`<config>  
<output path=’C:\Users\ndavis\git\OnfInfoModelOutput\ModelDescriptions\TR-512.A.8\_OnfCoreIm-Appendix-TimingAndSynchronizationExamples.docx' />  
</config>

<context model=’C:\Users\ndavis\git\ONFInfoModel\OnfModel\CoreModel.uml' element=’{0}’ importedBundles='gmf;papyrus' searchMetamodels='true'/>

<gendoc><drop/>

Change path substrings above from “{path for output files}\” to your local path for the output files and “{path for CoreModel}\” to your local path for the Core Model. <drop/>

DELETE: Prior to publishing this –gd.docx (including for review), change path substrings above from “C:\Users\ndavis\git\OnfInfoModelOutput\” to “{path for output files}\” and from “C:\Users\ndavis\git\ONFInfoModel\OnfModel\” to “{path for CoreModel}\” <drop/>



Core Information Model (CoreModel)

TR-512.A.8

Appendix – Timing and Synchronization Examples

Version 1.5

September 2021

ONF Document Type: Technical Recommendation

ONF Document Name: Core Information Model version 1.5

**Disclaimer**

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Any marks and brands contained herein are the property of their respective owners.

Open Networking Foundation  
1000 El Camino Real, Suite 100, Menlo Park, CA 94025  
[www.opennetworking.org](http://www.opennetworking.org)

©2021 Open Networking Foundation. All rights reserved.

Open Networking Foundation, the ONF symbol, and OpenFlow are registered trademarks of the Open Networking Foundation, in the United States and/or in other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

**Important note**

This Technical Recommendations has been approved by the Project TST, but has not been approved by the ONF board.  This Technical Recommendation is an update to a previously released TR specification, but it has been approved under the ONF publishing guidelines for ‘Informational’ publications that allow Project technical steering teams (TSTs) to authorize publication of Informational documents.  The designation of ‘-info’ at the end of the document ID also reflects that the project team (not the ONF board) approved this TR.

Finalizing this document once generated… delete this text prior to publication:

* Replace “{{..}}” with square brackets (which trip up Gendoc)
* Select text in document from beginning of table of contents (first line) to end of document
  + Click menu item “Update Field” (on this large block of text)
    - if “Update Table…” dialogue appears select “Update entire table”
  + Repeat “update fields” 2 more times (on the same large block of text)
    - if “Update Table…” dialogue appears select “Update entire table”
* Remove reviewer comment

Note that the table of contents and figures need to be updated several times as the table length changes the page numbering and the cross references will need to be re-updated.

Table of Contents

[Disclaimer 2](#_Toc457510737)

[Open Networking Foundation 2](#_Toc457510738)

[Document History 4](#_Toc457510739)

[1 Introduction 4](#_Toc457510740)

[2 References 4](#_Toc457510741)

[3 Definitions 4](#_Toc457510742)

[4 Conventions 4](#_Toc457510743)

[5 Introduction to the Core Network Model 5](#_Toc457510744)

[5.1 Understanding the figures 6](#_Toc457510745)

[6 Forwarding and Termination model detail 6](#_Toc457510746)

[6.1.1.1 LogicalTerminationPoint (LTP) 7](#_Toc457510747)

[6.1.1.2 LayerProtocol (LP) 8](#_Toc457510748)

[6.1.2 Forwarding 9](#_Toc457510749)

[6.1.2.1 ForwardingDomain (FD) 9](#_Toc457510750)

[6.1.2.2 ForwardingConstruct (FC) 10](#_Toc457510751)

[6.1.2.3 FcPort 11](#_Toc457510752)

[6.1.2.4 Link 12](#_Toc457510753)

[6.1.2.5 LinkPort 13](#_Toc457510754)

[6.1.3 NetworkElement, NetworkControlDomain and SdnController 14](#_Toc457510755)

[7 Explanatory Figures 14](#_Toc457510756)

[7.1 Forwarding 14](#_Toc457510757)

[7.1.1 Basic Forwarding 14](#_Toc457510758)

[7.1.2 Forwarding the topology 15](#_Toc457510759)

