

<config>

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

</config>

<context model='${generic\_generation\_model}' element=’{0}’ importedBundles=‘gmf;RobotML\_Queries;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 generate documentation.<drop/>

This document has been generated using **gendoc2** : More information available here :

<http://marketplace.eclipse.org/content/gendoc2>

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

|  |
| --- |
| **Attention** : la responsabilité des entreprises et des organismes ayant participé à l’élaboration de ce document ne peut en aucun cas être engagée en cas de dommages ou de pertes résultant de l’utilisation ou de l’exploitation des informations qui y sont contenues.  **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. |

<gendoc>

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

[if elt.oclIsTypeOf(Package)] <drop/>

[*let* p : Package = elt.oclAsType(Package)]<drop/>

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

# Package : [p.name/]

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

[for (d : Diagram | p.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/>

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

[if elt.oclIsTypeOf(Class)] <drop/>

[let c : Class = elt.oclAsType(Class)] <drop/>

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

Description : <dropEmpty>[for(s : String | c.ownedComment.\_body)][clean(s)/][/for]</dropEmpty>

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

<dropEmpty>[clean(displayNamedElement(\_e))/]</dropEmpty>

[/for] <drop/>

[/let] <drop/>

[/if] <drop/>

[if elt.oclIsTypeOf(DataType)] <drop/>

[let d : DataType = elt.oclAsType(DataType)] <drop/>

- [d.name/]

Description : <dropEmpty>[for(s : String | d.ownedComment.\_body)][clean(s)/][/for]</dropEmpty>

[for (\_e : NamedElement | d.ownedElement)] <drop/>

<dropEmpty>[clean(displayNamedElement(\_e))/]</dropEmpty>

[/for] <drop/>

[/let] <drop/>

[/if] <drop/>

[if elt.oclIsTypeOf(Interface)] <drop/>

[let i : Interface = elt.oclAsType(Interface)] <drop/>

- [i.name/]

Description : <dropEmpty>[for(s : String | i.ownedComment.\_body)][clean(s)/][/for]</dropEmpty>

[for (o : Operation | i.ownedOperation)]<drop/>

[clean(o.name)/]

[/for] <drop/>

[/let] <drop/>

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

[/let]<drop/>

[/if]<drop/>

[/for]<drop/>

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

# Model root :

[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 2 *[clean(d.name)/]*

[/if] <drop/>

[/for] <drop/>

[/if]<drop/>

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

[if elt.oclIsTypeOf(Class)] <drop/>

[let c : Class = elt.oclAsType(Class)] <drop/>

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

Description : <dropEmpty>[for(s : String | c.ownedComment.\_body)][clean(s)/][/for]</dropEmpty>

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

<dropEmpty>[clean(displayNamedElement(\_e))/]</dropEmpty>

[/for] <drop/>

[/let] <drop/>

[/if] <drop/>

[if elt.oclIsTypeOf(DataType)] <drop/>

[let d : DataType = elt.oclAsType(DataType)] <drop/>

- [d.name/]

Description : <dropEmpty>[for(s : String | d.ownedComment.\_body)][clean(s)/][/for]</dropEmpty>

[for (\_e : NamedElement | d.ownedElement)] <drop/>

<dropEmpty>[clean(displayNamedElement(\_e))/]</dropEmpty>

[/for] <drop/>

[/let] <drop/>

[/if] <drop/>

[if elt.oclIsTypeOf(Interface)] <drop/>

[let i : Interface = elt.oclAsType(Interface)] <drop/>

- [i.name/]

Description : <dropEmpty>[for(s : String | i.ownedComment.\_body)][clean(s)/][/for]</dropEmpty>

[for (o : Operation | i.ownedOperation)]<drop/>

[clean(o.name)/]

[/for] <drop/>

[/let] <drop/>

[/if] <drop/>

[/for]<drop/>

</gendoc>