Going into details - Property Sets with RailCOMPLETE

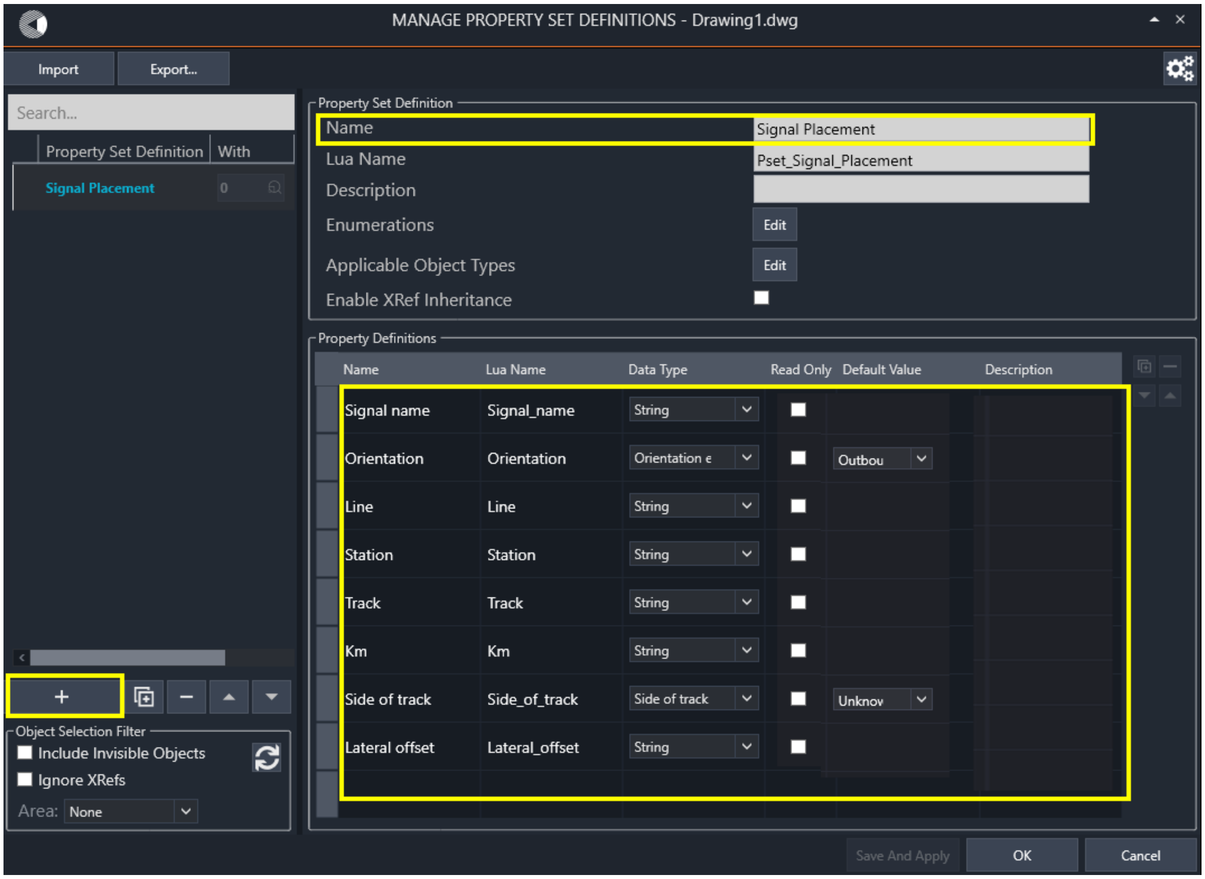
Defining a Custom Property Set

A **Property Set** has a recognizable **Property Set Name** and an associated collection of **Properties**. Each property has a **Property** Name and a **Property Value** field that will hold a value consistent with a specified **Data Type**. Let us explore this example from the railway realm in detail:

SignalPlacement

Signal name	*37113(L)*	*Text (String)*
Orientation	*Outbound*	*Enumeration (Outbound | Inbound)*
Line	*Mountain Railway*	*Text (String)*
Station	*Hillside St. (No.37)*	*Text (String)*
Track	*1*	*Text (String)*
Km	*462,031*	*Number (Double)*
Side of track	*Left*	*Enumeration (Left | Right | Unknown)*
LateralOffset	*3,850*	*Number (Double)*

Use the **RailCOMPLETE Property Set Definition Manager** to enter the property set's definition and possible default values. Press the '+' button to start a new property set definition:



Property Set Definition

Name: Signal Placement

Lua Name: Pset_Signal_Placement

Description:

Enumerations

Applicable Object Types

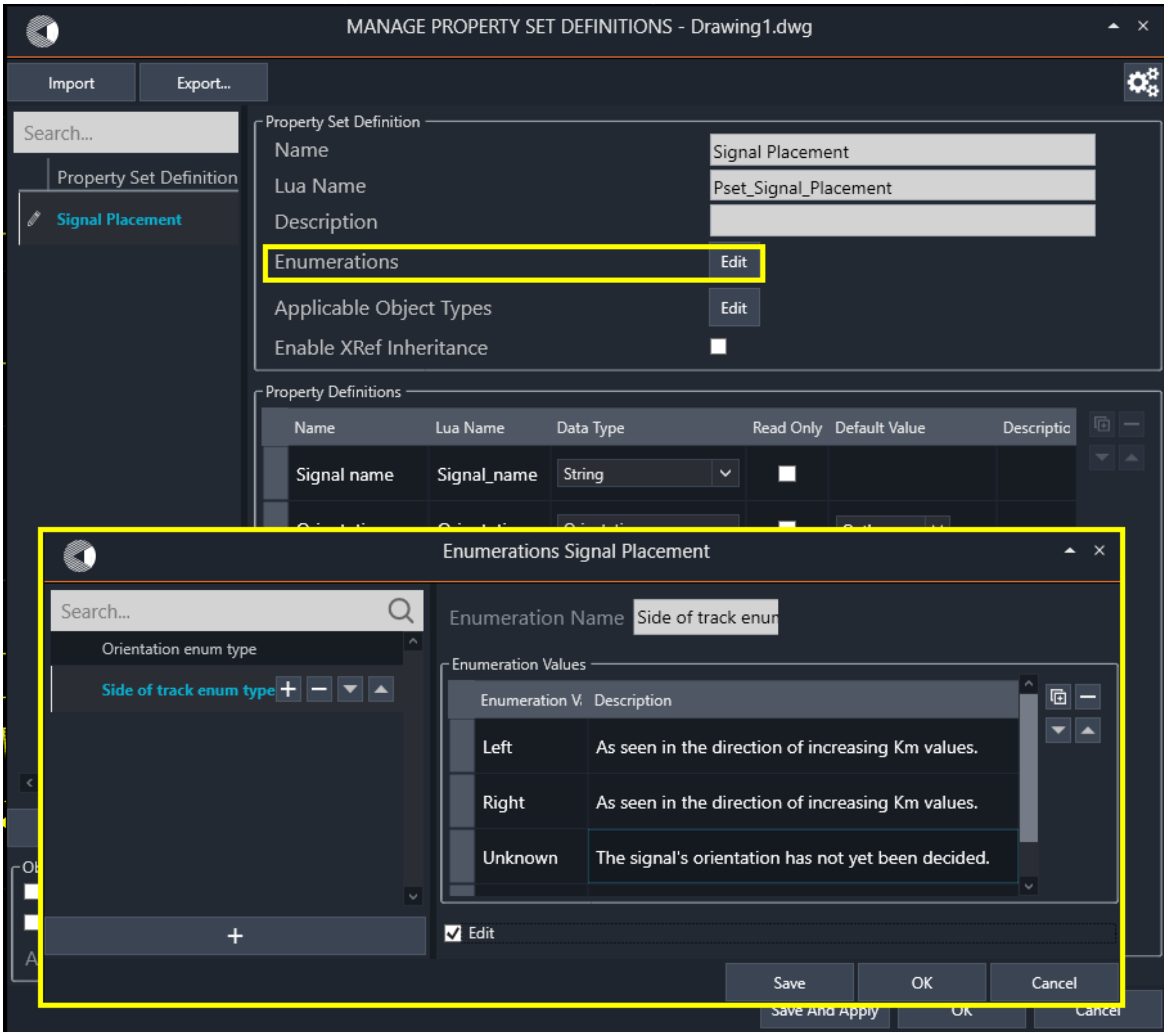
Enable XRef Inheritance □

Property Definitions

Name	Lua Name	Data Type	Read Only	Default Value	Description
Signal name	Signal_name	String	□		
Orientation	Orientation	Orientation e	□	Outbou	
Line	Line	String	□		
Station	Station	String	□		
Track	Track	String	□		
Km	Km	String	□		
Side of track	Side_of_track	Side of track	□	Unknov	
Lateral offset	Lateral_offset	String	□		

Figure 1 **Example Pset 'Signal Placement'**

An enumeration type acts as an extra, custom defined, data type. To modify an enumeration, use the **Property Set Enumerations Editor**:



Name: Signal Placement

Lua Name: Pset_Signal_Placement

Description:

Enable XRef Inheritance □

Name	Lua Name	Data Type	Read Only	Default Value	Descriptio
Signal name	Signal_name	String	□		