[7.2 Termination 16](#_Toc457510760)

[7.3 Directionality 22](#_Toc457510761)

[8 Fragment: Insert class <drop/> 28](#_Toc457510762)

[9 Fragment: Insert standard diagram <drop/> 28](#_Toc457510763)

[10 Fragment: Insert small diagram <drop/> 29](#_Toc457510764)

[11 Fragment: Insert attribute row brief <drop/> 30](#_Toc457510765)

[12 Fragment: Start attribute table brief <drop/> 30](#_Toc457510766)

[13 Fragment: Insert Attribute table brief (old) <drop/> 30](#_Toc457510767)

[14 Fragment: Insert Attribute table brief <drop/> 31](#_Toc457510768)

[15 Fragment: Insert Ten Specified Attribute table brief <drop/> 31](#_Toc457510769)

List of Figures

[Figure 1-1 Methodology of IM and DS Development 6](#_Toc430780029)

Document History

| **Version** | **Date** | **Description of Change** |
| --- | --- | --- |
|  |  | Appendix material was not published prior to Version 1.3 |
| 1.3 | September 2017 | Version 1.3 {{Published via wiki only}} |
| 1.3.1 | January 2018 | Addition of text related to approval status. |
| 1.4 | November 2018 | No changes. |
| 1.5 | September 2021 | Enhancements to model structure |

# Introduction to document suite

This document is an appendix of the addendum to the TR-512 ONF Core Information Model and forms part of the description of the ONF-CIM. For general overview material and references to the other parts refer to [TR-512.1](../TR-512.1_OnfCoreIm-Overview.pdf).

## References

For a full list of references see [TR-512.1](../TR-512.1_OnfCoreIm-Overview.pdf).

## Definitions

For a full list of definition see [TR-512.1](../TR-512.1_OnfCoreIm-Overview.pdf).

## Conventions

See [TR-512.1](../TR-512.1_OnfCoreIm-Overview.pdf) for an explanation of:

* UML conventions
* Lifecycle Stereotypes
* Diagram symbol set

## Viewing UML diagrams

Some of the UML diagrams are very dense. To view them either zoom (sometimes to 400%) or open the associated image file (and zoom appropriately) or open the corresponding UML diagram via Papyrus (for each figure with a UML diagram the UML model diagram name is provided under the figure or within the figure).

## Understanding the figures

Figures showing fragments of the model using standard UML symbols and also figures illustrating application of the model are provided throughout this document. Many of the application-oriented figures also provide UML class diagrams for the corresponding model fragments (see [TR-512.1](../TR-512.1_OnfCoreIm-Overview.pdf) for diagram symbol sets). All UML diagrams depict a subset of the relationships between the classes, such as inheritance (i.e. specialization), association relationships (such as aggregation and composition), and conditional features or capabilities. Some UML diagrams also show further details of the individual classes, such as their attributes and the data types used by the attributes.

## Appendix Overview

This document is part of the Appendix to TR-512. An overview of the Appendix is provided in [TR-512.A.1](TR-512.A.1_OnfCoreIm-AppendixOverview.pdf).

# Introduction to this Appendix document

This document provides a description of time and frequency synchronization in a telecommunications network and provides examples of the use of the CIM abstractions to model these synchronization functions.

The examples in this document extend the simple examples given in [TR-512.11](TR-512.11_OnfCoreIm-ProcessingConstruct.pdf).

# Network synchronization overview

The figure below shows an example of a (simple) synchronization network.



Figure 3-1 Example synchronization distribution network

From a management perspective, the Master Clock can be represented by an instance of the clock and the node containing a network equipment clock can be represented by instances of Processing Construct (PC), that encompass LTP, FD and clock, with links to interconnect the nodes. Typically, only a subset of the links in a network are enabled, by management/control actions, to support the transfer timing information. Two types of information may be carried over these links: The timing information (frequency or time stamp) and (optionally) information about the source of the timing information. These links are typically used to carry both timing information and network traffic but some may be dedicated to synchronization. This is illustrated in the figure below.



