Content generated from UML model file startUmlFile..endUml.

TOC

[1 Intro 1](#_Toc244874931)

[2 Package structure (Normative) 1](#_Toc244874932)

[2.1 General 1](#_Toc244874933)

[2.2 startUmlPackage.Dummy.endUml 1](#_Toc244874934)

[2.3 startUmlPackage.IEC61970.endUml 1](#_Toc244874935)

TOF

**Error! No table of figures entries found.**

TOT

**Error! No table of figures entries found.**

# Intro

CIM version used is startUmlAttribute.IEC61970CIMVersion.version.endUml.

Inexistent attribute: startUmlAttribute.IEC61970CIMVersion.dummy.endUml.

Namespace names:

**IEC61850 namespace name:** “startUmlIec61850NsName.IEC61850\_7\_2Namespace.endUml”

**IEC61850 namespace name:** “startUmlIec61850NsName.IEC61850\_7\_3Namespace.endUml”

**IEC61850 namespace name:** “startUmlIec61850NsName.IEC61850\_7\_4Namespace.endUml”

**IEC61850 namespace name:** “startUmlIec61850NsName.IEC61850\_7\_420Namespace.endUml”

Figure 1 shows the main diagram:

startUmlDiagram.IEC61970.Main.endUml

Figure 1 – my first figure

Here we add a table, without referencing it, to ensure we number the following tables correctly.

Table 1 – my first test table

|  |  |
| --- | --- |
|  |  |
|  |  |

Next will show nothing (inexistent diagram):

startUmlDiagram.DummyPackage.DummyDiagram.endUml

Figure 2 – inexisting diagram

Here we add another table, to ensure we number the following tables correctly.

Table 2 - my second test table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Figure below should be always printed: it’s in a Doc\*\*\* package:

startUmlDiagram.DocIEC61970.MyDocFigure.endUml

Figure 3 – Diagram referenced in template, not printed in the regular package doc

# Package structure (Normative)

## General

Content generated from UML model file startUmlFile..endUml.

## start%UmlPackage.IEC61970.endUml

# Individual arbitrary packages

Below, first package does not exist.

Second package should be printed only if config option is set (to include informative and private packages).

We show below that any sub-package (and all its contents) can be put in a package placeholder. For official IEC documents for CIM, you do not need to specify a separate placeholder for each sub-package – you can just use a single package placeholder for IEC61970 or IEC61968 or IEC62325 instead, and all their normative content will be included automatically – see previous section.

## startUmlPackage.Dummy.endUml

## startUmlPackage.InformativeAndPrivate.endUml

## startUmlPackage.Domain.endUml

## startUmlPackage.Ext1.endUml

# Optional data index (e.g., in appendix)

This has been implemented for IEC61850-7-3 and -7-4, but can also be used for assessing CIM attributes and making their documentation and types uniform. Currently, we print only attributes; if needed, we could do the same for association ends.

## startUmlDataIndex.Core.endUml

# IEC61850 (for testing formats, dumb examples)

## Abbreviated terms

startUmlAbbreviations.Abbreviations.endUml

## startUmlPresenceConditions.PresenceConditions.endUml

## startUmlPackage.CDCAnalogueInfo.endUml

## startUmlPackage.LogicalNodes.endUml

## startUmlDataIndex.CommonDataClasses.endUml

## startUmlPackage.DAEnums.endUml

## startUmlSclEnums.DAEnums.endUml

## startUmlDataIndex.LogicalNodes.endUml

# Individual classes and diagram notes

This is to test the newly introduces placeholders: for individual classes and for diagram notes.

Note for existing diagram comes next:

startUmlDiagNote.IEC61970.Main.endUml

Note for inexisting diagram comes next:

startUmlDiagNote.IEC619\_70.Main.endUml

One existing CIM class, one existing 61850 class, one enumeration and one inexisting class follow.

## startUmlClass.Core.BasePower.endUml

## startUmlClass.CDCStatusInfo.DPS.endUml

## startUmlClass.Domain.UnitSymbol.endUml

## startUmlClass.Core.InexistingClassName.endUml

# For debugging

## 61850: range vs. minVal/maxVal

### startUmlClass.ConstructedDAs.RangeConfig.endUml

### startUmlClass.CDCStatusInfo.HST.endUml

## Inherited deprecation

### startUmlClass.Core.ConnectivityNodeContainer.endUml

### startUmlClass.Core.DumbSubterminal.endUml