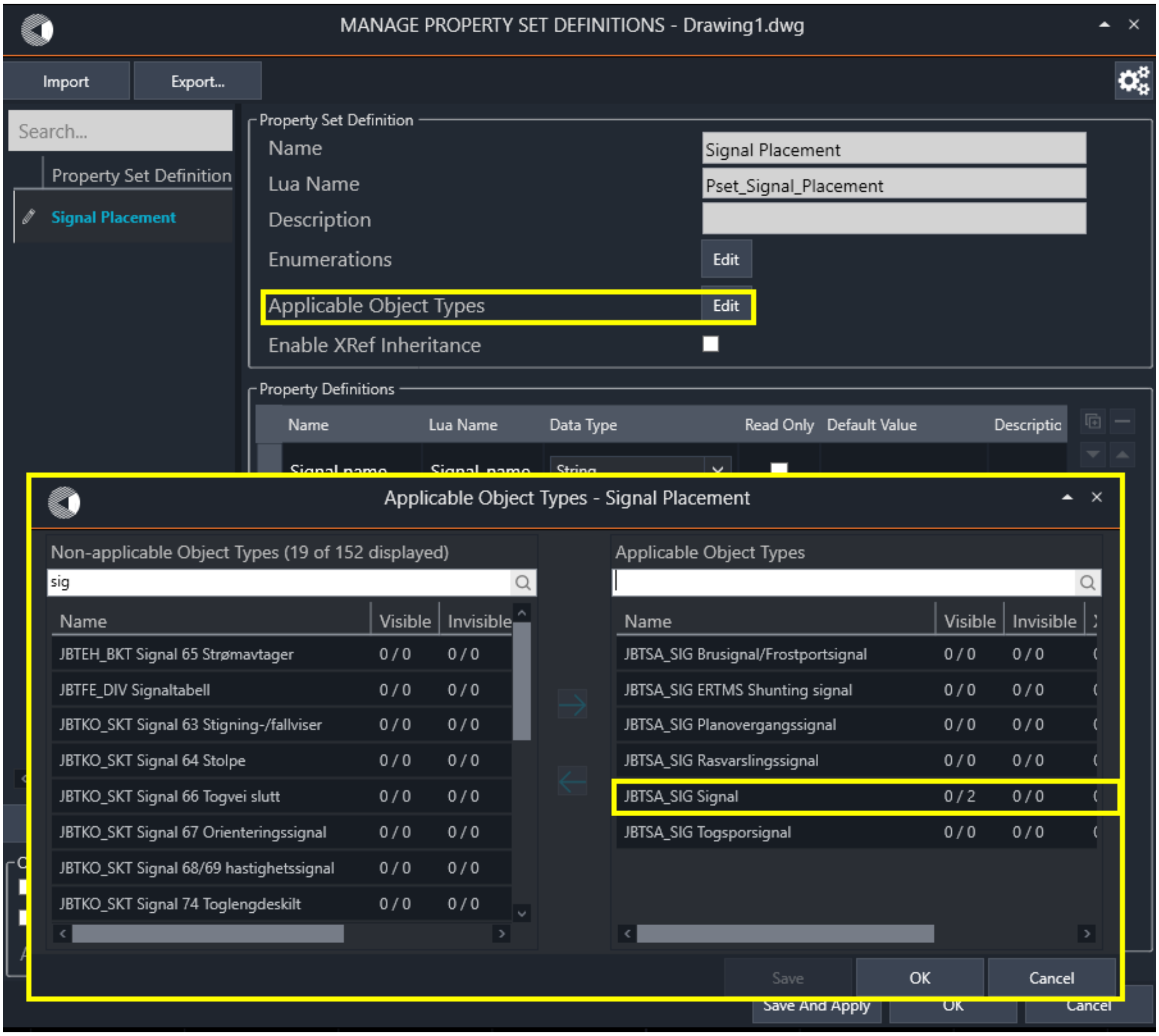
Enumeration Name: Side of track enur

Enumeration V.	Description
Left	As seen in the direction of increasing Km values.
Right	As seen in the direction of increasing Km values.
Unknown	The signal's orientation has not yet been decided.

☑ Edit

Figure 2 Enumerations editor

RailCOMPLETE keeps track for you about which RC object types are the intended users of a given property set. This information is handled in the **Property Set Applicability Editor**:



Property Set Definition

Name	Signal Placement
Lua Name	Pset_Signal_Placement
Description	
Enumerations	Edit
Applicable Object Types	Edit
Enable XRef Inheritance	□

Property Definitions

Name	Lua Name	Data Type	Read Only	Default Value	Descriptio
Signal name	Signal name	String	□		

Non-applicable Object Types (19 of 152 displayed)

Name	Visible	Invisible
JBTEH_BKT Signal 65 Strømavtager	0 / 0	0 / 0
JBTFE_DIV Signaltabell	0 / 0	0 / 0
JBTKO_SKT Signal 63 Stigning-/fallviser	0 / 0	0 / 0
JBTKO_SKT Signal 64 Stolpe	0 / 0	0 / 0
JBTKO_SKT Signal 66 Togvei slutt	0 / 0	0 / 0
JBTKO_SKT Signal 67 Orienteringssignal	0 / 0	0 / 0
JBTKO_SKT Signal 68/69 hastighetssignal	0 / 0	0 / 0
JBTKO_SKT Signal 74 Toglengdeskilt	0 / 0	0 / 0

Applicable Object Types

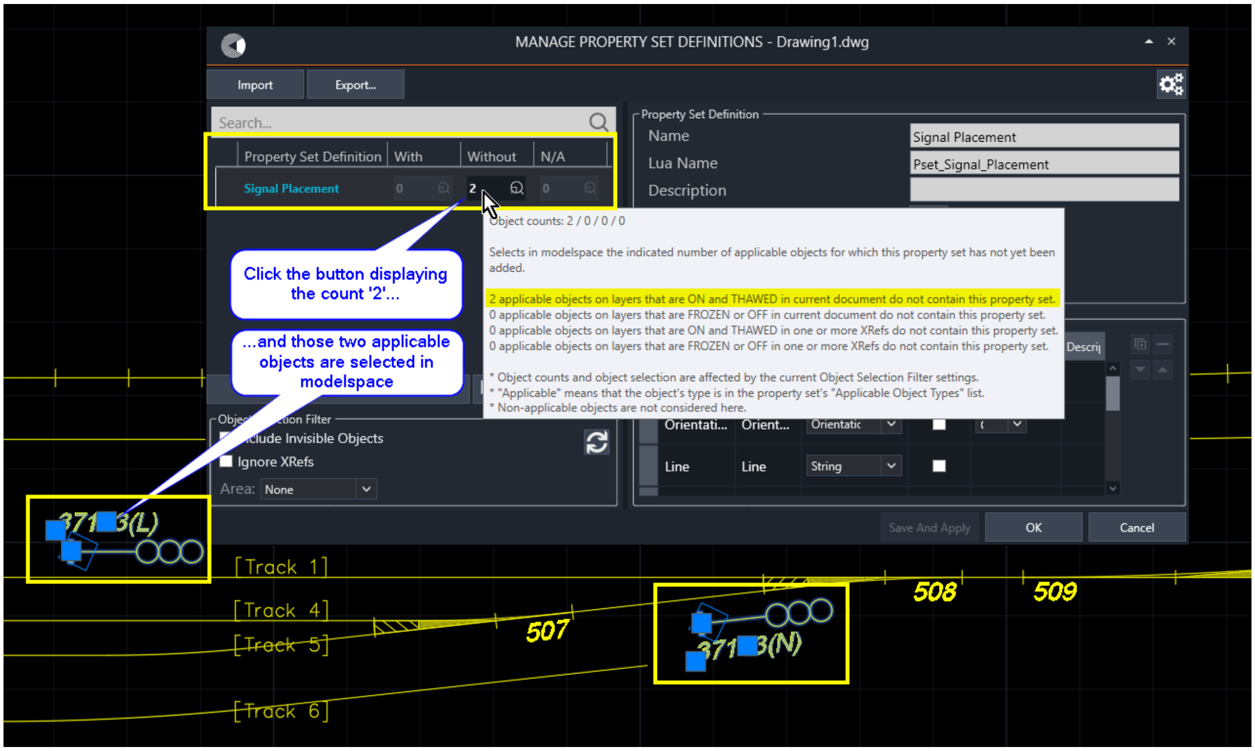
Name	Visible	Invisible
JBTSA_SIG Brusignal/Frostportsignal	0 / 0	0 / 0
JBTSA_SIG ERTMS Shunting signal	0 / 0	0 / 0
JBTSA_SIG Planovergangssignal	0 / 0	0 / 0
JBTSA_SIG Rasvarslingssignal	0 / 0	0 / 0
JBTSA_SIG Signal	0 / 2	0 / 0
JBTSA_SIG Togsporsignal	0 / 0	0 / 0

Figure 3 **Applicability editor**

To provide oversight, the numbers ‘a / b’ denote the current count of objects (of a given RC object type) that have been assigned the property set, versus the total number of objects of that type in the drawing. Similar counts are provided for objects in XRef’ed drawings. There are two “JBTSA_SIG Signal” objects that have not yet been assigned the “Signal Placement” property set in the above example.

Assigning Custom Property Sets to RC objects

Once the property set definition has been saved and applied to the drawing, you may conveniently use the Property Set Definition Manager’s browser pane to select objects that shall be assigned a property set. See below - click the the button displaying the count ‘2’ in the browser’s column “Without”:



Property Set Definition
Name: Signal Placement
Lua Name: Pset_Signal_Placement
Description:

Object counts: 2 / 0 / 0 / 0

Selects in modelspace the indicated number of applicable objects for which this property set has not yet been added.

2 applicable objects on layers that are ON and THAWED in current document do not contain this property set.
0 applicable objects on layers that are FROZEN or OFF in current document do not contain this property set.
0 applicable objects on layers that are ON and THAWED in one or more XRefs do not contain this property set.
0 applicable objects on layers that are FROZEN or OFF in one or more XRefs do not contain this property set.

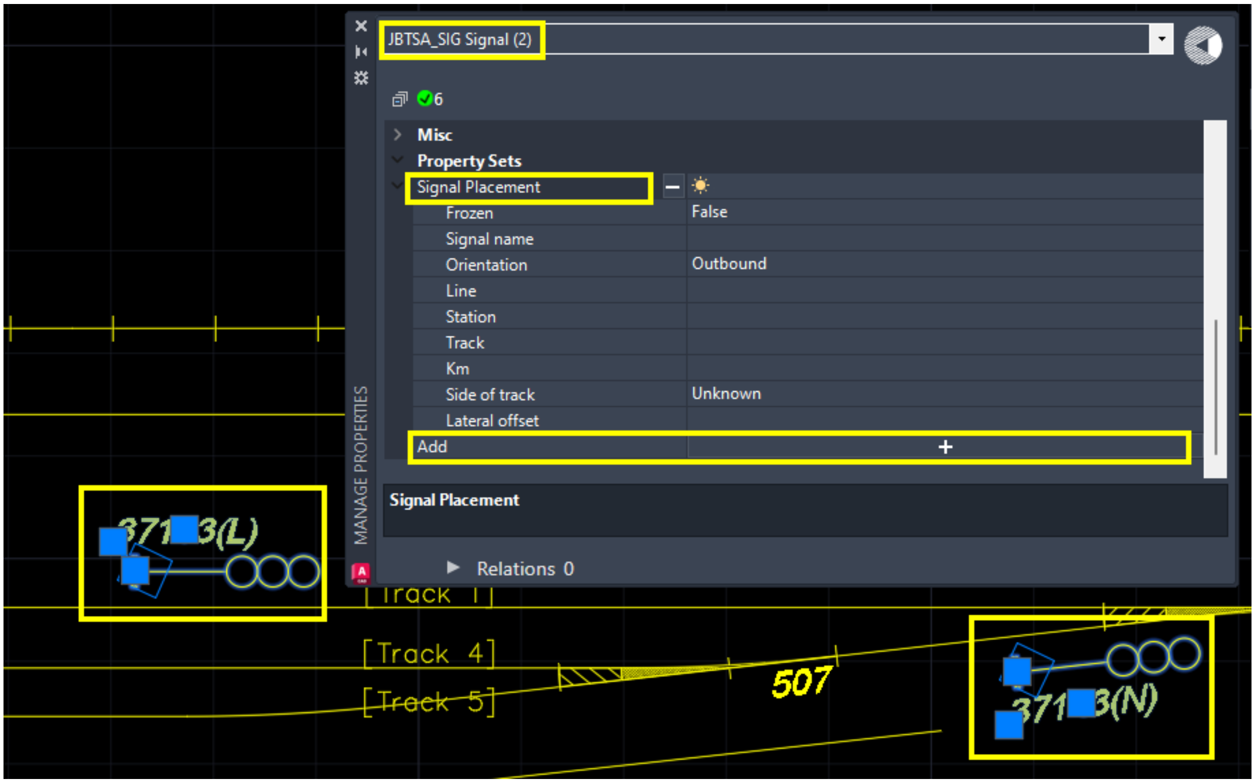
* Object counts and object selection are affected by the current Object Selection Filter settings.
* "Applicable" means that the object's type is in the property set's "Applicable Object Types" list.
* Non-applicable objects are not considered here.