Figure 3-2 Full timing distribution topology

The ensure the correct operation of the synchronization network each node will select one of the timing inputs and distribute the resultant timing information. This results in a pruned topology as illustrated below.



Figure 3-3 Reduced/pruned timing distribution topology

The pruned topology could be defined manually or: By allowing the nodes to be autonomously configured by PTP (using BMCA) or by Sync Status Messages (SSM) or: by some combination of manual and autonomous configuration. If some degree of autonomous control is permitted, then the selected topology will be updated when a failure occurs. Typically, a network operator would define the set of inputs that are used in the autonomous selection process and the priority assigned to each of these enabled inputs.

To ensure correct operation of the synchronization network the input used by the network element clocks should, when possible, be derived from the master clock. It is essential that, under fault conditions, the formation of timing loops is prevented.

Standard SSM messages only provide clock quality information which is insufficient to guarantee loop free operation under fault conditions so the links that are enabled to support timing information must be selected to avoid timing loops.

The PTP protocol provides both clock quality information and additional information about the clock source including its identity and domain membership. This allows the BCMA to select the best quality input, from those clocks within the timing domain, and avoid timing loops.

The simple network example shows a single master clock, however, in a typical synchronization network additional (secondary) master clocks are present.

# Processing of timing information in a node

To fully describe and manage the synchronization aspects of a node the PC describe above must be expanded to expose the internal details. This expansion, for a PTP clock, is shown in the figure below.



Figure 4-1 PTP node expansion

The termination of the trails carrying the timing information is represented with an LTP with the appropriate layer protocol(s). The extraction of the components of the timing information (time stamp and timing source information) is represented by a LTP with a layer protocol of synchronization (Sync LTP). The time stamp is forwarded to a FC, this FC has m inputs and one output. The timing source information is forwarded to a Configuration and switch controller (C&SC). The Sync LTP and the relationship to the FC and C&SC is only present for those inputs that have been enabled to support synchronization. The management/control system also assigns a selection priority to each input. The C&SC uses the timing source information together with the locally configured priority and any local commands to configure the FC to forward the time stamp from the selected input to the clock. If none of the inputs are selected the clock enters hold-over or free-run mode. The clock is an analog device that essentially integrates the time stamps and produces a smoothed output time stamp. The output time stamp is forwarded to all of the Sync LTP that have been enabled to support timing information. The clock source/quality information is also forwarded to these sync LTP. The C&SC informs the sync LTPs which input has been selected and the Sync LTPs modify the clock source/quality information.

The same abstraction, with the appropriate parameters, can be used for frequency synchronization.

Typically, a node will support redundant clocks. This is illustrated in the figure below.



Figure 4-2 Node with redundant clocks

Both clocks receive the same time stamp input, one is selected as the master and provides a phase alignment signal to the slave. The FC protection selects the master clock and forwards the time stamp to the enabled sync LTPs. The clock source/quality information is also forwarded to these sync LTP by the FC protection. The master can be selected either automatically based on the status of the clocks, or by a local management inputs.

## Refactoring the synch model

If visibility of the operation of the node is not required the sync model can be refactored into a Processing Construct that encompasses the Sync LTPs, FC-selector, C&SC, clock and FC protection. This is illustrated in the figure below.



Figure 4-3 Node represented by a processing construct

In this case the Sync LTP attributes are reported against ports on the PC and the parameters of the clock are reported against the PC.

# Synchronization model attributes

The attributes provided in this section[[1]](#footnote-1) support frequency synchronization using SSM and time synchronization using the Telecom profile of {{IEEE 1588}}. These parameters need to be validated by ITU-T SG15, Q13/15 and Q14/15.

