PDT Norge

PDT Guide

Data templates and data dictionaries

– An example guide for traffic sign assembly information modelling and exchange

Outline of document:

Outline of document: 2
Introduction and use case description 3
Introduction and scope of document 3
Use case description 3
Part 1 – Structuring the information 4
Information need and specification of deliveries 4
Approaches for structuring the information 5
Identification of pieces of information 5
Specifying a context for each property 6
Defining the information delivery 7
Defining parts of the delivery 8
Making the information machine-readable 8
Example of machine-readable properties 10
Part 2 – Exchanging templates 11
Structuring the template payload 11
Overall structure 11
Subjects and subject relationships 11
Properties and property relationships 13
Other entities 15
RelationshipType 15
Quantity kinds 15
Units 16
External documents 17
The traffic sign assembly template bundle 19
Top-level traffic sign assembly template 20
Direction sign template 22
Traffic light template 23
Part 3 – Exchange of data/values 24

Introduction and use case description

Introduction and scope of document

The purpose of this document is to provide an introduction to how information about something can be identified, structured and modelled in a data dictionary and template.

The approach is practical and example based. The document has three parts:

- Part 1 – Structuring the information
- Part 2 – Exchanging templates
- Part 3 – Exchanging data

The first part is itended for everyone that deals with information modelling.

The second part is intended for developers (or others) that need to interact with a template communicated as a json file.

The third part is intended for developers (or other) that need to provide data according to a template and exchange these data as json.

Use case description

A road junction with traffic lights typically has gantries holding assemblies of traffic lights and direction signs.

The use case is the gantry seen below with its four traffic lights and four direction signs controlling the three lanes of traffic entering the junction.

The purpose of the example is to demonstrate how data for the example can be exchanged employing the approach of ISO 23387:2025 using data dictionaries according to ISO 12006-3:2022.

Part 1 – Structuring the information

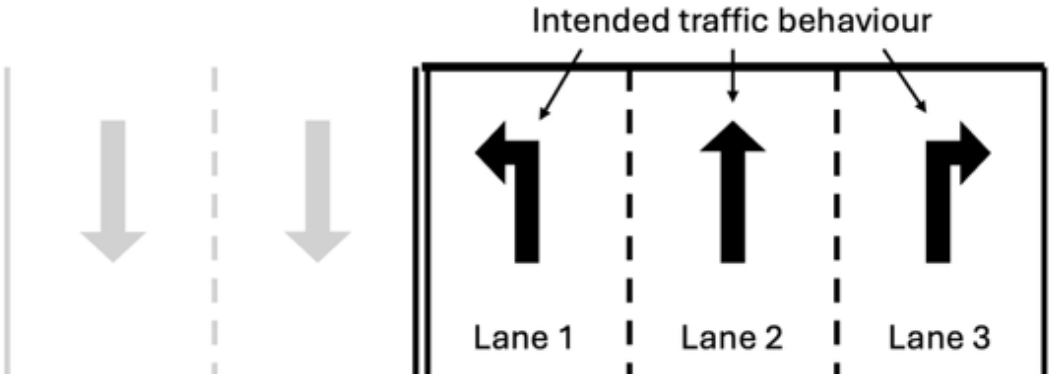
Information need and specification of deliveries

What we need to know about the complete assembly is the combined information about the junction itself and its four traffic lights and four direction signs.

All four traffic lights are standard traffic lights, and all four direction signs are standard direction signs, but each light or sign communicate different information.

Each traffic light shows a coloured light telling drivers of one lane of traffic to stop or go and each direction sign indicates which lane to use if you want to go to a certain destination.

There are three lanes entering the junction, and two (in grey) leaving it. Each lane entering the junction has a number and an intended traffic behaviour.



Approaches for structuring the information

Where multiple physical objects provide the same kind of information, such as direction signs, all direction signs follow the same *direction sign template*. They all have a height and width, a colour and a name or text describing to where they direct the traffic.

The height, width, colour, name etc. are referred to as *properties* representing a characteristic or a trait of the direction sign.

Each *property* is associated with a value describing a certain characteristic of the object the property belongs to. The value may be text, numbers or other information.

In the example, direction sign 1, 2 and 3 all share the same yellow background colour and the same height, but they have different text and/or symbols printed on them. Direction sign 4 differs from the others in colour and height.

The height in millimetres, yellow or white colour etc. are the *values* of the populated (data) template characterising each direction sign. The values are collected in a data *sheet*.