Click the button displaying the count '2'...

...and those two applicable objects are selected in modelspace

Figure 4 Select applicable objects

Next, use the familiar RC Property Manager to add one ore more property sets to the selected objects – Click the 'Add' (+) button and select a property set from the dropdown list that appears:



Signal Placement	—
Frozen	False
Signal name	
Orientation	Outbound
Line	
Station	
Track	
Km	
Side of track	Unknown
Lateral offset	
Add	+

Signal Placement

Figure 5 Assign a Pset to RC objects

Property Set Automation using Lua

Entering literally thousands of values into your object’s property set properties takes a lot of time and can be prone to errors. But RailCOMPLETE’s unrivalled Lua automation system allows you to easily enter Lua formulas instead of plain default values for each property in the property set definition. In our ‘Signal Placement’ example, let’s add this formula to property ‘Signal name’:

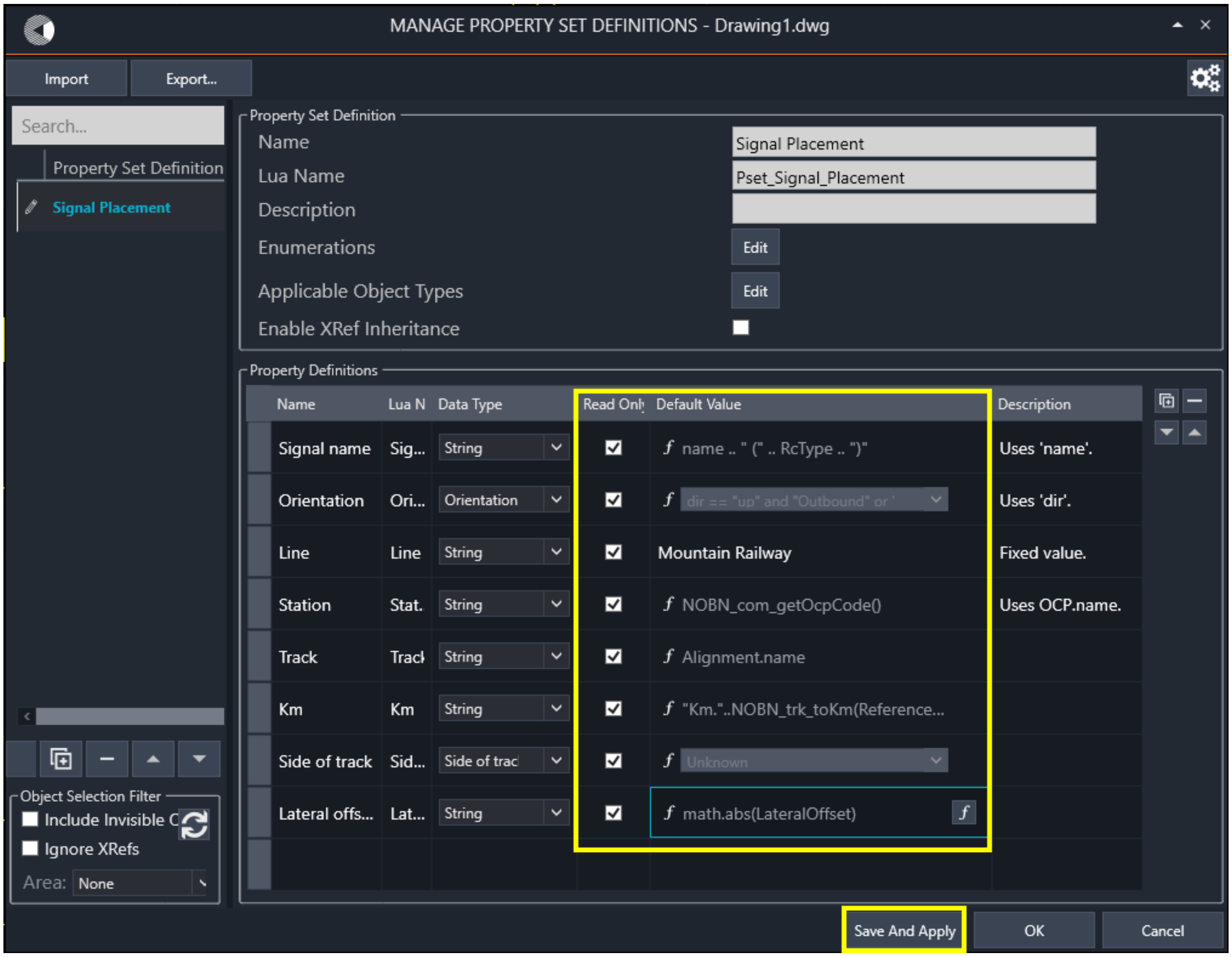
```
      name .. " (" .. RcType .. ")"
```

and this formula to the ‘Orientation’ property:

```
      dir == "up" and "Outbound" or "Inbound"
```

To distribute these automation formulas to the objects already featuring the property set, check the ‘Read Only’ box for the automated properties before you save and apply changes.

In the example below, we have entered automation formulas for all the properties (except the “Line” property, to which we have assigned a fixed value, since our example drawing is all about that one railway line). The user can now move RC objects around, change their names etc, and the property set values adapt automatically.



MANAGE PROPERTY SET DEFINITIONS - Drawing1.dwg

Property Set Definition

Name	Signal Placement
Lua Name	Pset_Signal_Placement
Description	
Enumerations	Edit
Applicable Object Types	Edit
Enable XRef Inheritance	□

Property Definitions

Name	Lua N	Data Type	Read Onl	Default Value	Description
Signal name	Sig...	String	☑	*ƒ* name .. " (" .. RcType .. ")"	Uses 'name'.
Orientation	Ori...	Orientation	☑	*ƒ* dir == "up" and "Outbound" or '	Uses 'dir'.
Line	Line	String	☑	Mountain Railway	Fixed value.
Station	Stat.	String	☑	*ƒ* NOBN_com_getOcpCode()	Uses OCP.name.
Track	Track	String	☑	*ƒ* Alignment.name	
Km	Km	String	☑	*ƒ* "Km."..NOBN_trk_toKm(Reference...	
Side of track	Sid...	Side of trac	☑	*ƒ* Unknown	
Lateral offs...	Lat...	String	☑	*ƒ* math.abs(LateralOffset)	

Object Selection Filter

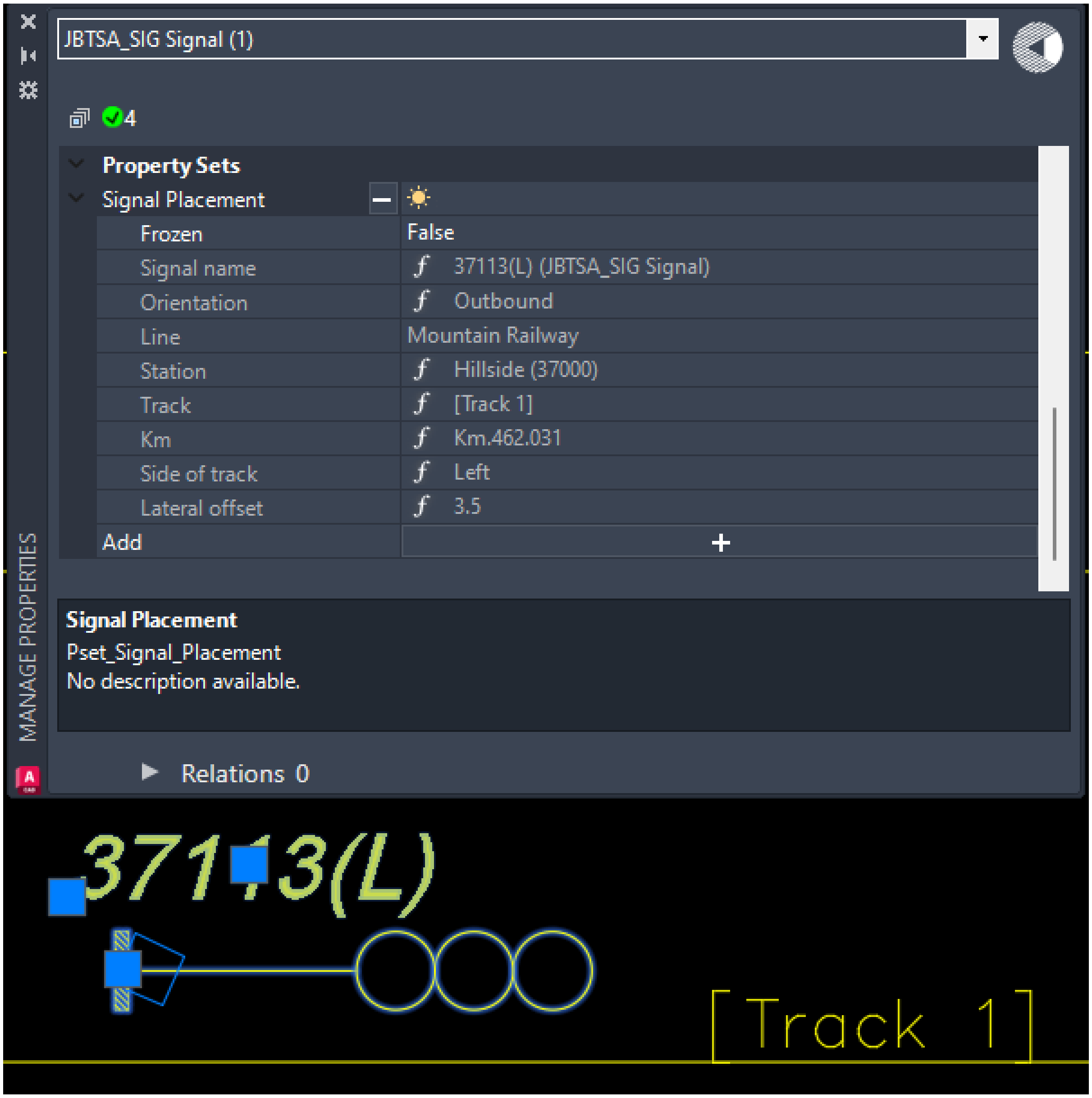
□ Include Invisible C

□ Ignore XRefs

Area: None

Figure 6 **Automation of property values with Lua code**

Here is what an updated object’s properties now look like:



JBTSA_SIG Signal (1)

4

Property Sets

Signal Placement

Frozen	False
Signal name	ƒ 37113(L) (JBTSA_SIG Signal)
Orientation	ƒ Outbound
Line	Mountain Railway
Station	ƒ Hillside (37000)
Track	ƒ [Track 1]
Km	ƒ Km.462.031
Side of track	ƒ Left
Lateral offset	ƒ 3.5
Add	+

Signal Placement
Pset_Signal_Placement
No description available.

