Connection to the properties file containing model path <drop/>

<properties path=’${project\_loc}/template/conf.properties’ /> <drop/>

Ouput path is located in property file <drop/>

<config>

<output path=’${outputPath}’ />

</config>

<context model=’${model}’ element=’{0}’ importedBundles=’gmf;papyrus’/>

|  |  |  |
| --- | --- | --- |
|  | <gendoc>[self.oclAsType(uml::NamedElement).name.clean()/]</gendoc> |  |

Document generated by Gendoc (<http://www.eclipse.org/gendoc>)

Help and support at : <https://www.eclipse.org/forums/index.php/f/286/>

Copyright (c) 11/30/2016  
All rights reserved. This program and the accompanying materials  
are made available under the terms of the Eclipse Public License v1.0  
which accompanies this distribution, and is available at  
<http://www.eclipse.org/legal/epl-v10.html>

Contenu

**Aucune entrée de table des matières n'a été trouvée.**

[Figure 1 : [d.name.clean()/] 6](#_Toc468282220)

<gendoc> <drop/>

Recursive navigation into the model <drop/>

[self.oclAsType(uml::Package).chapter(1)/]

</gendoc> <drop/>

Fragments declaration <drop/>

# title : This fragment generates the correct style for headings <drop/>

<fragment name=’title’> <drop/>

<arg name=’s’ type=’String’/> <drop/>

<arg name=’e’ type=’Integer’/> <drop/>

[if (e = 1)] <drop/>

# [s/]

[elseif (e = 2)] <drop/>

## [s/]

[elseif (e = 3)] <drop/>

### [s/]

[elseif (e = 4)] <drop/>

#### [s/]

[elseif (e = 5)] <drop/>

##### [s/]

[elseif (e = 6)] <drop/>

###### [s/]

[/if] <drop/>

</fragment> <drop/>

# chapter : This fragment generates recursively chapters for each sub packages <drop/>

<fragment name=’chapter’ importedFragments=’displayComment;title;diagrams’ > <drop/>

<arg name=’e’ type=’uml::Package’/> <drop/>

<arg name=’val’ type=’Integer’/> <drop/>

Display the name of the package <drop/>

[self.name.clean().title(val)/] <drop/>

Display the diagrams of the package <drop/>

[self.diagrams (val + 1)/] <drop/>

Display the comments of the package <drop/>

[self.displayComment()/] <drop/>

[for (p : uml::Package | e.eContents()->filter(uml::Package))] <drop/>

For each sub package a new chapter <drop/>

[self.chapter(val +1)/] <drop/>

[/for] <drop/>

</fragment>

# displayComment : This fragment handles correctly the comments <drop/>

<fragment name=’displayComment’> <drop/>

<arg name=’e’ type=’uml::Element’/> <drop/>

[for (description : String | e.eContents()->filter(uml::Comment)->select(c |c.annotatedElement->includes(e)).\_body.splitNewLine())][description.clean()/]

[/for] <drop/>

</fragment> <drop/>

# diagrams : generates the diagrams of the given element <drop/>

<fragment name=’diagrams’ importedBundles=’gmf;papyrus’ importedFragments=’title;displayComment’> <drop/>

<arg name=’e’ type=’uml::Element’/> <drop/>

<arg name=’val’ type=’Integer’/> <drop/>

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

The diagram is display if it is not empty <drop/>

[if (not d.isDiagramEmpty())] <drop/>

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

Figure 1 : [d.name.clean()/]

</image> <drop/>

For each element in the diagram which is not a package the diagram is displayed <drop/>

[for (eobject : uml::Element | d.getElementsInDiagram()->filter(uml::Element)->select(not oclIsKindOf(uml::Package)))] <drop/>

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

[if (eobject.oclIsKindOf(uml::NamedElement))] <drop/>

[eobject.oclAsType(uml::NamedElement).name.clean().title(val)/] <drop/>

[/if] <drop/>

[eobject.diagrams(val + 1)/] <drop/>

[eobject.displayComment()/] <drop/>

[/if] <drop/>

[/for] <drop/>

[/if] <drop/>

[/for] <drop/>

</fragment> <drop/>