$project.name

$DocumentTitle

Author: $Author

Revision: $Revisions.lastChild.name

|  |  |
| --- | --- |
| $CompanyName |  |
| $CompanyAddress | $date.get(“MMMMM dd, yyyy”) |

**Approval**

|  |
| --- |
| The original of this document is approved and signed by:  Name:  Surname:  Title:  Date:  Signature: |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Description** | **Author** |
| #forrow($rev in $Revisions)  $rev.name | $rev.date | $rev.description | $rev.author #endrow |
|  |  |  |  |

| Table of Contents |
| --- |
| Microsoft Word users please click here and press F9 to create Table of Contents.  OpenOffice.org users please remove this text and select Insert Table of Content from menu. |

| Table of Figures |
| --- |
| Microsoft Word users please click here and press F9 to create Table of Contents.  OpenOffice.org users please remove this text and select Insert Table of Content from menu. |

## #import('hrsort', 'com.hiltonroscoe.mdreportext.Sorter')

## #macro (classdoc $class)

## Class $bookmark.create($class.ID, $class.name)

$class.documentation

#if ($class.hasImage())

$class.image

1. $class.name

#end

#if ($class.hasOwnedAttribute())

#set ($attrArray = $array.createArray($class.ownedAttribute))

| **Property** | **Multiplicity** | **Type** | **Property Description** |
| --- | --- | --- | --- |
| #forrow($attr in $hrsort.sortByRule($attrArray))#if($attr.association){#end#if (!$report.isEmpty($attr.name))$bookmark.create($attr.ID,$attr.name)#else$bookmark.create($attr.ID,$attr.type.name)#end#if($attr.association)}#end | $attr.multiplicity | #if($elements.contains($attr.type))$bookmark.open($attr.type.ID, $attr.type.name)#else$attr.type.name#end | $attr.documentation#endrow |

#foreach($el in $class.ownedElement)

#if ($report.containsStereotype($el,"psmDocumentation"))

#if(!$report.isEmpty($report.getStereotypePropertyString($el, "psmDocumentation", "documentation\_purpose")))

$report.getStereotypePropertyString($el, "psmDocumentation", "documentation\_purpose").concat(“:”)

#end

$el.documentation$el.body

#end

#end

##START OCL

#set ($constraintList = $array.createArray())  
#foreach ($constraint in $Constraint)  
#if ($constraint.constrainedElement.contains($class))  
#set ($void = $constraintList.add($constraint))

#end

#end  
#end

#if(!$list.isEmpty($constraintList))

### Business Rules

#foreach($constraint in $constraintList)

#### $constraint.name

$report.getStereotypePropertyString($constraint, "validationRule", "errorMessage"):

## This api is not published and may break

#foreach($bodyPart in $constraint.specification.getBody())

$bodyPart

#end

#end

##ENDOCL

#end

#if ($class.hasOwnedOperation())

### Operation Detail

#foreach ($ope in $sorter.sort($class.ownedOperation))

#### $ope.name

$ope.documentation

|  |  |
| --- | --- |
| **Type** | $ope.type.name |
| **Visibility** | $ope.visibility |
| **Is Abstract** | $ope.isAbstract |
| **Parameter** | #set ($paraList = $array.createArray())  #foreach($para in $sorter.sort($ope.ownedParameter))  #if($para.direction!=”return”)  #set ($void = $paraList.add($para))  #end  #end  #foreach($para in $sorter.sort($paraList))  #set($paraName = “$para.direction $para.name : $para.type.name”)  #if($para.multiplicity!=””)  #set($paraName = “$paraName [$para.multiplicity]”)  #end  #if($para.defaultValue && $para.defaultValue.hasText())  #set($paraName = $para.defaultValue.text)  #end   * $paraName   #end |

#end

#end

#if ($GroupSubtypes == "true")

## recurse over subtypes

#set($relationList = $report.getRelationship($class))

$group.init()

#foreach ($relation in $relationList)

#set($void = $group.put($relation.humanType, $relation))

#end

#if (!$relationList.isEmpty())

#if ($group.contains("Generalization"))

#set ($relatedList = $array.createArray())

#foreach($rel in $sorter.sort($group.get("Generalization")))

#foreach ($re in $sorter.sort($rel.relatedElement, "name"))

#set ($void = $relatedList.add($re))

#end

#end

#foreach ($relClass in $sorter.sort($relatedList, "name"))

#if($relClass != $class)

#classdoc($relClass)

#end

#end

#end

#end

#end

#end

# Introduction

## Purpose

$Purpose

## Scope

$Scope

## Overview

$Overview

#macro (removeLineBreak $s)

#set ($displayedName = “”)

#foreach ($c in $s.toCharArray())

#if($c.getType($c)==15)

#set($displayedName = $displayedName.concat(“ “))

#else

#set($displayedName = $displayedName.concat($c.toString()))