▶ Relations 0

Figure 7 **Signal 37113(L) with automated values**

Exporting property set values to DWG or to IFC

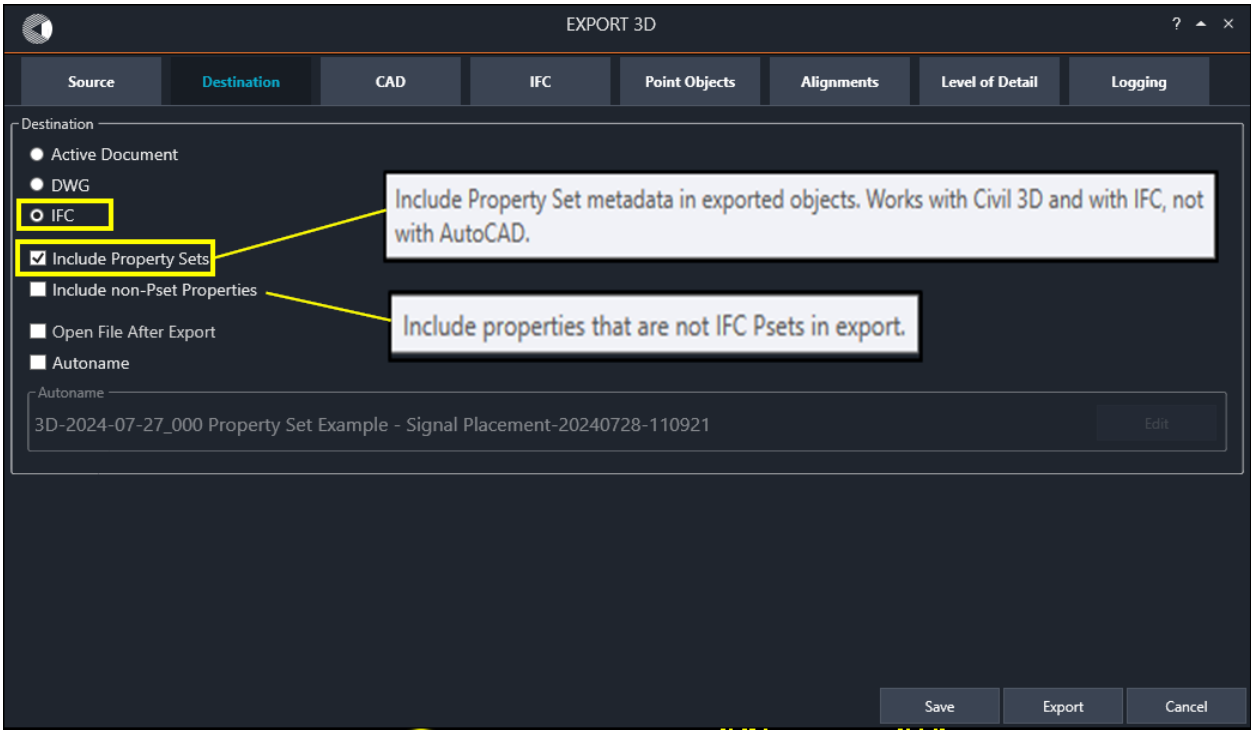
Exporting to IFC

RailCOMPLETE includes an IFC export engine which means that exports to IFC can be done while running RailCOMPLETE as a plugin to AutoCAD™, instead of having to run it under its ‘big brother’ Civil 3D™, which would be more expensive.

Check the option ‘IFC’ in the ‘Destination’ tab in the RC-Export3D command’s window and make sure that

You can choose between the following:

- No properties being exported
- Property sets (with their properties’ current values)
- RC objects’ regular intrinsic and custom properties (there are lots of them!)
- Both Property Set properties and regular properties



Destination

○ Active Document
○ DWG
● IFC

☑ Include Property Sets
☐ Include non-Pset Properties
☐ Open File After Export
☐ Autoname

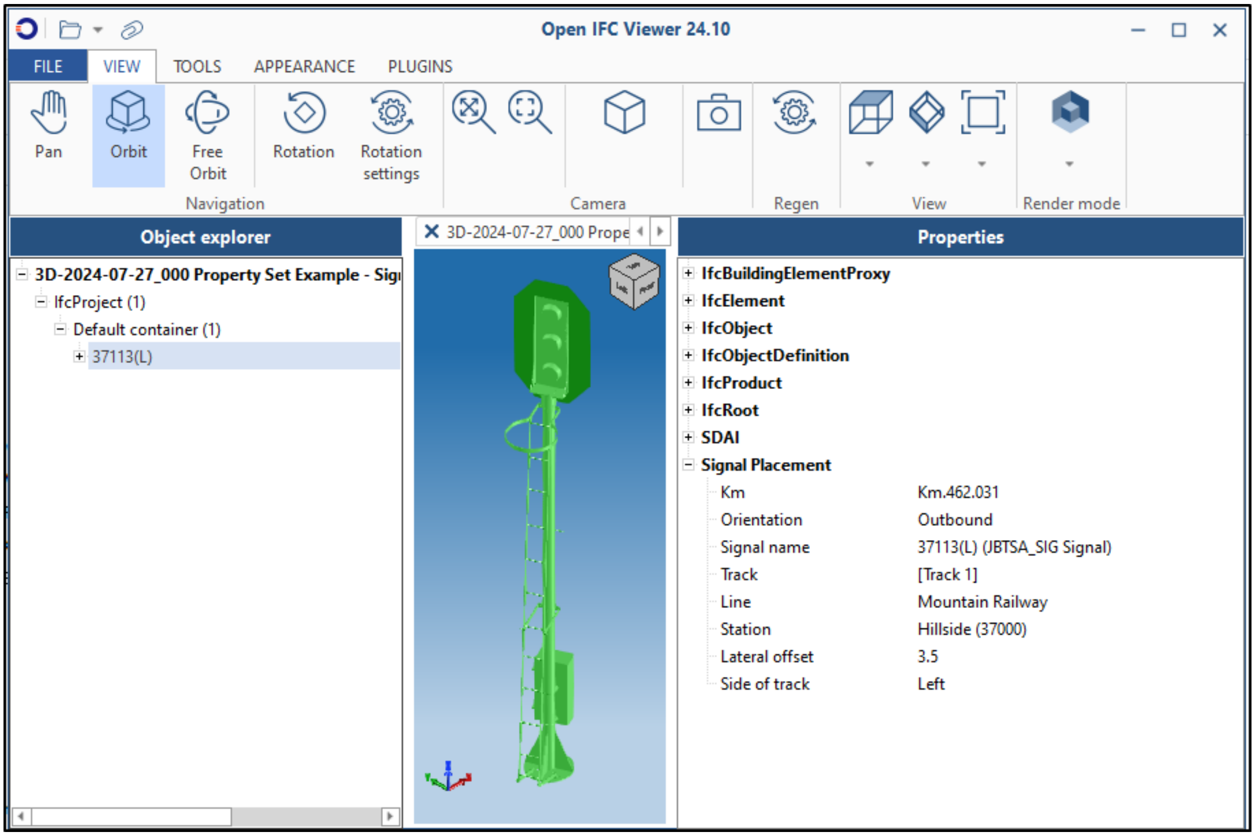
Autoname
3D-2024-07-27_000 Property Set Example - Signal Placement-20240728-110921

Include Property Set metadata in exported objects. Works with Civil 3D and with IFC, not with AutoCAD.

Include properties that are not IFC Psets in export.

Figure 8 Exporting RC objects' property set instances to IFC or to DWG along with their 3D geometry

This is what it might look like in IFC, here using the Open Design Alliance’s free viewer:



- 3D-2024-07-27_000 Property Set Example - Sig
  - IfcProject (1)
    - Default container (1)
      - 37113(L)

3D-2024-07-27_000 Prope

- IfcBuildingElementProxy
- IfcElement
- IfcObject
- IfcObjectDefinition
- IfcProduct
- IfcRoot
- SDAI
- Signal Placement

Km	Km.462.031
Orientation	Outbound
Signal name	37113(L) (JBTSA_SIG Signal)
Track	[Track 1]
Line	Mountain Railway
Station	Hillside (37000)
Lateral offset	3.5
Side of track	Left

Figure 9 Exported properties to IFC

Exporting to DWG

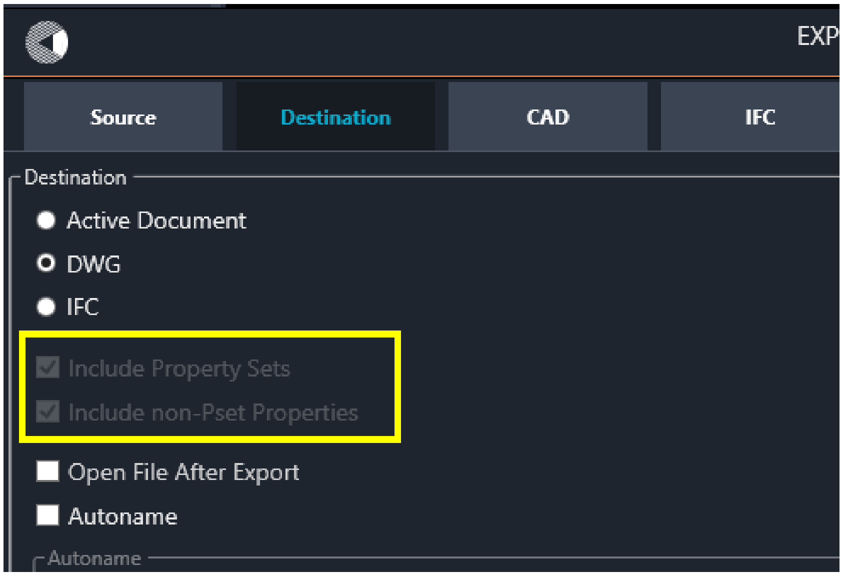
AutoCAD™ cannot attach property sets to CAD entities, but Civil 3D™ does. If you run RC as a plugin to Civil 3D, then an export of 3D geometry with properties to the DWG format will make them appear in DWG files loaded into Civil 3D or in the NavisWorks™ viewer.

Check the option ‘DWG’ in the ‘Destination’ tab in the RC-Export3D command’s window.

You can choose between the following:

- No properties being exported
- Property sets (with their properties’ current values)
- RC objects’ regular intrinsic and custom properties (there are lots of them!)
- Both Property Set properties and regular properties

Note that if you are running RailCOMPLETE under AutoCAD, then the two options ‘Include Property Sets’ and ‘Include non-Pset Properties’ are grayed down since they would not have any effect, anyway.



