|  |
| --- |
| Copy by Family Attribute Copy Rule |
| copy\_by\_family\_attribute\_copy\_rule.docx |
| Catalysts |

Table of Contents

[1. Revision History 3](#_Toc247105303)

[2. Introduction 3](#_Toc247105305)

[3. References 3](#_Toc247105307)

[4. Definitions 3](#_Toc247105309)

[5. Default Attribute Copy Behavior 3](#_Toc247105317)

[6. Attribute Copy Rule 4](#_Toc247105320)

[6.1 What is in it? 4](#_Toc247105322)

[7. Structure 4](#_Toc247105324)

[7.1 Example 4](#_Toc247105326)

[7.2 Write a Attribute Copy Rule for each Business Object 4](#_Toc247105328)

[7.3 Attributes for which the rule decides the target value 5](#_Toc247105330)

[7.4 Rule logic for an attribute 5](#_Toc247105331)

[7.4.1 Property Get or Getter 7](#_Toc247105332)

[7.4.2 Method Parameters 7](#_Toc247105334)

[7.4.3 AttributeRuleResult 7](#_Toc247105336)

[8. Catalog 7](#_Toc247105338)

[8.1 When to add a new Attribute Copy Rule? 7](#_Toc247105340)

[8.2 How to add the new rule? 7](#_Toc247105341)

[8.2.1 Information required to update catalog 7](#_Toc247105342)

[8.2.2 Bulk load Spreadsheet 8](#_Toc247105344)

[8.2.3 Verification 8](#_Toc247105345)

[9. Debugging 9](#_Toc247105346)

[9.1 Rule does not get called 9](#_Toc247105347)

[9.2 Get Property in the Rule does not get called 9](#_Toc247105348)

[9.3 Attribute List Lookup does not find the attribute 9](#_Toc247105349)

[9.4 Target value does not get updated 10](#_Toc247105350)

[10. Conclusion 10](#_Toc247105351)

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev** | **Status** | **Author** | **Date** | **Description** |
| 0.1 | Draft | Rpattabi | 18-Nov-2009 | Initial release |
|  |  |  |  |  |

# Introduction

The users can write a ‘rule’ in order to override the default attribute copy behavior of Copy by Family of the attributes of their interest.

This is the document intended to help the rule writers to write Copy by Family Attribute Copy Rule.

# References

|  |  |
| --- | --- |
| Item | Description |
| [TR-CP-162329](http://jts.ingrppo.com/report.asp?ProductGroup=1&JTSNo=162329+&view=executive&voEdit=ON&voChildren=ON) - Bevel Information is wrong after Copy Symmetry | JTS item which initiated the requirement to have the possibility of customizing attribute copy behavior in Copy by Family |
| [Catalysts Home](http://share.intergraph.com/ppm/spknowledge/tab2/scrum2/CA1/Wiki%20Pages/Home.aspx) | Home page of the team who originally enabled attributes copy customization for Copy by Family. |

# Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| *Attribute* | Property on an entity |
| *Attribute Copy* | While copying or modifying the entities using Copy by Family, attributes are also dealt with. This process is usually termed as Attribute Copy. |
| *Parameter* | The attributes listed in the ***Parameter Tab*** of property page. Usually entities like Physical Connection, Assembly Connection, Collar, Features, etc would have parameters. |
| *Attribute Interface* | Related attributes are grouped together. The group is called attribute interface. |
| *Attribute Name* | It is name of the attribute. This name could be slightly different from the displayed name in property page. |

# Default Attribute Copy Behavior

After copying a source entity to target, Copy By Family functionality takes the attribute values of source entity and copies them to target entity. This is the same behavior in modify command.

For example, if source entity is having two attributes A and B. As shown in the figure below, after the copy operation, value of attribute A of the source entity is copied to corresponding attribute of the target entity. Same is the case with attribute B.

Target entity

**Attribute**

***Name:*** A

***Value:*** 10

**Attribute**

***Name:*** B

***Value:*** 20

**Attribute**

***Name:*** A

***Value:*** 10

**Attribute**

***Name:*** B

***Value:*** 20

Source entity

As shown above, Copy by Family functionality figures out the attributes automatically and takes care of copying the values to corresponding attributes of target entity. This default behavior could now be customized as explained in subsequent sections below.