The direction signs are static over time, but the traffic lights provide dynamic information. The same traffic light may show both read, yellow and green light, but only one light at a time or in predefined combinations. This distinction will not be handled in this document.

Identification of pieces of information

Which pieces of information – properties – that are needed to fulfil the information requirement requires knowledge about the objects (junction, traffic lights, direction signs) and what the information shall be used for.

Each property represents a characteristic of an object. In order to understand which characteristic, each property needs a description stating what the property *is* – a property definition.

If the definition is short, it may be the name of the property, but for long or more abstract definitions, a name (term) is given in addition.

Note that the definition is what makes the property unique, not its name.

Example for the property “colour”:

Name	colour								
Definition	specification of the perceived visual appearance of the light reflected, emitted, or transmitted by an object, expressed through a predefined set of colour values								
Values	white	black	grey	red	orange	yellow	green	blue	violet

The definition of the property tells us that the colour is not expressed as a colour code, spectral wavelength or similar, but in the sense of common colour values such as red, green, blue etc.

Specifying a context for each property

The characteristic colour may be used for both direction signs and traffic lights, and for all four instances of each in the example. In order to know what the value “yellow” represents – the colour of direction sign 2 or colour of traffic light 3 – a context has to be specified for the property.

Such context specification is solved by including the property “colour” in collections for *both* direction signs and traffic lights. Secondly, the collections must allow for different *values* across the different direction sign collections as the colour of direction sign 1, 2, 3 and 4 may be different.

Another kind of context specification is the limitation of possible values based on the context. In general, the property colour may have many possible, predefined values such as black, white, grey, red, orange, yellow, green, blue, violet, etc.

Depending on the context – direction sign or traffic light – a subset of possible values may be used:

- Direction signs use yellow, white and blue in Norway (blue is not shown in the example).
- Traffic lights use red, yellow and green.

The *possible* values for the property colour is the “long” list of colour values, and the *permitted* values in a certain context are the shorter list.

Context	**Possible colour values**								
Generic	white	black	grey	red	orange	yellow	green	blue	violet
Traffic light				red		yellow	green		
Direction sign	white					yellow		blue	

Such context dependence is of interest in many use cases, and different approaches are possible:

1. Multiple properties – each context create an individual property with a unique list of values for the context such as “colour [traffic lights]” and “colour [direction sign]”, being two independent properties in addition to the generic property “colour”.
2. Property relationships – each context creates individual properties as above, but relates “colour [traffic light]” or “colour [direction sign]” to the generic property “colour” to show that they are a specific kind of colour.
3. Collections and filters – the list of possible values is limited using a filter mechanism when the property “colour” is used in a certain context (included in a certain collection).

Defining the information delivery

The complete package of information about the assembly is the collection of the information for the junction itself, the four populated traffic light templates and the four populated direction sign templates.

This package can be modelled using different approaches. To facilitate the choice of approach, the following questions are asked:

	Question	Comment
1	Does the information describe a combination of things?	The word “things” can be physical objects such as in the example, but also situations where different pieces of information cover the same object such as a product with information specific to each item and (standalone) information about the product type such as EPD.
2	If yes, does the information describe both the combination and its parts?	Where combinations are described as one combination and the parts are not of interest, the information still cover “one thing”. An example is a composite wall where the wall as a combination of products is the interesting part, not the structural parts, cladding etc.
3	If yes, does any of the information rely on which part of the combination it describes?	This applies both where multiple parts are of the same kind such as in the example, and where the same information applies to both the combination and any of its parts. Another example of the latter is a product identifier applied to both a window and the glazing of the window.

A single template can be used when the answer is No to the third question.

If the answer to question 3 is Yes, the information needs to be put in a context. This can be done using collections or by introducing property relationships where the context is provided per property.

Using collections is the preferred choice as the context often needs to be applied to multiple pieces of information. Two main types of collections are available:

- Data templates
- Groups of Properties

A data template defines “what is needed in this situation” – which property values needed for an information delivery. Data templates can include both groups of properties and other data templates. Not all properties of a group of properties have to be included in the template.

A group of properties defines “all there is to say about this context” – which properties that belong to a domain, reference document, or similar. A group of properties can contain other groups of properties, but not data templates.

Data templates and groups of properties collect properties differently. The data template defines the information delivery and contains a list of unique properties with corresponding

values. Each value is fixed within the scope of the template and within any group of properties in the template the property belongs to.

When the same property describes multiple parts (answer Yes to question 3), bundled data templates have to be used to allow for different *values* for each part.