Destination

- ○ Active Document
- ● DWG
- ○ IFC
- ☑ Include Property Sets
- ☑ Include non-Pset Properties
- □ Open File After Export
- □ Autoname

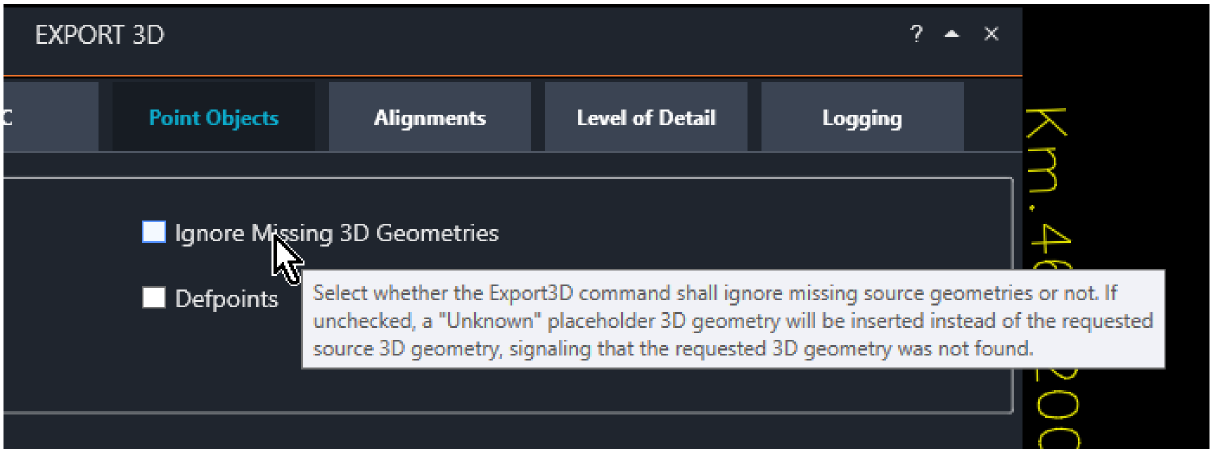
Autoname

Figure 10 **No properties will be exported to DWG when running RailCOMPLETE under AutoCAD**

Exporting properties to DWG or IFC from objects that do not have 3D geometry

There might be RC objects in your drawing that do not have an associated 3D geometry. Remember, 3D geometries can be added dynamically to any object, also to “abstract” object types.

To enable IFC or DWG viewers to display something in 3D that you can click on to represent such objects, you can either add representative graphics in RailCOMPLETE to those objects, or you can ensure that you have unchecked the “Ignore Missing 3D Geometries” in the RC-Export3D command window’s ‘Point Objects’ tab. This has the effect of inserting a default 3D geometry in the resulting DWG or IFC file for each such object, along with the properties that you have chosen to include.



- □ Ignore Missing 3D Geometries
- □ Defpoints

Select whether the Export3D command shall ignore missing source geometries or not. If unchecked, a "Unknown" placeholder 3D geometry will be inserted instead of the requested source 3D geometry, signaling that the requested 3D geometry was not found.

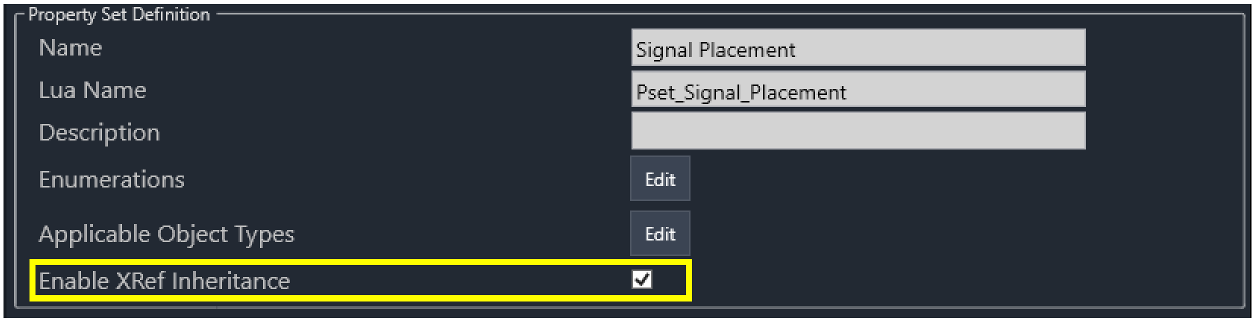
Figure 11 **Exporting properties to DWG or IFC for objects that do not have 3D geometry**

Most DNAs use a 3D question mark '?' symbol to represent an object without its own 3D geometry. Your DNA might not feature such a default 3D geometry file name, in which case you must explicitly add some dynamic 3D graphics to your abstract objects.

Advanced features with RailCOMPLETE property sets

XRef

Working with property set definitions being defined in an XRef'ed file is easy: Just check the "Enable XRef inheritance" box, save, and the property set definition becomes available to applicable objects in any DWG file that XRefs the file with the property set definition in it. Automation formulas apply as well.

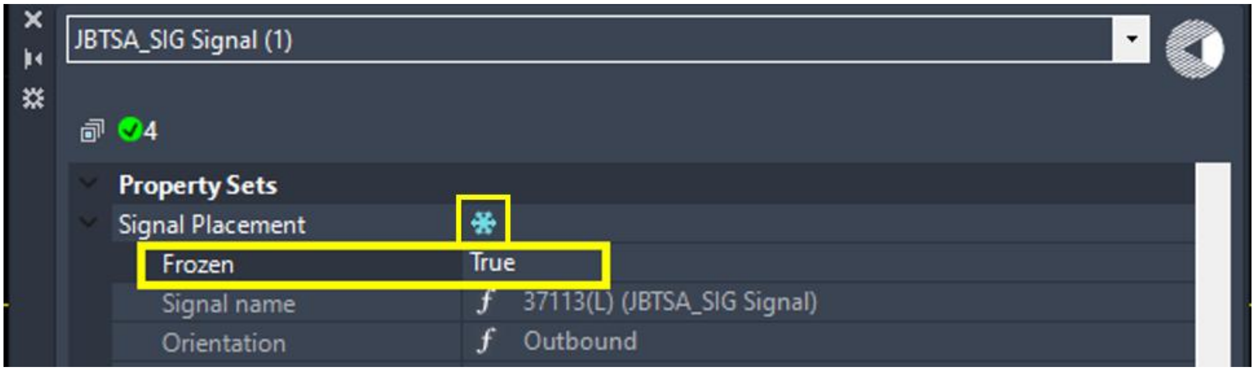

Property Set Definition

Name	Signal Placement
Lua Name	Pset_Signal_Placement
Description	
Enumerations	Edit
Applicable Object Types	Edit
Enable XRef Inheritance	☑

Figure 12 Use XRef inheritance

Frozen property set instance in RC object

In case you do not want to export every property set instance from your drawing to IFC, you may suppress all of them globally (in RC-Export3D), or you can set the "Frozen" attribute to True for the affected RC objects' property sets before exporting objects to IFC:


JBTSA_SIG Signal (1)

4

Property Sets

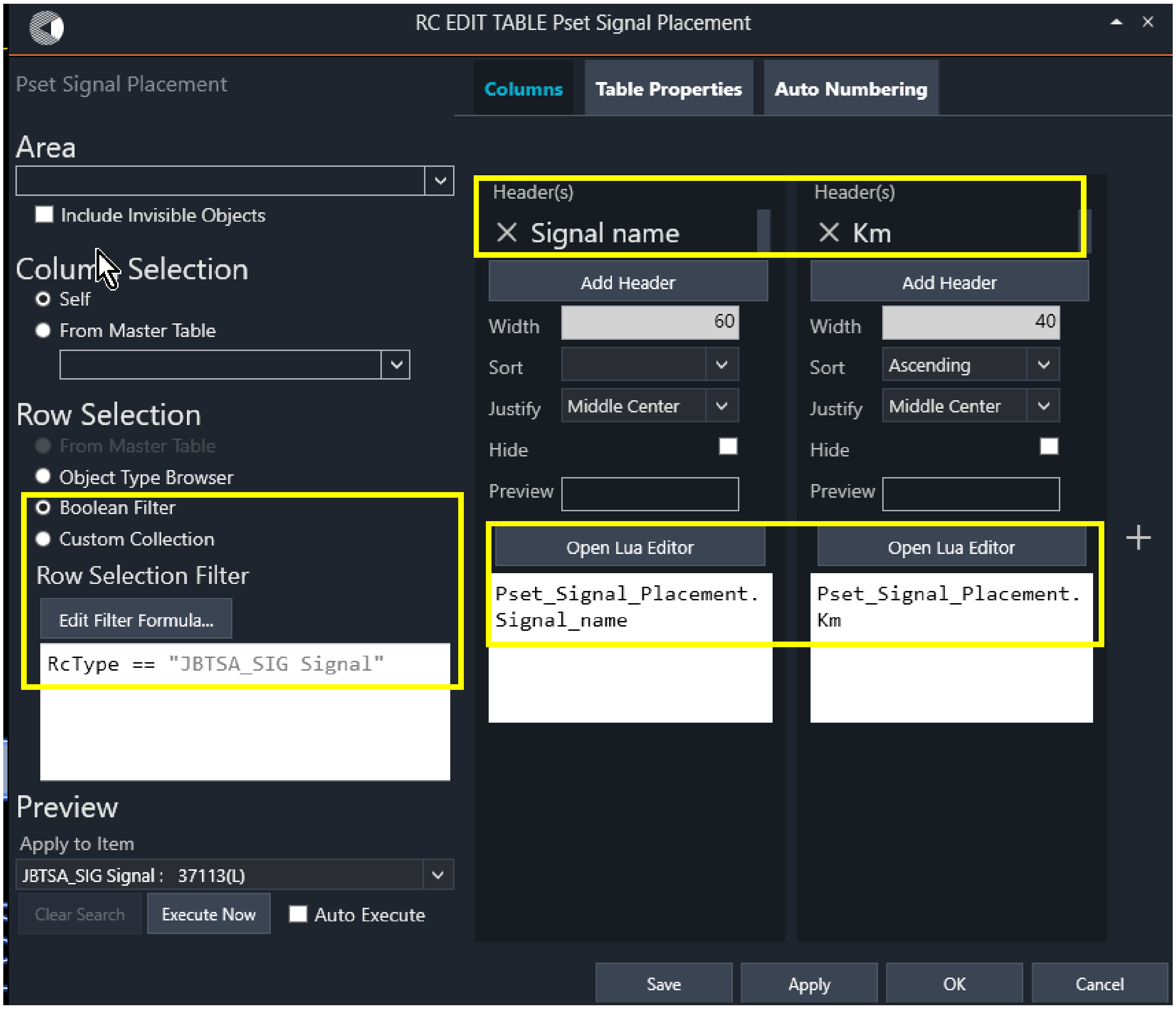
Signal Placement

Frozen	True
Signal name	37113(L) (JBTSA_SIG Signal)
Orientation	Outbound

Figure 13 Frozen property set - suppress output from RC object

Tables using Property Set values

Every property set and every property inside that property set must be given a unique Lua name. Scripts and tables in RailCOMPLETE can then access the drawing's database with a Lua program using those unique Lua identifiers. Here is an example from the RC table tool:



Pset Signal Placement

Area

□ Include Invisible Objects

Column Selection

● Self

○ From Master Table

Row Selection

○ From Master Table

○ Object Type Browser

● Boolean Filter

○ Custom Collection

Row Selection Filter

```
RcType == "JBTSA_SIG Signal"
```

Header(s): Signal name

Width: 60

Sort:

Justify: Middle Center

Hide: □

Preview:

```
Pset_Signal_Placement.
Signal_name
```

Header(s): Km

Width: 40

Sort: Ascending

Justify: Middle Center

Hide: □

Preview:

```
Pset_Signal_Placement.
Km
```

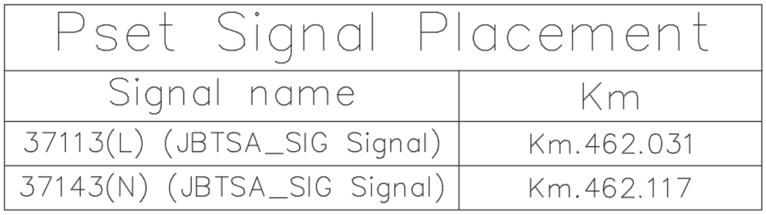
Preview

Apply to Item: JBTSA_SIG Signal : 37113(L)

□ Auto Execute

Figure 14 Table tool editor using Pset values

The resulting table is:



Pset Signal Placement	
Signal name	Km
37113(L) (JBTSA_SIG Signal)	Km.462.031
37143(N) (JBTSA_SIG Signal)	Km.462.117

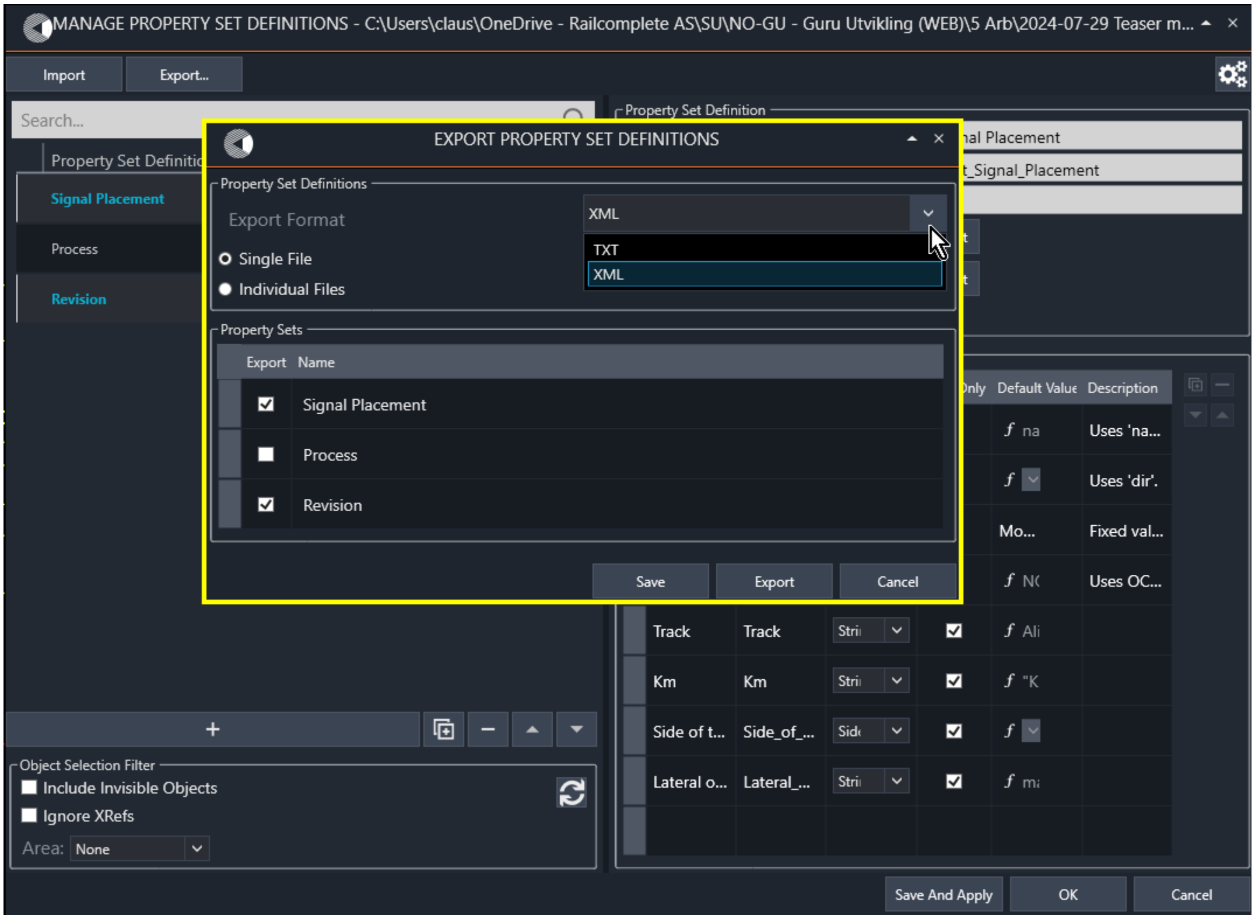
Figure 15 Table example displaying Pset values

Exporting and Importing RailCOMPLETE property set definitions

XML format

Property sets may be exported, one by one or several together, to a loss-less XML file format, proprietary to RailCOMPLETE. The lists of applicable RC object types are also stored, so you do not have to re-think the applicablility lists after importing those property set.

Upon export, pick from the list of available property set definitions which ones you want to export. Select either 'Single file' to bundle your selection into one XML file or select 'Individual files' to produce one XML file per property set definition. In the latter case, the file names are the same as the property set's names.


MANAGE PROPERTY SET DEFINITIONS - C:\Users\claus\OneDrive - Railcomplete AS\SU\NO-GU - Guru Utvikling (WEB)\5 Arb\2024-07-29 Teaser m...

EXPORT PROPERTY SET DEFINITIONS

Property Set Definitions

Export Format: XML (TXT / XML)

○ Single File
● Individual Files

Property Sets

Export	Name
☑	Signal Placement
□	Process
☑	Revision

Figure 16 Exporting Pset definitions to the lossless XML format

Upon import from an XML file, you can pick which property sets to import, in case you do not need them all.

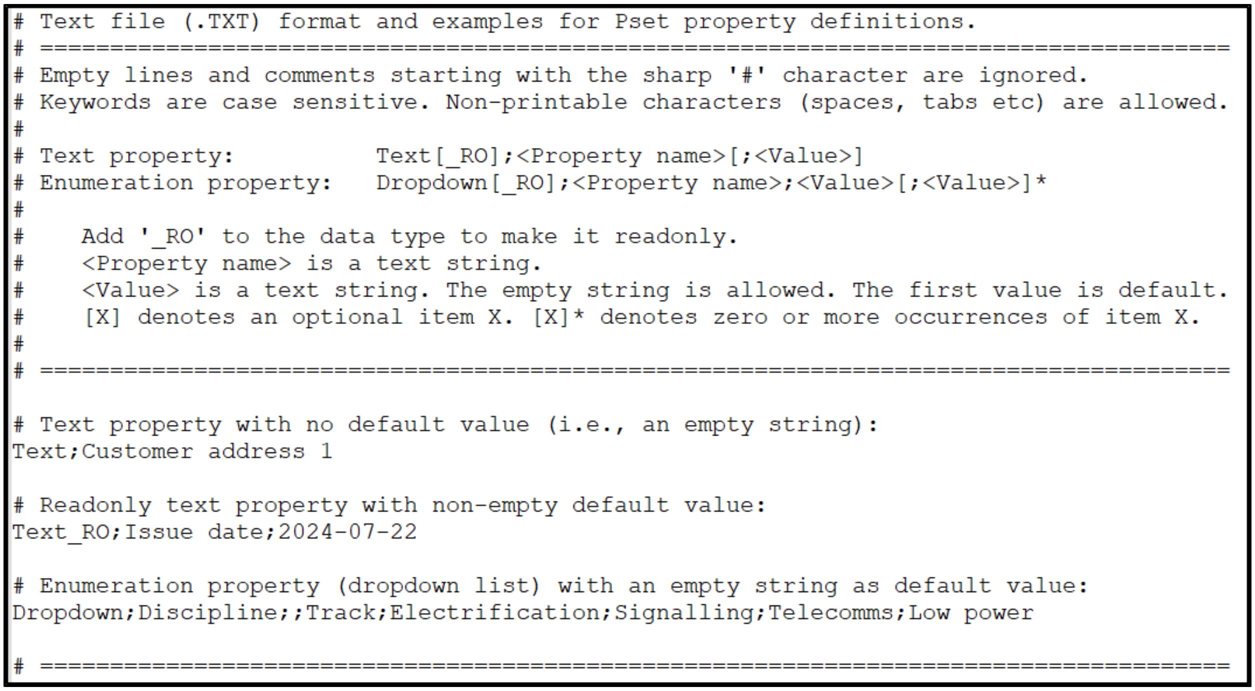
Need to share your property set definitions with your colleagues? Either store the property set definitions as a DWG file (remember to check "Enable Xref Inheritance"), or store as XML – then distribute that file, or its location, to everyone concerned.

TXT format

Property set definitions may be exported to a simple TXT file format, one file per property set definition. The property set's name is the same as the file name upon export/import. This simple format is also in use by other software products than RailCOMPLETE.

The properties' names, their data types (text or enumeration) and their default values are exported. Each property can be tagged as Read Only or not.