## Existing NE object

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| Sync\_Support\_freq | Indicate whether the NE has the capability to support frequency synchronization. | Enumeration – Read only: {{ITU-T G.813}}; {{ITU-T G.812}}; etc. |
| Freq Sync support enabled | Allows the frequency sync functions to be deactivated | Boolean – Read/Write:  Default: enabled |
| Sync\_Support\_time | Indicate whether the NE has the capability to support time synchronization. | Enumeration – Read only: Boundary clock (BC); Transparent clock (TC) etc. |
| Time Sync support enabled | Allows the time sync functions to be deactivated | Boolean – Read/Write: Default: enabled |

## Clock

### Frequency sync (SSM) pac

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| system clock ID | The ID of the SyncClock\_Frequency object. | Object ID  frequ and PTP could use a common attribute definition  Object ID – read only |
| associated node ID | The SyncClock\_Frequency object is associated with the NE of this node ID. | Object ID frequ and PTP could use a common attribute definition  Object ID – read only |
| run-mode | The run-mode of the frequency system clock, such as freerun, locked, and holdover. | Enumeration - Read only; Freerun; locked; holdover frequ and PTP could use a common attribute definition  Enumeration – read only |
| internal clock SSM | The SSM quality level of internal clock of the NE. | Enumeration - Read only {{ITU-T G.813}}; {{ITU-T G.812}}; ??? frequ and PTP could use a common attribute definition  Enumeration – read only |

### Time Sync (PTP) pac

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| PTP system clock ID | The ID of the SyncClock\_Time object. | Object ID frequ and PTP could use a common attribute definition  Object ID – read only |
| associated node ID | The SyncClock\_Time object is associated with the NE of this node ID. | Object ID frequ and PTP could use a common attribute definition  Object ID – read only |
| PTP enable status | Indicate whether the NE enables PTP function or not. | Boolean – read/write?  Boolean – read/write |
| run-mode | The run-mode of the PTP system clock, such as tracing and non-tracing. | Enumeration - Read only: Freerun; locked; holdover frequ and PTP could use a common attribute definition  Enumeration – read only |
| PTP domain | The PTP domain number of the NE. | Integer or string – Read/write?  Integer – read/write |
| PTP device-type | Three PTP device types are included: boundary clock (BC), transparent clock (TC), and ordinary clock (OC). | Enumeration - Read only boundary clock (BC), transparent clock (TC), and ordinary clock (OC). frequ and PTP could use a common attribute definition with different enumerations  Enumeration – read/write |
| PTP slaveonly | Indicate whether the NE can only be used as PTP slave or not. | Boolean - Read only?  Boolean – read/write |
| PTP source dataset | The PTP status dataset of current tracing source. | Ordered list - Read only:  grandmasterIdentity – Object ID – Read only,  parent ID – Object ID – Read only,  priority 1 – Integer – Read only,  priority 2 – Integer – Read only,  clockClass – Integer – Read only,  accuracy – Integer – Read only,  offsetScaledLogVariance – Integer – Read only,  timesource – Enumeration – Read only,  stepsRemoved – Integer – Read only,  currentUtcOffset – Integer – Read only,  ptpTimescale – Enumeration – Read only,  timeTraceable – Boolean – Read only,  frequencyTraceable – Boolean – Read only,  {{IEEE 1588}} protocol version – Integer – Read only,,  current absolute time – Integer – Read only. |
| PTP default dataset | The PTP status dataset of internal clock of the NE | Ordered list – Read/Write:  CLK ID – Object ID – Read/Write,  priority 1 – Integer – Read/Write,  priority 2 – Integer – Read/Write,  clockClass – Integer – Read/Write,  accuracy – Integer – Read/Write,  offsetScaledLogVariance – Integer – Read/Write,  timesource – Integer – Read/Write,  stepsRemoved – Integer – Read/Write,  {{IEEE 1588}} protocol version – Integer – Read/Write. |

## Sync LTP

### SSM in band pac

All of the attributes that report/manage SSM quality level can use a common enumeration.

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| clock port ID | The ID of the SyncLTP\_In\_Band\_Clock object. | Object ID – read only |
| associated port ID | The SyncLTP\_In\_Band\_Clock object is associated with the physical port LTP of this port ID. | Object ID – read only |
| SSM output enable status | Indicate whether to send SSM messages or not. | Boolean – read/write |
| SSM information | Current input and output SSM quality levels used by the port. The SSM quality level can be set manually or automatically. | Enumeration – Read/Write  e.g: PRC, SSU-A, SSU-B, QL-SEC, DNU, etc. |
| SSM mode | Indicate whether to use manual or automatic input and output SSM quality levels. | Enumeration– Read/Write:  Manual or Automatic |
| SSM configuration | The input and output SSM quality levels set manually. | Enumeration – read/write |