In the traffic junction example, we have a top-level data template for the assembly including a data template for traffic lights and a data template for direction signs.

Defining parts of the delivery

For the content within one template, two more questions should be answered:

	Question	Comment
5	Does it make sense to collect the information into subsets for specific use or context?	Where there are many pieces of information, it is useful to group the pieces into smaller collections. It is possible for one piece of information to be part of multiple such collections.
4	Does any piece of information rely on other pieces of information?	If one piece of information depends on information about its surroundings, the information has dependencies. Some dependencies are crucial to communicate, others more for information.

Question 4 asks for the needs for structuring the information within the templat(s). Any property can be included in multiple groups of properties within a data template. A rule of thumb is that if any property in a template is included in a group of properties, the rest should also be grouped.

Question 5 deals with how the properties are modelled. Dependencies are a specific kind of context and how they can be handled is dealt with later in the document.

Making the information machine-readable

This document is written in English. When a property, such as colour is used in information exchange, the recipient must recognise both the term itself and in which language it is given. In Norwegian, “farge” is the name (term) for the colour property, and the definition may also be translated.

Humans speaking both Norwegian and English will recognise “farge”, “colour” and “color” as the same property. A machine that receives “farge” or “color” will not recognise this as “colour” because the spelling is different.

Machine readability means that machines use language independent “labels” called unique identifiers to recognise properties or other pieces of information such as property values, metadata and context.

The unique identifier is a string of characters that may be human-readable information, but often, it is a system generated code referred to as a “globally unique identifier” (GUID) or a “universally unique identifier” (UUID).

Generally, every piece of information – or information concept – can be made machine-readable. The metadata of a concept is typically structured as:

Part	Metadata/attribute	Comment
Identification	Unique identifier	Unique code or identifier string
Concept definition	Name	Non-unique, language dependent label
	Definition	Unique language dependent expression
	Reference	Reference to a document or other source for the definition
Content specification	Different metadata/ attributes	Concept specific metadata/attributes: Properties: Specification of the value Template: Specification of which properties the template include. Documents: Author, publisher, ISBN and document information.

Assigning unique identifiers to the metadata in addition to the property itself provide different degrees of machine readability for properties. The table below provides an example.

Metadata/attribute	Low	Medium	High
Unique identifier	yes	yes	yes
Name	yes	yes	yes
Definition	–	possible	yes
External reference	–	–	if relevant

Example of machine-readable properties

Breaking down a property, we find the following metadata:

Part	Metadata/attribute	Comment
Conceptual property definition	Name	Non-unique, language dependent label
	Definition	Unique language dependent expression
	Reference	Reference to a document or other source for the definition
Value specification	Possible values	List of enumerated/predefined values
	Boundary values	Numerical interval defining the range of allowed (numerical) values
	Value data type	The digital encoding of each value, e.g., string, whole or decimal number, date, true/false etc.
	Physical nature	The physical quantity (if any) the property represents, e.g., length, pressure, volume etc.
	Unit of measurement	The unit of measurement for numerical values, e.g., mm, m, MPa, bar, m^3, l.

Assigning unique identifiers to the metadata in addition to the property itself provide different degrees of machine readability for properties. The table below provides an example.

Metadata/attribute	Low	Medium	High
Unique identifier	yes	yes	yes
Name	yes	yes	yes
Definition	possibly	possibly	yes
External reference	–	–	if relevant
Possible values	–	if relevant	if relevant
Boundary values	–	–	if relevant
Value data type	–	yes	yes
Unit of measurement[1]	–	if relevant	if relevant
Physical dimension[1]	–	(implicit)[1]	yes
Physical quantity[1,2]	–	–	yes

[1] The physical dimension relates a quantity to its unit of measurement in terms of the exponents of the base units m, kg, s, A, K, mol and cd

[2] The physical quantity is a conceptual description of what the quantity (numerical value) represents such as different definitions of a length measure and a width measure

Part 2 – Exchanging templates

Structuring the template payload

Overall structure

The payload including the template and other content from the data dictionary follows the relational data base structure specified in ISO 12006-3:2022.

The payload for a requested entry contain the requested entry and all other data dictionary concepts referred to by that entry.

The different data dictionary concepts (entities) contained in the database are:

- Data templates
- Groups of properties
- Object types
- Properties
- Relationship types
- QuantityKinds
- Units
- External documents
- Data dictionaries
- Countries
- Subdivisions
- Languages

Subjects and subject relationships

Data templates, Groups of properties and Object types are subject kind entities. As described in Defining the information delivery, Data templates and Groups of properties behave differently.