# Attribute Copy Rule

To provide the ability to customize the default attribute copy logic, the concept of Attribute Copy Rule is introduced. This Attribute Copy Rule is a rule or some programming logic that is triggered during the copy attribute process. This rule tells Copy by Family what is the desired target attribute value.

There can be one Attribute Copy Rule per Business Object type. That is, the user could write a rule for Physical Connection attributes and another rule for Assembly Connection attributes etc.

Attribute Copy Rule could be written in Microsoft Visual Basic 6.0 programming language. This is in the same lines as writing other rules in the product. These other rules include Name Rule, SmartOccurrence Rule etc.

## What is in it?

Attribute Copy Rule provides the following to Copy by Family:

1. ***Scope*** - The name of attributes for which rule will decide the desired value for the target entity.
2. ***Desired Attribute Values*** – Rule will host specific *properties* (in other words, logic) for each of the attributes mentioned as a part of Scope. The logic in these *properties* will decide the desired attribute value for the target entity.

# Structure

Following sections detail the important elements of the Attribute Copy Rule.

## Example

Sample rule is provided along with this document. This rule is being provided just for reference. The actual rule that will be written needs to be uploaded into Catalog. The related steps are mentioned in more detail in subsequent sections.

The details of the sample rule are explained below. This sample rule could serve as a blue print for creating new Attribute Copy Rules.

## Write a Attribute Copy Rule for each Business Object

For each Business Object of interest, a corresponding Attribute Copy Rule should be written. This rule will specifically deal with attributes of that specific Business Object.

## Attributes for which the rule decides the target value

The rule has to provide attribute names for which it decides the target value. This is be done by implementing **IJCbFAttributeCopyRule** interface. Here is a example of the rule written for Physical Connection and it shows how rule specifies scope of the customization:

MicroSoft Visual Basic 6.0 Code

Implements IJCbFAttributeCopyRule

Private Sub IJCbFAttributeCopyRule\_GetCustomizedAttributeInfo( \_

ByVal customizedAttributesInfo As RuleManagement.customizedAttributesInfo)

With customizedAttributesInfo

.Add "RefSideFirstBevelAngle"

.Add "AntiRefSideFirstBevelAngle"

End With

End Sub

**This is the only method on the interface.**

**This is how the name of the attributes is provided. Above mentioned attributes are treated as customized.**

**This is the interface to be implemented by Attribute Copy Rule.**

## Rule logic for an attribute

The following is the convention to be used to implement the logic for each attribute:

1. Create a ‘Public Property Get’ method.
2. The name of the method must be same as the attribute name.
3. The return value must be an object of type **AttributeRuleResult**.

Following sample code shows how attribute methods are written.

Public Property Get RefSideFirstBevelAngle( \_

ByVal interfaceName As String, \_

ByVal sourceAttrList As IJAttributeLookup, \_

ByVal targetAttrList As IJAttributeLookup) As AttributeRuleResult

Set RefSideFirstBevelAngle = New AttributeRuleResult

If AreReferenceSidesDifferent(interfaceName, sourceAttrList, targetAttrList) Then

Dim sourceAntiRef As IJDAttribute

Set sourceAntiRef = sourceAttrList.Lookup(interfaceName, "AntiRefSideFirstBevelAngle")

If Not sourceAntiRef Is Nothing Then

RefSideFirstBevelAngle.UpdateTargetValue sourceAntiRef.Value

End If

End If

End Property

Public Property Get AntiRefSideFirstBevelAngle( \_

ByVal interfaceName As String, \_

ByVal sourceAttrList As IJAttributeLookup, \_

ByVal targetAttrList As IJAttributeLookup) As AttributeRuleResult

Set AntiRefSideFirstBevelAngle = New AttributeRuleResult

If AreReferenceSidesDifferent(interfaceName, sourceAttrList, targetAttrList) Then

Dim sourceRef As IJDAttribute

Set sourceRef = sourceAttrList.Lookup(interfaceName, "RefSideFirstBevelAngle")

If Not sourceRef Is Nothing Then

AntiRefSideFirstBevelAngle.UpdateTargetValue sourceRef.Value

End If

End If

End Property

Private Function AreReferenceSidesDifferent( \_

ByVal interfaceName As String, \_

ByVal sourceAttrList As IJAttributeLookup, \_

ByVal targetAttrList As IJAttributeLookup) As Boolean

Dim sourceRefSide As IJDAttribute

Set sourceRefSide = sourceAttrList.Lookup(interfaceName, "ReferenceSide")

