file where the type info is provided. Specify that value in the 'Type' attribute.

Lab-2: Ignore Interference between objects (IFC post processing)

This lab explains how clashes between two objects can be suppressed based on different rules you can create under the "**IgnoreClashesBetweenObjects"** tag. Few of the allowed sub-tags are explained in below cases. Refer GUIDE for more information.

Case-A: Using ObjectType/PG/Name/Path/Interfaces/Attribute/ RulesCombination

Open the SP3DTrain.CodelessIFCRules.config file. Comment the rules added earlier. Add the below XML Tags (rules) under the **IgnoreClashesBetweenObjects** node and save the config file.

Add below rule to suppress interference of any conduit with another conduit.

```
<ByObjectType List1="Conduits" List2="Conduits"/>
```

Add below rule to suppress interference of any Equipment which belongs to "Misc" permission group with MemberParts which belongs to "Ghost" permission group.

```
<ByObjectPG List1="Misc" LimitToObjectTypes1 = "Equipment"
    List2 = "Ghost" LimitToObjectTypes2 = "Member Part Linear, Member Part Curve" />
```

Add below rule to suppress interference of any object which has "Pump" in its name with other object which has "TestPump" as its name.

```
<ByName Name1Like="*Pump*" Name2Equals = "TestPump" />
```

Add below rule to suppress interference between structural entities which has "*tekla*" in its name.

```
<BySystemPath Name1Like="*\tekla\*" Name2Like="*\tekla\*"
        LimitToObjectTypes="Member Part Linear, Member Part Curve, Slab, Footings, Equipment Foundations,
        Stairs, Ladders, Handrails, Wall Part"/>
```

Add below rule to suppress interference between two Equipments whose one equipment part number is either “PUMP 001A-E” or “PUMP001A_IMP-E” AND other equipment part number is either “PUMP 001AM-E” or “PUMP 001A_IMPM-E". Note that, the pipe character | is used to provide a list of such values (this works only with EQ and NE, Like, Not Like operators)

```
<ByAttribute LimitToObjectTypes="Equipment"
    Interface1="IJDPart" RelationNavigationPath1="IJSmartOccurrence%toSI_ORIG[1]"
    Attribute1="PartNumber" Operator1="EQ" Value1="PUMP 001A-E | PUMP001A_IMP-E"
    Interface2="IJDPart" RelationNavigationPath2="IJSmartOccurrence%toSI_ORIG[1]"
    Attribute2="PartNumber" Operator2="EQ" Value2="PUMP 001AM-E | PUMP 001A_IMPM-E" />
```

Add below set of rules as a combination to suppress clashes between CableTrays and Equipment based on their PGs and Atrribute values.

```
<ByRulesCombination LimitToObjectTypes1="Cable Trays" LimitToObjectTypes2=Equipment">
    <ByObjectPG List1="Underground Cabletray" List2="Underground Equipment"/>
    <ByAttribute
        Interface1="IJConstructionInfo" Attribute1="ConstructionType" Operator1="EQ" Value1=":17"
        Interface2="IJConstructionInfo" Attribute2="ConstructionType" Operator2="EQ" Value2=":17"/>
    <ByAttribute
        Interface1="IJDObject" Attribute1="ApprovalStatus" Operator1="EQ" Value1=":8"
        Interface2="IJDObject" Attribute2="ApprovalStatus" Operator2="EQ" Value2=":8"/>
</ByRulesCombination>
```

As explained in previous lab, to test the above rules, for each rule, place both the objects according to the rules framed and clash with each other. You can observe no clash markers are created for these cases.

Case-B: Using ByCommonAttributeValues

Open the SP3DTrain.CodelessIFCRules.config file. Comment the rules added earlier.
Add the below rules under the **IgnoreClashesBetweenObjects** node and save the file.

Add below rule to suppress interference between any two MemberParts if their MemberSystems are under same parent.

```
<ByCommonAttributeValues ObjectTypesList="Member Part Linear, Member Part Curve"
        RelationNavigationPath =
        "ISPSMemberSystemChild%MemberDesignParent[1]:ISPSMemberSystemSystemChild%MemberSystSyst
        emParent[1]" Interface1="IJNamedItem" Attribute1="Name" />
```

Add below rule to suppress interference between WallParts and Slab if the WallSystem and the Slab are under same system.

```
<ByCommonAttributeValues ObjectTypesList1="Wall Part" ObjectTypesList2="Slab"
RelationNavigationPath1 = IJSystemChild%SystemParent[1]:IJSystemChild%SystemParent[1]"
RelationNavigationPath2 = "IJSystemChild%SystemParent[1]"
Interface1="IJNamedItem" Attribute1="Name"  />
```

To test the above rules:

1. Switch to Structure task. Under the "AdminTest" Generic System (created in Lab1), create a new StructuralSystem with the default name.
2. Place two members with the above newly created StructuralSystem as parent and clash them with each other.
3. Place a wall and slab under the same system clashing each other.
4. In both the above cases, clash markers are not created.
5. Now, place another member clashing with the slab and note that the clash is generated.

