**IRule 3.2 Installation and basic operation**

Contents

[What is IRule 2](#_Toc495478698)

[System requirements 2](#_Toc495478699)

[Installation 2](#_Toc495478700)

[Basic Operation 3](#_Toc495478701)

[IRule Object Attributes 3](#_Toc495478702)

[Object Rule Attributes 4](#_Toc495478703)

[Allocation Rule Attributes 4](#_Toc495478704)

## What is IRule

The IRule CostPerform add-on allows for the management of model allocations on the “Meta model level”. IRule extends the capabilities of conditional driver to include not only the destination but also the source object definition.

IRule benefits:

* Streamline model design
* Consistent model behavior
* Reduced construction and maintenance times
* Single logic for multiple models
* Management of allocations versions
* Additional allocation definition capabilities
* In model documentation

For more information go to [www.in-sight-analytics.com](http://www.in-sight-analytics.com) and contact us using the knowledge center or contact form.

## System requirements

CostPerform 9.2 and above

## Installation

1. Unzip the IRule.zip file to [CostPerform Installation folder]\IRule\
2. You can install IRule in a different folder, this will require you to edit the following files:
   * IRule Execute Compiled – set the path variable to your installation path
   * IRuleDestribute Execute Compiled – set the path variable to your installation path
3. Import the attribute tabs – optional
   * IRule - Applied Rule Attribute Tab.xml – used to identify and control the applied rule on an object
   * IRule Attribute Tab.xml – used to configure a rule
4. Add custom actions, set both to not show script window and use costmodel context
   * C:\Program Files (x86)\CostPerform\IRule\IRule Execute Compiled.ccs – for use of IRule on a single model\period
   * C:\Program Files (x86)\CostPerform\IRule\IRuleDestribute Execute Compiled.ccs – for use of IRule with multiple model\periods

## Basic Operation

1. Upgrading a metamodel to use IRule
   1. From the installation folder run the IRule Metadata Upgrade.ccs script file. You will be asked which metamodels would you like to upgrade, you can choose either a single metamodel or multiple metamodels to upgrade. The script will add the Rules layer and required attributes both to the Rules layer, common attributes and allocation attributes.
   2. It is recommended to place the Rules layer as the first layer of the model as a best practice
2. Creating you first rule
   1. To create a new rule simply add an object to the Rules layer in your model
   2. To activate or deactivate the rule use the object allocation state, to deactivate a rule simply set the state to false
   3. Now fill in the IRule object attributes, a full list of attributes, their functions and use follows

### IRule Object Attributes

|  |  |  |
| --- | --- | --- |
| Attribute | Usage | Syntax |
| Symbol | Unique identifier |  |
| Name | Name of the Rule |  |
| Description | Description of the Rule, will be applied to every source object |  |
| Rule Diagnostics Summary | In case of errors in Rule application or syntax, delivers detailed explanation of error or errors | Result variable, no input required |
| Rule Collision Status | In case of the Rule being applied on an object alongside with one or more additional rules, an error will be displayed | Result variable, no input required |
| Rule Source Object condition | Definition of source objects to apply the Rule upon | Filter syntax |
| Rule Source Object Allocates | Sets the object allocation state of the source object | Value list |
| Calculation Rule | Sets the calculation rule of the source object | Value list |
| Rule Rate Attribute symbol | Sets the rate attribute symbol of the source object when using calculation rule 3 - rate | Attribute symbol |
| Delayed allocation order | Sets the delayed calculation order of the source object | Value list |
| Moment of volume calculation | Sets the volume of volume calculation of the source object | Value list |
| Rule capacity Attribute symbol | Allows to use an attribute to set the capacity attribute of the source object, useful for calculated capacity | Attribute symbol |
| Rule Destination Object Condition | The definition of destination objects of each Rule, can be either a static definition or a dynamic one. The destination object is not limited to allocating objects | Static definition: Filter syntax  Dynamic definition: Conditional driver value syntax |
| Rule Destination Allocation Value | Sets the value 1 attribute of the allocation. The default value is 1 | Value or Conditional driver value syntax |
| Driver | Sets the driver attribute for the source object |  |
| Object Distributes | Sets the object distributes attribute of the source object | Value list |
| Conditional driver | Sets the conditional driver attribute for the source object | Value list |
| Conditional driver value | Sets the conditional driver value attribute for the source object | Conditional driver value syntax |
| Rule Residue Destination Object Condition | The definition of destination residue objects of each Rule, can be either a static definition or a dynamic one | Static definition: Filter syntax  Dynamic definition: Conditional driver value syntax |
| Rule Residue Default Value | Sets the value 1 attribute of the residue allocation. The default value is 1 | Value or Conditional driver value syntax |

### Object Rule Attributes

|  |  |
| --- | --- |
| Attribute | Usage |
| Applied Rule | The symbol and name of the Rule applied on the object |
| Applied Rule Description | The description of the rule applied on the object |
| Rule Override | Allows to set the rule to ignore this object as a source object |

### Allocation Rule Attributes

|  |  |
| --- | --- |
| Attribute | Usage |
| Allocation Created by Rule | Designates the symbol and name of the rule that created the allocation |
| IRule Allocation Override | Allows to set the rule to ignore this allocation when applying a rule on the source object |