

<config>

<output path=’${generic\_generation\_output}' />

</config>

<context model='${generic\_generation\_model}' element=’{0}’ importedBundles=‘gmf;SysML14Queries;papyrus’ searchMetamodels=‘true’/><drop/>

This is a comment, if you want to modify this template please care to change the parameters ${generic\_generation\_output}’ and ${generic\_generation\_model} with your custom values. Note that, you will not be able to use the wizard in that case. Process by clicking on the template and choose generation documentation.<drop/>

This document has been generated using **Eclipse Gendoc** : More information available here :

<https://www.eclipse.org/gendoc/>

The model was produced using Papyrus and SysML 1.4 module :

<https://www.eclipse.org/papyrus/>

<https://marketplace.eclipse.org/content/papyrus-sysml-14>

|  |  |  |  |
| --- | --- | --- | --- |
|  | Author(s) | Checked by | Approbation |
| Name |  |  |  |
| Compagny |  |  |  |
| Department |  |  |  |
| Date |  |  |  |
| Visa |  |  |  |
| summary |  |  |  |

|  |
| --- |
| **Disclaimer** : Contractors participating to this report shall incur no liability whatsoever for any damage or loss which may result from the use or exploitation of information and/or Rights contained in this report. |

Table des matières

[[self.name/] 6](#_Toc365363286)

[[for (s : String | c.getAppliedStereotypes().name)][ clean(s)/] [/for] : [c.name/] 8](#_Toc365363287)

[[c1.name/] 9](#_Toc365363288)

[Package : [p.name/] 11](#_Toc365363289)

[[for (s : String | c.getAppliedStereotypes().name)][ clean(s)/] [/for] : [c.name/] 12](#_Toc365363290)

[Package : [p\_.name/] 16](#_Toc365363291)

[Figure 1 *[clean(d.name)/]* 5](#_Toc365372508)

[Figure 2 *[clean(d1.name)/]* 6](#_Toc365372509)

[Figure 3 *[clean(d.name)/]* 8](#_Toc365372510)

[Figure 4 *[clean(d1.name)/]* 9](#_Toc365372511)

[Figure 5 *[clean(d.name)/]* 11](#_Toc365372512)

[Figure 6 *[clean(d1\_.name)/]* 12](#_Toc365372513)

<gendoc>

[if (self.getPapyrusDiagrams()->size()>0)]<drop/>

# [self.name/]

[for (s : String | self.ownedComment.\_body)][clean(s)/] [/for]

[for (d : Diagram | self.getPapyrusDiagrams())]<drop/>

[if (isDiagramEmpty(d)=false)]<drop/>

<image object=‘[d.getDiagram()/]’ maxW=‘true’ keepH=‘false’><drop/>

</image>

Figure 1 *[clean(d.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

[*for* (elt : Element | self.ownedElement)]<drop/>

[let c : Class = elt] <drop/>

<drop/>

## [for (s : String | c.getAppliedStereotypes().name)][ clean(s)/] [/for] : [c.name/]

[if (c.getPapyrusDiagrams()->size()>0)]<drop/>

[for (d1 : Diagram | c.getPapyrusDiagrams())]<drop/>

[if (isDiagramEmpty(d1)=false)]<drop/>

<image object=‘[d1.getDiagram()/]’ maxW=‘true’ keepH=‘false’><drop/>

</image>

Figure 2 *[clean(d1.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

**Comment :** [for (s1 : String | splitNewLine(displayComment(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Ports :** [for (s1 : String | splitNewLine(displayPort(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Parts :** [for (s1 : String | splitNewLine(displayPart(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Attributes :** [for (s1 : String | splitNewLine(displayAttribute(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Operations :** [for (s1 : String | splitNewLine(displayOperation(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Receptions :** [for (\_ne : NamedElement | c.ownedReception)] <dropEmpty>[\_ne.name/]</dropEmpty>

[/for]<drop/>

[for (e : NamedElement | c.ownedElement)] <drop/>

[let c1 : Class = e] <drop/>

### [c1.name/]

<drop/>

* **Comment :** [for (s1 : String | splitNewLine(displayComment(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Ports :** [for (s1 : String | splitNewLine(displayPort(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Parts :** [for (s1 : String | splitNewLine(displayPart(c1)))] <dropEmpty>[s1/]</dropEmpty>

[/for]<drop/>

* **Attributes :** [for (s1 : String | splitNewLine(displayAttribute(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Operations :** [for (s1 : String | splitNewLine(displayOperation(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Receptions :** [for (\_ne : NamedElement | c1.ownedReception)] <dropEmpty>[\_ne.name/]</dropEmpty>

[/for]<drop/>

[/let]<drop/>

[/for] <drop/>

[/let] <drop/>

[/for]

