ONAP Information Model

<config services=’TagFileBuffer’>  
<output path=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\ModelStructure\OnapModelStructure.docx' />

</config>

## Common Model

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Common.notation' element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Diagrams

[for (d : notation::Diagram |notation::Diagram.allInstances()->sortedBy(name))]<drop/>

#### [d.name/]

[/for]<drop/>

</gendoc><drop/>

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Common.uml’ element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Classes

[for (cl:Class | self.eAllContents(Class)->sortedBy(name))]<drop/>

#### [cl.name/]

[/for]<drop/>

### Data Types

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

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

#### [dt.name/]

[else][/if]<drop/>

[/for]<drop/>

### Enumerations

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(Enumeration)]<drop/>

#### [dt.name/]

Contains Enumeration Literals:

[for (e:EnumerationLiteral|dt.oclAsType(Enumeration).ownedLiteral)]<drop/>

* [e.name/]:
  + [for (co:Comment | e.ownedComment)]<drop/>
  + <dropEmpty>[co.\_body.clean()/]
  + </dropEmpty>[/for]<drop/>

[/for]<drop/>

[else] [/if]<drop/>

[/for]<drop/>

### Primitives

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(PrimitiveType)]<drop/>

#### [dt.name/]

[else] [/if]<drop/>

[/for]<drop/>

</gendoc><drop/>

## Service Model

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Service.notation' element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Diagrams

[for (d : notation::Diagram |notation::Diagram.allInstances()->sortedBy(name))]<drop/>

#### [d.name/]

[/for]<drop/>

</gendoc><drop/>

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Service.uml’ element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Classes

[for (cl:Class | self.eAllContents(Class)->sortedBy(name))]<drop/>

#### [cl.name/]

[/for]<drop/>

### Data Types

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

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

#### [dt.name/]

[else][/if]<drop/>

[/for]<drop/>

### Enumerations

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(Enumeration)]<drop/>

#### [dt.name/]

Contains Enumeration Literals:

[for (e:EnumerationLiteral|dt.oclAsType(Enumeration).ownedLiteral)]<drop/>

* [e.name/]:
  + [for (co:Comment | e.ownedComment)]<drop/>
  + <dropEmpty>[co.\_body.clean()/]
  + </dropEmpty>[/for]<drop/>

[/for]<drop/>

[else] [/if]<drop/>

[/for]<drop/>

### Primitives

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(PrimitiveType)]<drop/>

#### [dt.name/]

[else] [/if]<drop/>

[/for]<drop/>

</gendoc><drop/>

## Resource Model

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Resource.notation' element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Diagrams

[for (d : notation::Diagram |notation::Diagram.allInstances()->sortedBy(name))]<drop/>

#### [d.name/]

[/for]<drop/>

</gendoc><drop/>

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Resource.uml’ element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Classes

[for (cl:Class | self.eAllContents(Class)->sortedBy(name))]<drop/>

#### [cl.name/]

[/for]<drop/>

### Data Types

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

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

#### [dt.name/]

[else][/if]<drop/>

[/for]<drop/>

### Enumerations

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(Enumeration)]<drop/>

#### [dt.name/]

Contains Enumeration Literals:

[for (e:EnumerationLiteral|dt.oclAsType(Enumeration).ownedLiteral)]<drop/>

* [e.name/]:
  + [for (co:Comment | e.ownedComment)]<drop/>
  + <dropEmpty>[co.\_body.clean()/]
  + </dropEmpty>[/for]<drop/>

[/for]<drop/>

[else] [/if]<drop/>

[/for]<drop/>

### Primitives

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(PrimitiveType)]<drop/>

#### [dt.name/]

[else] [/if]<drop/>

[/for]<drop/>

</gendoc><drop/>

## Vnf Model

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Vnf.notation' element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Diagrams

[for (d : notation::Diagram |notation::Diagram.allInstances()->sortedBy(name))]<drop/>

#### [d.name/]

[/for]<drop/>

</gendoc><drop/>

<context model=’C:\Users\jjewitt58\git\modelspec\papyrus\ONAP Information Model\Vnf.uml’ element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

### Classes

[for (cl:Class | self.eAllContents(Class)->sortedBy(name))]<drop/>

#### [cl.name/]

[/for]<drop/>

### Data Types

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

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

#### [dt.name/]

[else][/if]<drop/>

[/for]<drop/>

### Enumerations

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(Enumeration)]<drop/>

#### [dt.name/]

Contains Enumeration Literals:

[for (e:EnumerationLiteral|dt.oclAsType(Enumeration).ownedLiteral)]<drop/>

* [e.name/]:
  + [for (co:Comment | e.ownedComment)]<drop/>
  + <dropEmpty>[co.\_body.clean()/]
  + </dropEmpty>[/for]<drop/>

[/for]<drop/>

[else] [/if]<drop/>

[/for]<drop/>

### Primitives

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(PrimitiveType)]<drop/>

#### [dt.name/]

[else] [/if]<drop/>

[/for]<drop/>

</gendoc><drop/>