### SSM external clock

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| external port ID | The ID of the SyncLTP\_External\_Clock object. | Object ID – read only |
| external port enable status | Indicate whether to enable this external port or not. | Boolean – read/write |
| bits-type | The type of this port, such as 2048kb/s or 2048kHz. | Enumeration read only |
| SSM sa-bit | Indicate which sa-bit bits are used for carrying input and output SSM quality levels. | Enumeration – read/write |
| SSM out-threshold | The external port stops transmitting when the SSM quality level is lower than the threshold. | Enumeration – read/write |
| SSM information | Current input and output SSM quality levels used by the port. The SSM quality level can be set manually or automatically. | Enumeration – read only |
| SSM mode | Indicate whether to use manual or automatic input and output SSM quality levels. | Enumeration– Read/Write:  Manual or Automatic |
| SSM configuration | The input and output SSM quality levels set manually. | Enumeration – read/write |

### PTP pac

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| PTP port ID | The ID of the SyncLTP\_PTP object. | Object ID – read only |
| associated port ID | The SyncLTP\_PTP object is associated with the physical port LTP of this port ID. | Object ID – read only |
| PTP port enable status | Indicate whether to enable this PTP port or not. | Boolean – read/write |
| PTP port state | The current PTP state of the PTP port, such as master, slave, passive, initializing, listening, premaster, uncalibrated, and faulty. | Enumeration – read only |
| PTP asymmetry-correction | The asymmetry correction value of this PTP port. | Integer, nano seconds – read/write same as external time port delay compensation?  Integer – read/write |
| PTP clock-step | Indicate whether one-step or two-step mechanism is adopted. | Enumeration – read only |
| PTP udp-egress configuration | The configuration of PTP UDP encapsulation, including source IP address, destination IP address and IPv4/IPv6 protocol. | Ordered list – Read/Write:  source IP address – String – Read/Write,  destination IP address – String – Read/Write,  IPv4/IPv6 protocol – Enumeration – Read/Write. |
| PTP mac-egress configuration | The configuration of PTP MAC encapsulation, including source MAC address, destination MAC address and VLAN configuration. | Ordered list – Read/Write:  source MAC address – String – Read only,  destination MAC address – String – Read only,  VLAN configuration – String – Read/Write |
| PTP announce-interval | The sending interval of PTP announce message. | Integer: milli seconds – read/write |
| PTP announce receipt-timeout | It is used for fault detection of PTP announce messages. | Integer: milli seconds – read/write |
| PTP sync-interval | The sending interval of PTP Sync message. | Integer: milli seconds – read/write |
| PTP min-delayreq-interval | The sending interval of PTP Delay\_req message. | Integer: milli seconds – read/write |

### PTP 1PPS + ToD pac

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Type** |
| external port ID | The ID of the SyncLTP\_External\_Time object. | Object ID – read only |
| external time port status | Indicate whether this external time port is used as an input or output port. | Boolean - read only |
| external time port dataset | The status dataset of this port | Ordered list – Read/Write:  grandmasterIdentity – Object ID – Read/Write,  priority 1 – Integer – Read/Write,  priority 2 – Integer – Read/Write,  clockClass – Integer – Read/Write,  accuracy – Integer – Read/Write,  offsetScaledLogVariance – Integer – Read/Write,  timesource – Integer – Read/Write,  stepsRemoved – Integer – Read/Write,  currentUtcOffset – Integer – Read/Write. |
| external time port delay compensation | The delay compensation value of this external port. | Integer, nano seconds – read/write Same as PTP asymmetry-correction?  Integer – read/write |

**End of Document**

</gendoc><drop/>

To take latest template: <drop/>

* delete text from “Template version…” to end of file <drop/>
* insert a line in “Normal” style<drop/>
* insert text (Insert 🡪 Object 🡪 Text from File… (alt njf)) from: <drop/>
  + TR-512.GT\_OnfCoreIm-CommonGendocTemplate-Fragments.docx <drop/>