[*for* (elt : Element | self.ownedElement)]<drop/>

[*let* p : Package = elt]<drop/>

<drop/>

## Package : [p.name/]

[if (p.getPapyrusDiagrams()->size()>0)]<drop/>

[for (d : Diagram | p.getPapyrusDiagrams())]<drop/>

[if (isDiagramEmpty(d)=false)]<drop/>

<image object=‘[d.getDiagram()/]’ maxW=‘true’ keepH=‘false’><drop/>

</image>

Figure 3 *[clean(d.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

[*for* (\_elt : Element | p.ownedElement)]<drop/>

[let c : Class = \_elt] <drop/>

### [for (s : String | c.getAppliedStereotypes().name)][ clean(s)/] [/for] : [c.name/]

[if (c.getPapyrusDiagrams()->size()>0)]<drop/>

[for (d1 : Diagram | c.getPapyrusDiagrams())]<drop/>

[if (isDiagramEmpty(d1)=false)]<drop/>

<image object=‘[d1.getDiagram()/]’ maxW=‘true’ keepH=‘false’><drop/>

</image>

Figure 4 *[clean(d1.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

**Comment :** [for (s1 : String | splitNewLine(displayComment(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Ports :** [for (s1 : String | splitNewLine(displayPort(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Parts :** [for (s1 : String | splitNewLine(displayPart(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Attributes :** [for (s1 : String | splitNewLine(displayAttribute(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Operations :** [for (s1 : String | splitNewLine(displayOperation(c)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Receptions :** [for (\_ne : NamedElement | c.ownedReception)] <dropEmpty>[\_ne.name/]</dropEmpty>

[/for]<drop/>

[for (e : Element | c.ownedElement)] <drop/>

[let c1 : Class = e] <drop/>

#### [c1.name/]

<drop/>

* **Comment :** [for (s1 : String | splitNewLine(displayComment(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Ports :** [for (s1 : String | splitNewLine(displayPort(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Parts :** [for (s1 : String | splitNewLine(displayPart(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Attributes :** [for (s1 : String | splitNewLine(displayAttribute(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Operations :** [for (s1 : String | splitNewLine(displayOperation(c1)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Receptions :** [for (\_ne : NamedElement | c1.ownedReception)] <dropEmpty>[\_ne.name/]</dropEmpty>

[/for]<drop/>

[/let]<drop/>

[/for] <drop/>

[/let] <drop/>

[/for]

[*for* (\_elt : Element | p.ownedElement)]<drop/>

[let p\_ : Package = \_elt]<drop/>

### Package : [p\_.name/]

[if (p\_.getPapyrusDiagrams()->size()>0)]<drop/>

[for (d : Diagram | p\_.getPapyrusDiagrams())]<drop/>

[if (isDiagramEmpty(d)=false)]<drop/>

<image object=‘[d.getDiagram()/]’ maxW=‘true’ keepH=‘false’><drop/>

</image>

Figure 5 *[clean(d.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

[*for* (elt\_ : Element | p\_.ownedElement)]<drop/>

[let c\_ : Class = elt\_] <drop/>

<drop/>

#### [for (s\_ : String | c\_.getAppliedStereotypes().name)][ clean(s\_)/] [/for] : [c\_.name/]

[if (c\_.getPapyrusDiagrams()->size()>0)]<drop/>

[for (d1\_ : Diagram | c\_.getPapyrusDiagrams())]<drop/>

[if (isDiagramEmpty(d1\_)=false)]<drop/>

<image object=‘[d1\_.getDiagram()/]’ maxW=‘true’ keepH=‘false’><drop/>

</image>

Figure 6 *[clean(d1\_.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

**Comment :** [for (s1 : String | splitNewLine(displayComment(c\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Ports :** [for (s1 : String | splitNewLine(displayPort(c\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Parts :** [for (s1 : String | splitNewLine(displayPart(c\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Attributes :** [for (s1 : String | splitNewLine(displayAttribute(c\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Operations :** [for (s1 : String | splitNewLine(displayOperation(c\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

**Receptions :** [for (\_ne : NamedElement | c\_.ownedReception)] <dropEmpty>[\_ne.name/]</dropEmpty>

[/for]<drop/>

[for (e\_ : Element | c\_.ownedElement)] <drop/>

[let c1\_ : Class = e\_] <drop/>

##### [c1\_.name/]

<drop/>

* **Comment :** [for (s1 : String | splitNewLine(displayComment(c1\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Ports :** [for (s1 : String | splitNewLine(displayPort(c1\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Parts :** [for (s1 : String | splitNewLine(displayPart(c1\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Attributes :** [for (s1 : String | splitNewLine(displayAttribute(c1\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Operations :** [for (s1 : String | splitNewLine(displayOperation(c1\_)))] <dropEmpty>[clean(s1)/]</dropEmpty>

[/for]<drop/>

* **Receptions :** [for (\_ne : NamedElement | c1\_.ownedReception)] <dropEmpty>[\_ne.name/]</dropEmpty>

[/for]<drop/>

[/let]<drop/>

[/for] <drop/>

[/let] <drop/>

[/for]<drop/>

[/let]<drop/>

[/for] <drop/>

[/let]<drop/>

[/for]<drop/>

</gendoc>