Dim targetRefSide As IJDAttribute

Set targetRefSide = targetAttrList.Lookup(interfaceName, "ReferenceSide")

If (Not sourceRefSide Is Nothing) And (Not targetRefSide Is Nothing) Then

AreReferenceSidesDifferent = (sourceRefSide.Value <> targetRefSide.Value)

End If

End Function

MicroSoft Visual Basic 6.0 Code

**These are the standard parameters the getter will have.**

**Property Getter with method name same as the attribute name**

**Return value of the getter.**

**After the logic, this is how the rule decides to provide an alternative value for attribute at target.**

**If the reference sides are different, this method decides the target value for *AntiRefSideFirstBevelAngle* should be same as *RefSideFirstBevelAngle* of the source entity.**

**This how you can get a particular attribute.**

### Property Get or Getter

For each of the target attributes for which copying is customized, there needs to be a public get property. The name of it will be same as the attribute name.

### Method Parameters

This section explains parameters the above getter will receive from Copy by Family during attribute copy.

#### InterfaceName

Related attributes are grouped. The group is called attribute interface. This parameter will provide the name of that interface for the current attribute. This interface name would be required if a specific attributes need to be retrieved from source or target entity. Otherwise, this could be ignored.

#### SourceAttributeList Lookup

This is the object using which the rule can retrieve source attributes. It has **Lookup** method. If you pass interface name and required attribute name, the source attribute will be returned.

#### TargetAttributeList Lookup

This is similar to the above one except that it will return the target attribute.

### AttributeRuleResult

The getter has to return the object of type **AttributeRuleResult**. If the getter decides to override the target attribute value, **UpdateTargetValue** method should be called on this object with the desired target value. This could be ignored if the rule decides the default Copy by Family attribute copy behavior is good enough.

# Catalog

The Attribute Copy Rule for the Business Object has to be listed in the Catalog. Copy by Family will read the Catalog and trigger the Attribute Copy Rules if present accordingly. If there are no rules, the default Copy By Family behavior will take over.

The following sections provide details about how to update Catalog with Attribute Rule information.

## When to add a new Attribute Copy Rule?

Please note that there is only one rule allowed per Business Object type.

For example, if a few attributes of Assembly Connection Business Object needs to be customized, a new rule needs to be written for that Business Object.

If there is an already existing Attribute Copy Rule for Assembly Connection Business Object, rule needs to be updated if more customization is required for other attributes of Assembly Connection Business Object.

## How to add the new rule?

You have to create a new ActiveX DLL Project using Microsoft Visual Basic 6.0 environment and add a class to customize the attribute copy behavior for the specific Business Object. If desired, all the Attribute Copy Rule classes could be hosted in the same ActiveX DLL project.

### Information required to update catalog

The following is the information required to update the Catalog:

1. The **ProgId** of the new Attribute Copy Rule
2. Rule name of the Business Object

#### Rule Name

Business Object class name will be treated as the rule name.

Class names are already familiar to rule writers. For example, for Physical Connection the class name is **CPStructPhysicalConnection**. So it will be treated as rule name.

A quick look at the **Metadata Browser** will provide the class name of a given Business Object. Alternatively, Intergraph could provide class name of the business objects if required.

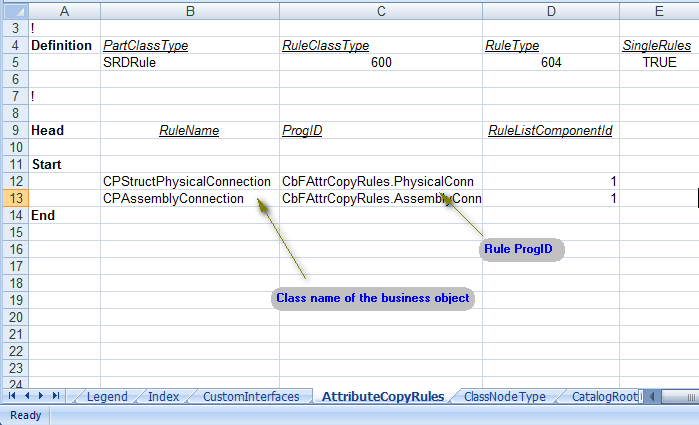
### Bulk load Spreadsheet

The Excel workbook that is used for bulk loading the Attribute Copy Rules information is provided with the product installation and could be located at:

<Product Installation Directory>\ShipCatalogData\BulkLoad\DataFiles\CopyByFamilyRules.xls

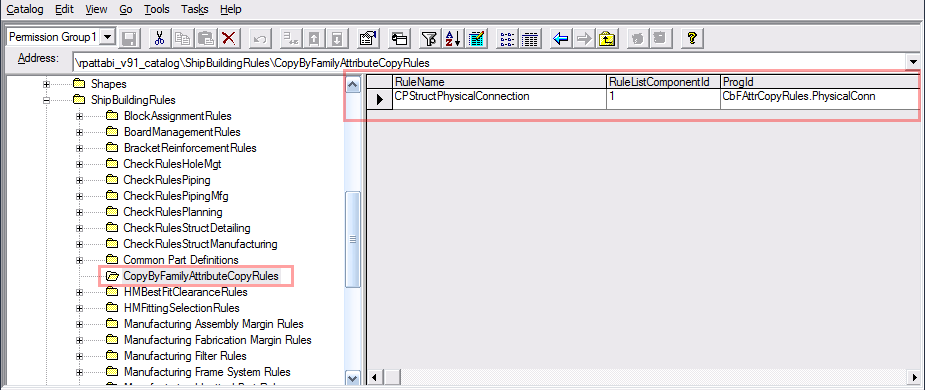
If a new Attribute Copy Rule needs to be added or modified, sheet AttributeCopyRules in Excel workbook is updated as shown in the below figure.

The usual bulk loading procedure is to be followed to update the catalog.



### Verification

After bulkloading the data into Catalog, Attribute Copy Rules listed in the Catalog. This could be viewed using Catalog Task as shown below.



# Debugging

This section provides the possible problems rule writer might face while developing Attribute Copy Rule. Also, a few ways are provided to understand and resolve these problems.

## Rule does not get called

Copy by Family may not trigger the rule due to the following possible reasons:

1. The Attribute Copy Rule project is either not built properly or registered.
2. The Attribute Copy Rule ProgId has a *typo* in the Bulkload Excel workbook.
3. The Attribute Copy Rule name mentioned in the Catalog is incorrect. Rule name must ne the valid class name of Business Object. Please refer to section for more details.

Rule writer could ensure whether the rule is getting triggered or not by putting a break point or a message box in **IJCbFAttributeCopyRule** implementation.

## Get Property in the Rule does not get called

A particular property implemented in the rule may not be triggered due to the following possible reasons:

1. The Attribute Copy Rule may not be recognized by Copy by Family. Please ensure that the rule is getting triggered. Please see section for further details.
2. The attribute name is missed to be mentioned in **GetCustomizedAttributeInfo** implementation. It could be there is a typo in the name of the attribute.
3. Get Property name does not match with the attribute name. It may have a typo or capitalization differences.
4. The parameters of the Get Property are not correct. Please review section

## Attribute List Lookup does not find the attribute

The attribute list lookup may not get the desired attribute due to the following possible reasons:

1. The input parameters are incorrect. One or more of the following could go wrong:
   1. The attribute name has a typo.
   2. The attribute does not exist for the Business Object. For example, **Thickness** is not a valid attribute Profile System.
   3. The interface name is incorrect for an attribute. Please note that the interface name passed as input to Get Property may be suitable for most cases. But it may be possible that for a specific attribute you are looking for is under a different interface. In that case, please specify the correct interface name. If you don’t know the correct interface name, consult Metadata Browser. If all fails, please Intergraph Support.
2. The attribute list given does not have the attribute you are interested in. As of this writing, Copy by Family will not provide you with all the attributes of source or target entity. If you are interested in a specific attribute, and you don’t get it after ensuring the parameters of the Lookup is correct, please provide your requirement to Intergraph Support.

## Target value does not get updated

If you have ensured that there are no problems with the rule as per the above sections, and still the attribute value you provide does not get updated at the target entity, please try the following:

1. Please review and ensure that you have instantiated and updated the return type correctly.
2. Ensure, for example through debugger, that **UpdateTargetValue** call is indeed made.
3. Ensure that you have provided desired value while calling **UpdateTargetValue**.
4. Please Intergraph Support in case you still face the issue after trying out the instructions here.

# Conclusion

Attribute Copy Rule provides the users of Copy by Family to customize the behavior of attribute copy during copy similar and copy symmetry in a generic manner.

The architecture and workflow is designed to make it more intuitive and easy to write Attribute Copy Rules.