Template version 0.0.10 17 September 2017 <drop/>

# Fragment: Insert class <drop/>

<fragment name=’insertClass’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’cl’ type=’uml::Class’/><drop/>  
<arg name=’className’ type=’String’/><drop/>  
<arg name=’packageName’ type=’String’/><drop/>  
[if (not cl.qualifiedName.contains(packageName))]<drop/>  
[else] <drop/>  
[if(cl.name.contains(className))]<drop/>

Qualified Name: [cl.qualifiedName/]

[for (co:Comment | cl.ownedComment)]<drop/>

<dropEmpty>[cleanAndFormat(co.\_body.clean())/]</dropEmpty>

[/for]<drop/>  
[if (cl.isAbstract)]<drop/>

This class is abstract.

[/if]<drop/>

[if (cl.oclAsType(uml::Class).general ->notEmpty())]<drop/>

Inherits properties from:

[for (gen:Class | cl.oclAsType(uml::Class).general)]<drop/>

* [gen.name/]

[/for]<drop/>

[/if]<drop/>

[for (st:Stereotype | cl.getAppliedStereotypes())]<drop/>  
[if(not st.name.contains(‘OpenModelClass’))]<drop/>

This class is [st.name/].

[else] <drop/>  
[/if]<drop/>  
[/for]<drop/>  
[else] <drop/>  
[/if]  
[/if]  
</fragment><drop/>

# Fragment: Insert standard diagram <drop/>

<fragment name=’insertStandardDiagram’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’p’ type=’uml::Package’/><drop/>  
<arg name=’diagramName’ type=’String’/><drop/>  
<arg name=’diagramTitle’ type=’String’/><drop/>

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

[if d.name.contains(diagramName)]

<drop/>

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

CoreModel diagram: [d.name/]

Figure 6-2 [diagramTitle/]

[else]<drop/>

[/if]<drop/>

[/for]<drop/>  
</fragment><drop/>

# Fragment: Insert small diagram <drop/>

<fragment name=’insertSmallDiagram’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’p’ type=’uml::Package’/><drop/>  
<arg name=’diagramName’ type=’String’/><drop/>  
<arg name=’diagramTitle’ type=’String’/><drop/>

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

[if d.name.contains(diagramName)]

<drop/>

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

CoreModel diagram: [d.name/]

Figure 6-2 [diagramTitle/]

[else]<drop/>

[/if]<drop/>

[/for]<drop/>  
</fragment><drop/>

# Fragment: Insert attribute row brief not Obsolete<drop/>

<fragment name=’insertAttributeRowBriefNotObsolete’ importedBundles=’commons;gmf;papyrus’><drop/>

Does not work unless we have Mature stereotype… <drop/>  
<arg name=’p’ type=’uml::Property’/><drop/>

[for (st:Stereotype | p.getAppliedStereotypes())]<drop/>

[if(not st.name.contains(‘OpenModelAttribute’))]

[if(not st.name.contains(‘Obsolete’))]

| [p.name/] | [for (st:Stereotype | p.getAppliedStereotypes())]<drop/>  [if(not st.name.contains(‘OpenModelAttribute’))] [st.name/]  [/if]<drop/>  [/for]<drop/>  Do NOT remove the previous line as word throws an error if the cell is empty <drop/> | [if p.ownedComment->notEmpty()]<drop/>  [for (c:Comment | p.ownedComment)] <drop/>  [cleanAndFormat(c.\_body.clean())/]  [/for]  [else] [if (p.name.contains (‘\_’))]See referenced class  [else]To be provided  [/if]<drop/>  [/if]<drop/>  Do NOT remove the previous line as word throws an error if the cell is empty <drop/> |
| --- | --- | --- |