However, Lua automation code, descriptions and the list of applicable RC objects will not be exported to TXT files.



```
# Text file (.TXT) format and examples for Pset property definitions.
# =====================================================================================
# Empty lines and comments starting with the sharp '#' character are ignored.
# Keywords are case sensitive. Non-printable characters (spaces, tabs etc) are allowed.
#
# Text property:          Text[_RO];<Property name>[;<Value>]
# Enumeration property:   Dropdown[_RO];<Property name>;<Value>[;<Value>]*
#
#    Add '_RO' to the data type to make it readonly.
#    <Property name> is a text string.
#    <Value> is a text string. The empty string is allowed. The first value is default.
#    [X] denotes an optional item X. [X]* denotes zero or more occurrences of item X.
#
# =====================================================================================

# Text property with no default value (i.e., an empty string):
Text;Customer address 1

# Readonly text property with non-empty default value:
Text_RO;Issue date;2024-07-22

# Enumeration property (dropdown list) with an empty string as default value:
Dropdown;Discipline;;Track;Electrification;Signalling;Telecomms;Low power

# =====================================================================================
```

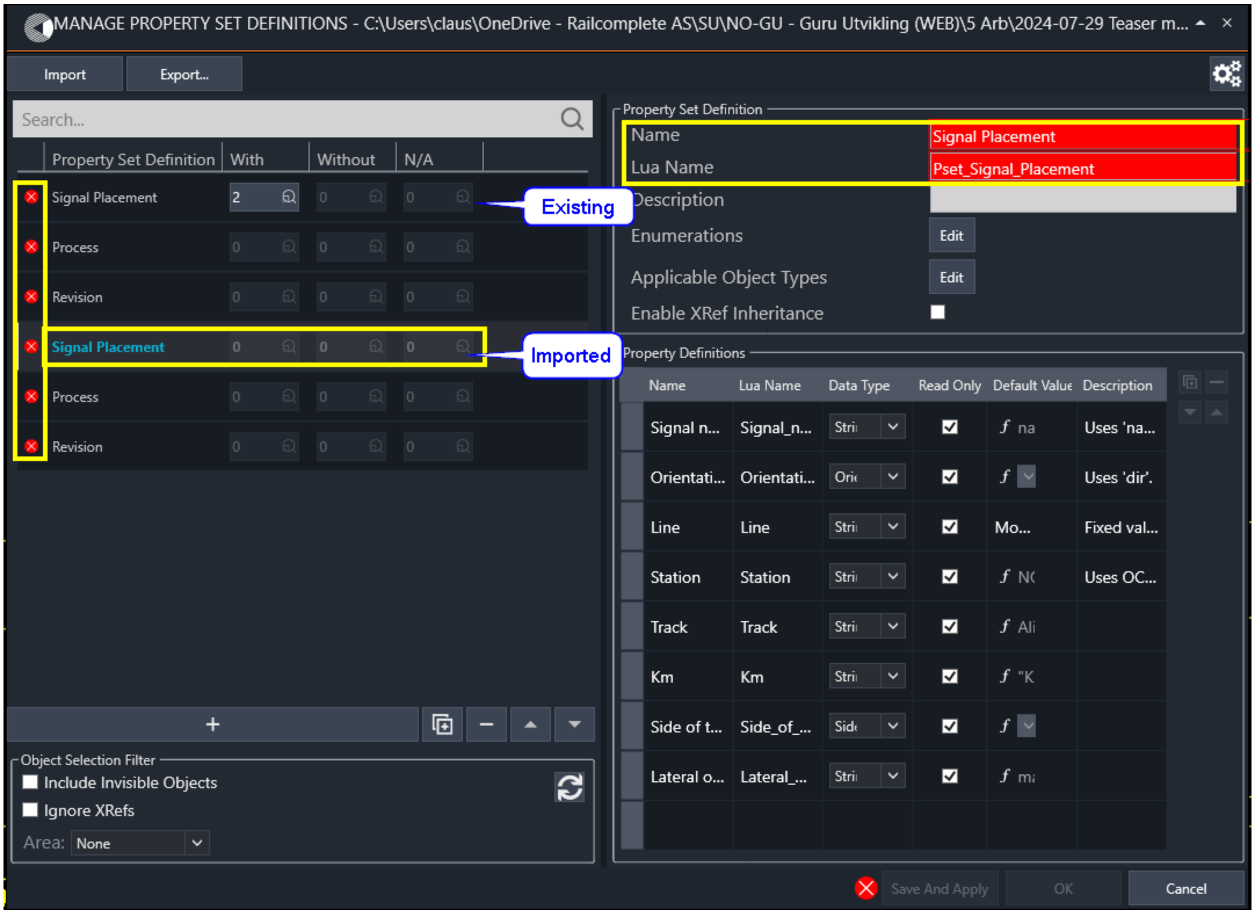
Figure 17 TXT format for Pset defintions

Updating existing property set definitions and their affected objects

Updating an existing property set definition is a functionality that may become available in future RailCOMPLETE versions.

Meanwhile, a little manual work is needed.

If you import a property set definition when there is already an existing definition occupying the same name, then both will be flagged in red as errouneous. The values that are already assigned to the existing property set instances in your RC objects will not be affected.



MANAGE PROPERTY SET DEFINITIONS - C:\Users\claus\OneDrive - Railcomplete AS\SU\NO-GU - Guru Utvikling (WEB)\5 Arb\2024-07-29 Teaser m...

Property Set Definition	With	Without	N/A
Signal Placement	2	0	0
Process	0	0	0
Revision	0	0	0
Signal Placement	0	0	0
Process	0	0	0
Revision	0	0	0

Property Set Definition

Name	Signal Placement
Lua Name	Pset_Signal_Placement
Description	
Enumerations	Edit
Applicable Object Types	Edit
Enable XRef Inheritance	□

Property Definitions

Name	Lua Name	Data Type	Read Only	Default Value	Description
Signal n...	Signal_n...	Stri	☑	ƒ na	Uses 'na...
Orientati...	Orientati...	Orie	☑	ƒ	Uses 'dir'.
Line	Line	Stri	☑	Mo...	Fixed val...
Station	Station	Stri	☑	ƒ NC	Uses OC...
Track	Track	Stri	☑	ƒ Ali	
Km	Km	Stri	☑	ƒ "K	
Side of t...	Side_of_...	Sid	☑	ƒ	
Lateral o...	Lateral_...	Stri	☑	ƒ m	

Object Selection Filter

□ Include Invisible Objects

□ Ignore XRefs

Area: None

Figure 18 Reimporting already existing Pset definitions

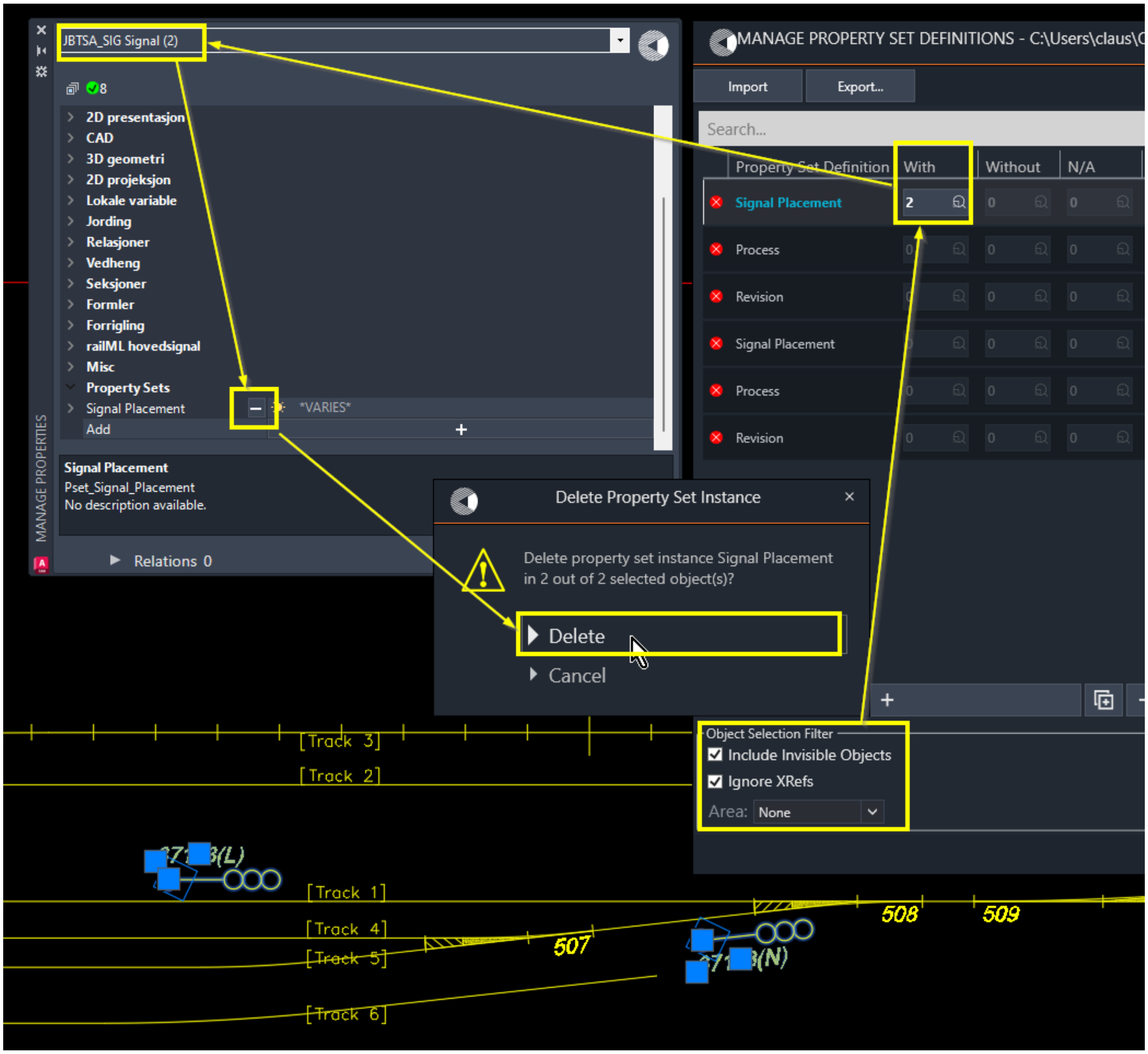
To proceed from here, you would need to inspect the two clashing property set definitions and carefully copy the new property names, enumerations, default values and automation formulas to update the old definition. After copying the contents, delete the newly imported definition. By checking the “Read Only” checkboxes and clicking “Save and Apply”, your existing RC objects’ properties will be updated according to their revised automation formulas. Properties that are not tagged as Read Only will not lead to changes in the RC objects’ corresponding existing property values.

Note that all flagged errors must be resolved before you can save and apply the revised property set definitions to your drawing. Delete the unused property set definitions that you do not need any longer, or change their property set names.

Once you have resolved all errors and clicked the Save and Apply button, the revised property set definitions will be applied to all existing RC objects referencing the property set definition, provided that the concerned properties have been tagged as Read Only in the property set definition.

Alternative solution for updating RC objects’ property sets

Alternatively, you may entirely scrap the data in your existing RC objects’ property set instances. Use the Property Set Definition Manager’s browser to select all visible and invisible objects in your current document, then use RC’s Property Manager to delete the property set instances – press the minus ‘-‘ symbol next to the property set’s name in the “Property Set” category.



Signal Placement
Pset_Signal_Placement
No description available.