Case-C: Using ByCommonAncestor/ ByObjectTypeAndAspect

Open the SP3DTrain.CodelessIFCRules.config file. Comment the rules added earlier.
Add the below rules under the **IgnoreClashesBetweenObjects** node and save the file.

Rule to eliminate clashes between the Members which are under the same UnitSystem.

```
<ByCommonAncestor LimitToObjectTypes="Member Part Linear, Member Part Curve"
AncestorInterfaces="IJUnitSystem" />
```

Rule to eliminate clashes between the maintenance aspects of the Equipment

```
<ByObjectTypeAndAspect ObjectType1="Equipment" Aspect1="Maintenance" ObjectType2=" Equipment"
Aspect2="Maintenance"/>
```

Or, more concisely (if both objectType1/2 are same, and Aspect1/2 are same), as

```
<ByObjectTypeAndAspect ObjectTypes=" Equipment" Aspects="Maintenance"/>
```

To test the above rules:

1. Switch to Structure task. Under the "AdminTest" Generic System (created in Lab1), create a new UnitSystem with the default name.
2. To test the ByCommonAncestor rule, place two members with the above newly created UnitSystem as parent and clash them with each other.
3. To test the ByObjectTypeAndAspect rule, switch to Equipment & Furnishings task. To make the other aspects visible, go to Format->View. In the "Selected Aspects", select the 'Maintenance', 'Operation' aspects.
4. Place any two equipment under the "AdminTest" GenericSystem. Using "Place Shape" command, add shape to both the Equipment selecting the "Display Aspect" as "Maintenance". See that both the aspects are clashing with each other and note that no clash is created.
5. Similarly, you can add "Insulation" aspect to the Equipment clashing with each other and observe that the clash is created.

Lab-3: Set Clash Type on interference object

So far, we have seen how to suppress the clashes. This lab explains how to set the type of clash based on the type of clashing objects (specifiable by comma separated list of Interfaces) and the geometric Aspect involved in the clash using **ByDefiningInterfacesAndAspect** tag.

Open the SP3DTrain.CodelessIFCRules.config file. Comment the rules added earlier. Add the below rules under the "**SetClashType"** node and save the file.

Below Rule sets the clash type as SOFT for clash involving Piping 'Simple Physical' with Equipment 'Maintenance' aspect.

```
<ByDefiningInterfacesAndAspect
        DefiningInterfacesList1="IJRtePiping"  Aspect1="Simple Physical"
        DefiningInterfacesList2="IJEquipmentFurnishings"    Aspect2="Maintenance"
        Type = "SOFT"/>
```

Rule to set the clash type as HARD for the Piping objects (any aspect) clashing with any object's 'Operation' aspect.

```
<ByDefiningInterfacesAndAspect
        DefiningInterfacesList1 = "IJRtePiping" Aspect1="*"
        DefiningInterfacesList2 ="*" Aspect2="Operation"
        Type="HARD"/>
```

To test the above rules:

1. Switch to Equipment & Furnishings task. Place any equipment and add 'Maintenance', 'Operation' aspects to it.
2. Switch to Piping task and route a PipeRun such that the 'Simple Physical' aspect of the pipe is clashing with the 'Maintenance' and 'Operation' aspects of the Equipment.
3. Observe the clash created for Pipe and Equipment's Maintenance aspect is SOFT (yellow in color) and the clash created for Pipe and Equipment's Operation aspect is HARD (red in color). For more info, refer to the log file in the %TEMP% directory.

Note: Combination of LimitToObjectTypes option and wild cards will let you does something like below to yield the same result as the example rules above.

```
<ByDefiningInterfacesAndAspect
        DefiningInterfacesList1="*" Aspect1="Simple Physical" LimitToObjectTypes1="Pipes, Piping *"
        DefiningInterfacesList2="*"Aspect2="Maintenance" LimitToObjectTypes2="Equipment"
        Type = "SOFT"/>
<ByDefiningInterfacesAndAspect
        DefiningInterfacesList1 = "*" Aspect1="*" LimitToObjectTypes1="Pipes, Piping *"
        DefiningInterfacesList2 ="*" Aspect2="Operation" LimitToObjectTypes2="*"
        Type="HARD"/>
```