[/if]<drop/>

[/if]<drop/>

[/for]<drop/>  
</fragment><drop/>

# Fragment: Insert attribute row brief <drop/>

<fragment name=’insertAttributeRowBrief’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’p’ type=’uml::Property’/><drop/>

| [p.name/] | [for (st:Stereotype | p.getAppliedStereotypes())]<drop/>  [if(not st.name.contains(‘OpenModelAttribute’))] [st.name/]  [/if]<drop/>  [/for]<drop/>  Do NOT remove the previous line as word throws an error if the cell is empty <drop/> | [if p.ownedComment->notEmpty()]<drop/>  [for (c:Comment | p.ownedComment)] <drop/>  [cleanAndFormat(c.\_body.clean())/]  [/for]  [else] [if (p.name.contains (‘\_’))]See referenced class  [else]To be provided  [/if]<drop/>  [/if]<drop/>  Do NOT remove the previous line as word throws an error if the cell is empty <drop/> |
| --- | --- | --- |

</fragment><drop/>

# Fragment: Start attribute table brief <drop/>

<fragment name=’insertAttributeTableHeader’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’cl’ type=’uml::Class’/><drop/>

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Lifecycle Stereotype (empty = Mature)** | **Description** |

</fragment><drop/>

# Fragment: Insert Attribute table brief <drop/>

<fragment name=’insertAttributeTableBrief’ importedBundles=’commons;gmf;papyrus’ importedFragments='insertAttributeTableHeader;insertAttributeRowBrief’><drop/>  
<arg name=’cl’ type=’uml::Class’/><drop/>  
[if cl.ownedAttribute->notEmpty()]<drop/>

Table 1: Attributes for [cl.name/]

<table><drop/>

[cl.insertAttributeTableHeader ()/]

[for (p:Property|cl.ownedAttribute)]<drop/>

[if (not p.name.contains(‘\_’))]<drop/>

[p.insertAttributeRowBrief ()/]

[/if]<drop/>

[/for]<drop/>

[for (p:Property|cl.ownedAttribute)]<drop/>

[if (p.name.contains(‘\_’))]<drop/>

[p.insertAttributeRowBrief ()/]

[/if]<drop/>

[/for]<drop/>

</table><drop/>

[/if]<drop/>

</fragment><drop/>

# Fragment: Insert Ten Specified Attribute table brief <drop/>

<fragment name=’insertTenSpecifiedAttributeTableBrief’ importedBundles=’commons;gmf;papyrus’ importedFragments='insertAttributeTableHeader;insertAttributeRowBrief’><drop/>  
<arg name=’cl’ type=’uml::Class’/><drop/>

<arg name=’p1’ type=‘String’/><drop/>

<arg name=’p2’ type=‘String’/><drop/>  
<arg name=’p3’ type=‘String’/><drop/>  
<arg name=’p4’ type=‘String’/><drop/>  
<arg name=’p5’ type=‘String’/><drop/>  
<arg name=’p6’ type=‘String’/><drop/>  
<arg name=’p7’ type=‘String’/><drop/>  
<arg name=’p8’ type=‘String’/><drop/>  
<arg name=’p9’ type=‘String’/><drop/>  
<arg name=’p10’ type=‘String’/><drop/>  
[if cl.ownedAttribute->notEmpty()]<drop/>

Table 1: Attributes for [cl.name/]

<table><drop/>

[cl.insertAttributeTableHeader ()/]

[for (p:Property|cl.ownedAttribute)]<drop/>

[if (p.name.contains(p1) or p.name.contains(p2) or p.name.contains(p3) or p.name.contains(p4) or p.name.contains(p5) or p.name.contains(p6) or p.name.contains(p7) or p.name.contains(p8) or p.name.contains(p9) or p.name.contains(p10))]<drop/>

[if (not p.name.contains(‘\_’))]<drop/>

[p.insertAttributeRowBrief ()/]

[/if]<drop/>

[/if]<drop/>

[if (p.name.contains(p1) or p.name.contains(p2) or p.name.contains(p3) or p.name.contains(p4) or p.name.contains(p5) or p.name.contains(p6) or p.name.contains(p7) or p.name.contains(p8) or p.name.contains(p9) or p.name.contains(p10))]<drop/>

[if (p.name.contains(‘\_’))]<drop/>

[p.insertAttributeRowBrief ()/]