Property Set Definition	With	Without	N/A
Signal Placement	2	0	0
Process	0	0	0
Revision	0	0	0
Signal Placement	0	0	0
Process	0	0	0
Revision	0	0	0

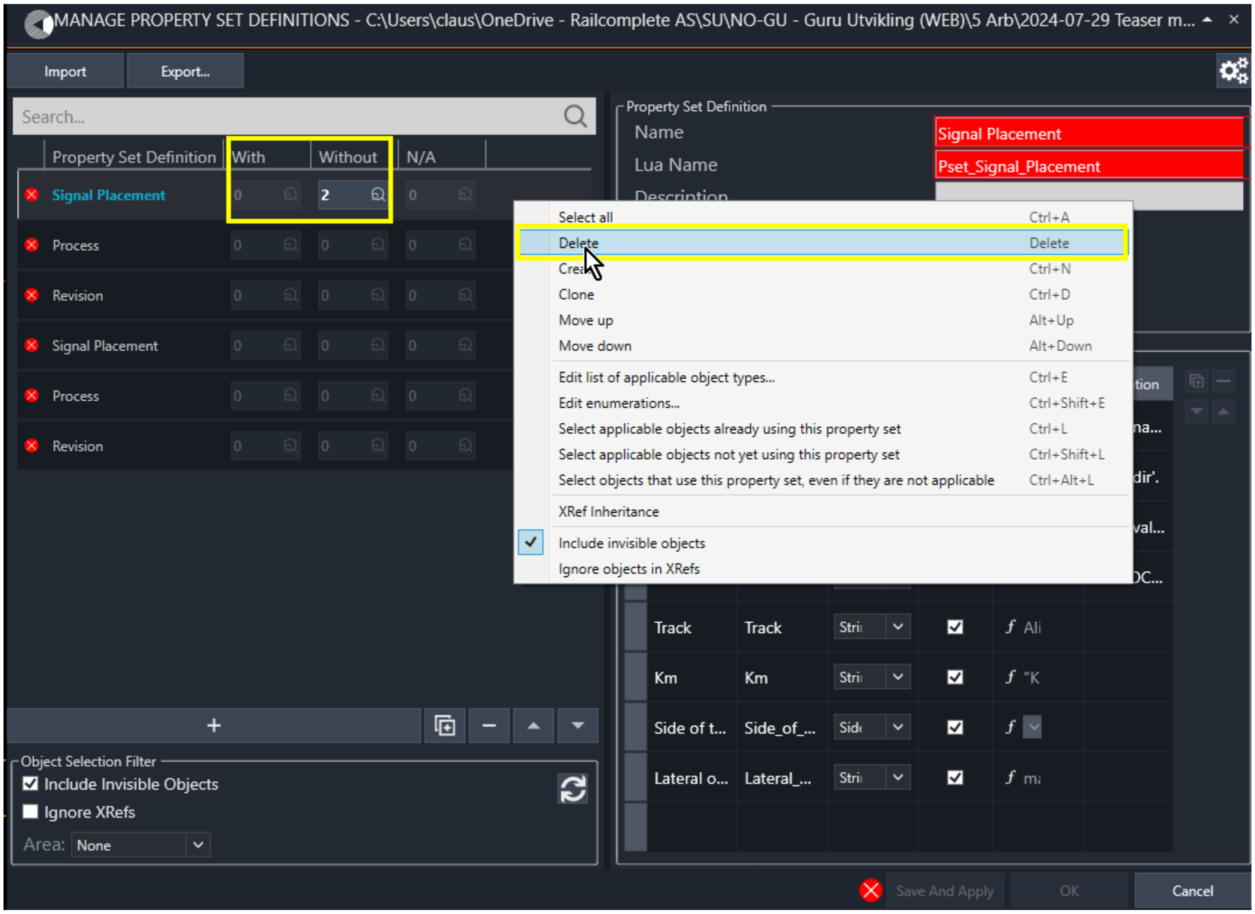
Delete Property Set Instance

Delete property set instance Signal Placement in 2 out of 2 selected object(s)?

Object Selection Filter
☑ Include Invisible Objects
☑ Ignore XRefs
Area: None

Figure 19 Deleting instances of a property set from RC objects

After successfully removing all existing instances of the old property set, you may proceed to delete its definition.



MANAGE PROPERTY SET DEFINITIONS - C:\Users\claus\OneDrive - Railcomplete AS\SU\NO-GU - Guru Utvikling (WEB)\5 Arb\2024-07-29 Teaser m...

Property Set Definition	With	Without	N/A
Signal Placement	0	2	0
Process	0	0	0
Revision	0	0	0
Signal Placement	0	0	0
Process	0	0	0
Revision	0	0	0

Property Set Definition

Name: Signal Placement

Lua Name: Pset_Signal_Placement

Description

Track	Track	Stri	☑	ƒ Ali
Km	Km	Stri	☑	ƒ "K
Side of t...	Side_of_...	Sid	☑	ƒ
Lateral o...	Lateral_...	Stri	☑	ƒ ma

Object Selection Filter

☑ Include Invisible Objects

□ Ignore XRefs

Area: None

Import
Export...
Search...
Select all Ctrl+A
Delete Delete
Create Ctrl+N
Clone Ctrl+D
Move up Alt+Up
Move down Alt+Down
Edit list of applicable object types... Ctrl+E
Edit enumerations... Ctrl+Shift+E
Select applicable objects already using this property set Ctrl+L
Select applicable objects not yet using this property set Ctrl+Shift+L
Select objects that use this property set, even if they are not applicable Ctrl+Alt+L
XRef Inheritance
✓ Include invisible objects
Ignore objects in XRefs
+
Save And Apply
OK
Cancel

Figure 20 Deleting an unused property set definition

After having deleted the old property set definition, the newly imported ‘Signal Placement’ property set definition from our example is no longer tagged as erroneous, and you can assign it to the applicable objects that used to have that property set. See further up in this article on how to add property sets to RC objects.

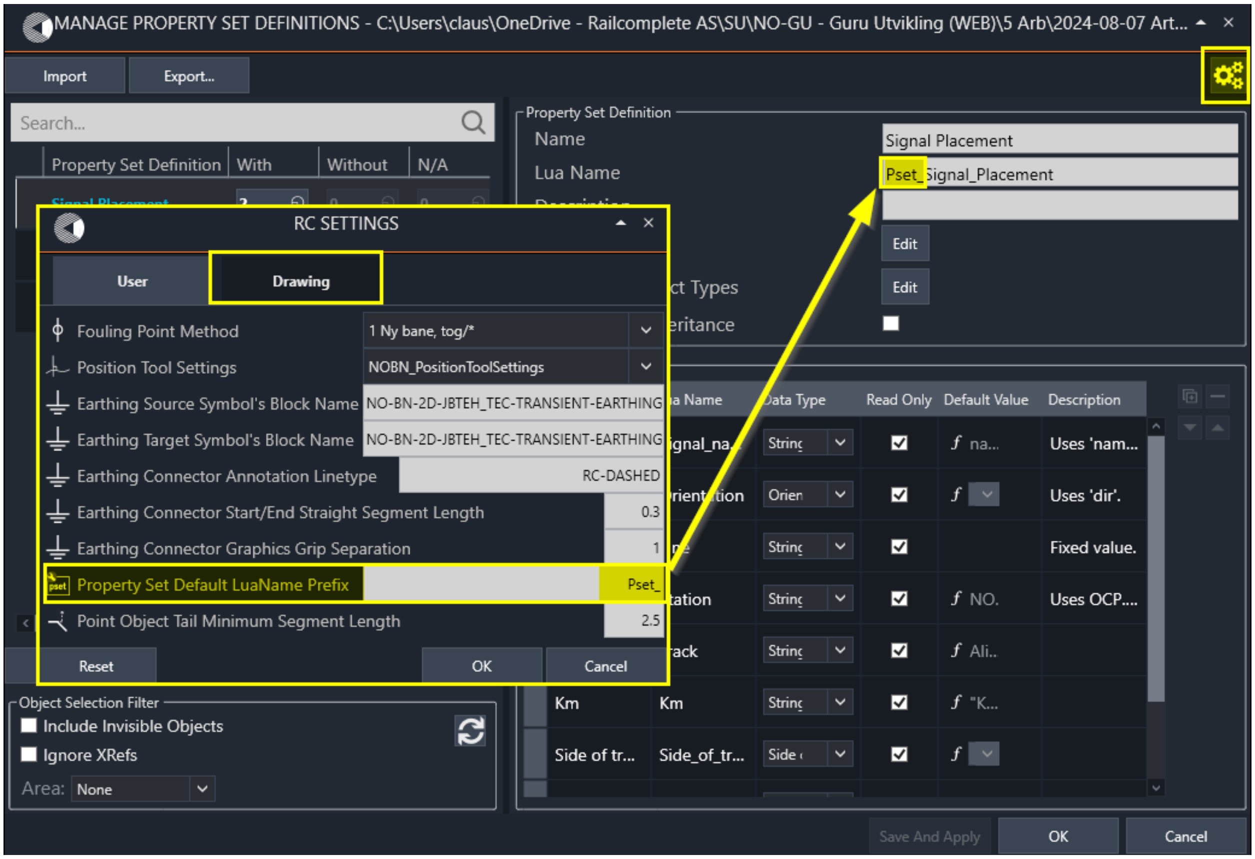
Lua Name prefix

Lua names are the identifiers used in Lua programming to access the content of a property set. Lua identifiers are not allowed to contain special characters such as commas and periods and they are not allowed to start with a digit. Nevertheless, your customer might require your property set to have a certain name such as “10.23: Installation”. The Property Set Definitions Manager will automatically convert the Name “10.23: Installation” into the suggested Lua Name “10_23__Installation”, where illegal characters have been replaced by the underscore ‘_’ character. However, that name will be flashed in red because it starts with a digit. You may then manually add an underscore or any other legal character as a prefix, to produce a valid Lua name: “_10_23__Installation”.

To ease this process, and to standardize across multiple users, your company might decide to require everyone to use a prefix such as “Pset_”. Such a prefix makes your property sets and their properties very recognizable in RailCOMPLETE’s Object Manager as well as in Lua formulas and scripts.

Click the Settings icon in the top right corner of the Property Set Definitions Manager window, or use the RC-Settings command and go to the tab for settings stored in your drawing (the current document). Enter a valid Lua identifier as the Property Set Default Lua Name Prefix.

The effect is that newly imported property set definitions, or the property set names that you enter manually, will be translated into Lua Names starting with the given prefix.



MANAGE PROPERTY SET DEFINITIONS - C:\Users\claus\OneDrive - Railcomplete AS\SU\NO-GU - Guru Utvikling (WEB)\5 Arb\2024-08-07 Art...

Property Set Definition

Name: Signal Placement

Lua Name: Pset_Signal_Placement

RC SETTINGS — Drawing

Fouling Point Method	1 Ny bane, tog/*
Position Tool Settings	NOBN_PositionToolSettings
Earthing Source Symbol's Block Name	NO-BN-2D-JBTEH_TEC-TRANSIENT-EARTHING
Earthing Target Symbol's Block Name	NO-BN-2D-JBTEH_TEC-TRANSIENT-EARTHING
Earthing Connector Annotation Linetype	RC-DASHED
Earthing Connector Start/End Straight Segment Length	0.3
Earthing Connector Graphics Grip Separation	1
Property Set Default LuaName Prefix	Pset_
Point Object Tail Minimum Segment Length	2.5

	a Name	Data Type	Read Only	Default Value	Description
	ignal_na...	String	☑	*f* na...	Uses 'nam...
	rientation	Orien	☑	*f*	Uses 'dir'.
	ne	String	☑		Fixed value.
	tation	String	☑	*f* NO.	Uses OCP....
	ack	String	☑	*f* Ali..	
Km	Km	String	☑	*f* "K...	
Side of tr...	Side_of_tr...	Side (	☑	*f*	

Object Selection Filter

□ Include Invisible Objects

□ Ignore XRefs

Area: None

Figure 21 Using RC-Settings to define a standard Lua Name prefix