#end

#end

#end

#set($packages = $array.createArray())

#set($void = $array.addCollection($packages, $Package))

#set($void = $array.addCollection($packages, $SmartPackage))

#foreach ($package in $sorter.sort($packages, “qualifiedName”))

#set($qualifiedName = $package.qualifiedName)

# Package $qualifiedName

$package.documentation

#if ($package.hasImage())

$package.image

1. $package.name

#end

|  |  |
| --- | --- |
| **Name** | $package.name |
| **Qualified Name** | $qualifiedName |

#set ($classList = $array.createArray())

#set ($interfaceList = $array.createArray())

#set ($enumerationList = $array.createArray())

#foreach ($element in $report.getOwnedElementsIncludingAdditional($package, true))

#if ($element.elementType == “class”)

#set ($void = $classList.add($element))

#elseif ($element.elementType == “interface”)

#set ($void = $interfaceList.add($element))

#elseif ($element.elementType == “enumeration”)

#set ($void = $enumerationList.add($element))

#end

#end

#foreach ($class in $sorter.sort($classList))

#if ($GroupSubtypes == "false" || $list.size($class.superClass) == 0)  
#classdoc ($class)

#end

#end

#foreach ($interface in $sorter.sort($interfaceList))

## Interface $bookmark.create($interface.ID, $interface.name)

$interface.documentation

#if ($interface.hasImage())

$interface.image

1. $interface.name

#end

#if ($interface.hasOwnedAttribute())

### Attribute Detail

#foreach ($attr in $sorter.sort($interface.ownedAttribute))

#### $attr.name

$attr.documentation

|  |  |
| --- | --- |
| **Type** | $attr.type.name |
| **Default Value** | #if($attr.defaultValue && $attr.defaultValue.hasText())  $attr.defaultValue.text  #end |
| **Visibility** | $attr.visibility |
| **Multiplicity** | $attr.multiplicity |

#end

#end

#if ($interface.hasOwnedOperation())

### Operation Detail

#foreach ($ope in $sorter.sort($interface.ownedOperation))

#### $ope.name

**Description**

$ope.documentation

|  |  |
| --- | --- |
| **Type** | $ope.type.name |
| **Visibility** | $ope.visibility |
| **Is Abstract** | $ope.isAbstract |
| **Parameter** | #set ($paraList = $array.createArray())  #foreach($para in $sorter.sort($ope.ownedParameter))  #if($para.direction!=”return”)  #set ($void = $paraList.add($para))  #end  #end  #foreach($para in $sorter.sort($paraList))  #set($paraName = “$para.direction $para.name : $para.type.name”)  #if($para.multiplicity!=””)  #set($paraName = “$paraName [$para.multiplicity]”)  #end  #if($para.defaultValue && $para.defaultValue.hasText())  #set($paraName = $paraName.concat(“=”).concat($para.defaultValue.text))  #end   * $paraName   #end |

#end

#end

#set($relationList = $report.getRelationship($interface))

$group.init()

#foreach ($relation in $relationList)$group.put($relation.humanType, $relation)#end

#if (!$relationList.isEmpty())

### Relation Detail

#foreach ($key in $sorter.sort($group.groupNames()))

#### $key

#foreach($rel in $sorter.sort($group.get($key)))

|  |  |
| --- | --- |
| **Name** | #if($rel.hasName())  $rel.name#end |
| **Related Element** | #foreach ($re in $sorter.sort($rel.relatedElement))  #if($re != $interface)   * $re.name   #end  #end |

#end

#end

#end

#end

#foreach ($enum in $sorter.sort($enumerationList))

## Enumeration $bookmark.create($enum.ID, $enum.name)

$enum.documentation

#if ($enum.hasImage())

$enum.image

1. $enum.name

#end

| **Value** | **Value Description** |
| --- | --- |
| #forrow ($enl in $enum.OwnedLiteral)$enl.name | $enl.documentation#endrow |

#set($relationList = $report.getRelationship($enum))

$group.init()

#foreach ($relation in $relationList)$group.put($relation.humanType, $relation)#end

#if (!$relationList.isEmpty())

### Relation Detail

#foreach ($key in $sorter.sort($group.groupNames()))

#### $key

#foreach($rel in $sorter.sort($group.get($key)))

|  |  |
| --- | --- |
| **Name** | #if($rel.hasName())  $rel.name#end |
| **Related Element** | #foreach ($re in $sorter.sort($rel.relatedElement))  #if($re != $enum)   * $re.name   #end  #end |

#end

#end

#end

#end

#end

# Appendix A: Diagram

#foreach ($diagram in $sorter.sort($Diagram))

#if ($diagram.diagramType == “Class Diagram”)

#removeLineBreak($diagram.name)

## $displayedName

$diagram.documentation

$diagram.image

1. $diagram.name

#end

#end