[/if]<drop/>

[/if]<drop/>

[/for]<drop/>

</table><drop/>

[/if]<drop/>

</fragment><drop/>

# Fragment: Insert DataType <drop/>

<fragment name=’insertDataType’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’dt’ type=’uml::DataType’/><drop/>  
<arg name=’dataTypeName’ type=’String’/><drop/>  
<arg name=’packageName’ type=’String’/><drop/>  
[if (dt.qualifiedName.contains(packageName))]<drop/>  
[if(dt.name.contains(dataTypeName))]<drop/>

Qualified Name: [dt.qualifiedName/]

[for (co:Comment | dt.ownedComment)]<drop/>

<dropEmpty>[cleanAndFormat(co.\_body.clean())/]</dropEmpty>

[/for]<drop/>  
[if (dt.oclAsType(uml::DataType).general ->notEmpty())]<drop/>

Inherits properties from:

[for (tp:DataType | dt.oclAsType(uml::DataType).general)]<drop/>

* [tp.name/]

[/for]<drop/>

[for (gen:Class | dt.oclAsType(uml::DataType).general)]<drop/>

* [gen.name/]

[/for]<drop/>

[/if]<drop/>

[for (st:Stereotype | dt.getAppliedStereotypes())]<drop/>  
This class is [st.name/].

[/for]<drop/>  
[else] <drop/>  
[/if]  
[/if]  
</fragment><drop/>

# Fragment: Start Data Type attribute table brief <drop/>

<fragment name=’insertDataTypeAttributeTableHeader’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’dt’ type=’uml::DataType’/><drop/>

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Lifecycle Stereotype (empty = Mature)** | **Description** |

</fragment><drop/>

# Fragment: Insert Data Type Attribute table brief <drop/>

<fragment name=’insertDataTypeAttributeTableBrief’ importedBundles=’commons;gmf;papyrus’ importedFragments='insertDataTypeAttributeTableHeader;insertAttributeRowBrief’><drop/>  
<arg name=’dt’ type=’uml::DataType’/><drop/>  
[if dt.ownedAttribute->notEmpty()]<drop/>

Table 1: Attributes for [dt.name/]

<table><drop/>

[dt.insertDataTypeAttributeTableHeader ()/]

[for (p:Property|dt.ownedAttribute)]<drop/>

[p.insertAttributeRowBrief ()/]

[/for]<drop/>

</table><drop/>

[/if]<drop/>

</fragment><drop/>

# Fragment: Insert enums <drop/>

<fragment name=’insertEnums’ importedBundles=’commons;gmf;papyrus’><drop/>  
<arg name=’dt’ type=’uml::DataType’/><drop/>

#### [dt.name/]

Qualified Name: [dt.qualifiedName/]

[for (co:Comment | dt.ownedComment)]<drop/>

<dropEmpty>[cleanAndFormat(co.\_body.clean())/]</dropEmpty>

[/for]<drop/>

Applied stereotypes:

[if dt.getAppliedStereotypes()->notEmpty()] <drop/>

[for (st:Stereotype | dt.getAppliedStereotypes())]<drop/>

* [st.name/]

[/for]<drop/>

[else] No stereotypes applied

[/if]<drop/>

[if (dt.oclAsType(uml::DataType).general ->notEmpty())]<drop/>

Inherits literals from:

[for (tp:DataType | dt.oclAsType(uml::DataType).general)]<drop/>

* [tp.name/]

[/for]

[/if]<drop/>

[if (dt.oclAsType(Enumeration).ownedLiteral->notEmpty())]<drop/>

Contains Enumeration Literals:

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

* [e.name/]:
  + [for (co:Comment | e.ownedComment)]<drop/>
  + <dropEmpty>[cleanAndFormat(co.\_body.clean())/]
  + </dropEmpty>[/for]<drop/>
  + [if dt.getAppliedStereotypes()->notEmpty()] <drop/>
  + Applied stereotypes:
    - [for (st:Stereotype | e.getAppliedStereotypes())]<drop/>
    - [st.name/]
    - [/for]<drop/>
  + [/if]<drop/>

[/for]<drop/>

[/if]<drop/>

</fragment><drop/>

1. NE sync status has not been fully covered in this release. [↑](#footnote-ref-1)