The definition of subjects follow this structure:

Metadata/attribute	DT	GoP	OT	Description
UniqueId	X	X	X	UUID of the subject
Names (FullName, ShortName)	X	X	X	Non-unique, language dependent label. Mandatory FullName and optional ShortName
Definition	X	X	X	Unique language dependent expression
ReferenceDocument	(X)	(X)	(X)	Optional reference to an ExternalDocument dictionary UUID

IsKindOf	X	X	X	RelationshipToSubject with reference to a TargetSubject UUID and an IsKindOf RelationshipType UUID
IsSubkindOf	(X)	(X)	(X)	Optional RelationshipToSubject. Not in use.
IsSubtypeOf	(X)	(X)	(X)	Optional RelationshipToSubject. Not in use.
HasPart	(X)	(X)	(X)	Optional list of RelationshipToSubjects with reference to a TargetSubject UUID of the same kind and a HasPart RelationshipType UUID.
HasGroupOfProperties	(X)	–	–	Optional list of RelationshipToSubjects from a data template to a Group of properties TargetSubject UUID and a HasGroupOfProperties RelationshipType UUID.
HasObjectType	(X)	–	–	Optional list of RelationshipToSubjects from a data template to an Object type TargetSubject UUID and a HasObjectType RelationshipType UUID..
Properties	X	X	–	List of SpecifiedProperty objects with a specification of the included Property UUID, optional list of PermittedValues UUIDs, SpecifiedUnit UUID and an indication of whether the property is mandatory or not.
Filters	(X)	(X)	–	Optional list of filter objects specifying the same information as the specified property object. Redundant.

The RelationShipToSubject is structured as follows:

Metadata/attribute	**Description**
TargetSubjects	List of UUIDs of the target subject the calling subject is related to.
ScopeSubjects	Optional list of UUIDs to subjects providing additional context. Not in use.
RelationshipType	UUID of the RelationshipType defining the relationship.

The SpecifiedProperty is structured as follows:

Metadata/attribute	Description
Property	UUID of the property.
PermittedValues	Optional list of UUIDs for the values permitted within the subject. The values are found within each property.
ThresholdValues	Optional interval providing legal or contextual limitiations to numerical values.
SpecifiedUnit	Optional UUID for the specified unit if none or multiple units are specified for the property itself.
IsMandatory	Nullable boolean specification of whether the property is considered mandatory within the context. Possible values depend on the context: • DataTemplates: true or false • GroupsOfProperties: null • RelationshipToProperty: true

Properties and property relationships

The definition of properties follow this structure:

Metadata/attribute	Description
UniqueId	Mandatory UUID of the subject
ExternalIdentifier	Mapping table to external data dictionaries or semantic frameworks
Names (FullName, ShortName)	Non-unique, language dependent label. Mandatory FullName and optional ShortName
Definition	Mandatory unique language dependent expression
ReferenceDocument	Optional reference to an ExternalDocument dictionary UUID
IsSpecialisationOf	Optional relationship object with a TargetProperty and an IsSpecialisationOf RelationshipType identifier.
ProxyProperty	Optional list of relationship objects with a TargetProperty and an ReferenceProxyProperty or AlternativeProxyProperty RelationshipType identifier.
DataType	Enum value according to ISO 12006-3:2022 as XTD_...
ModelType	Designation of the model type for the property in terms of: • Simple: No dependency relationships • Output: Property with function dependencies

	• Proxy: Property with proxy property • Table: Property with context parameters • TableOutput/TableProxy: Property with context parameters and function dependencies/proxy properties
ValueType	Designation of the kind of value: string, boolean, integer, decimal, dateTime, date or enum.
ValueDataType	The data type in terms of xsd:<datatype>
DataFormat	String proving a regex pattern for string values, the precision of decimal values etc.
Dimension	Designation of the physical dimension of the quantity expressed in terms of the exponents of the seven base quantities length (m), mass (kg), time (s), electric current (A), thermodynamic temperature (K), amount of substance (mol) and luminosity (cd). The physical dimension of properties shall correspond to the physical dimension for both the quantity kind and the units.
QuantityKind	UUID of the QuantityKind object.
PossibleValues	Optional list of value objects. Each value object has a UniqueId, an ExternalIdentifier mapping, Names in different languages, and a NominalValue representing the numerical value, code value or any other language independent representation of the value.
BoundaryValues	Optional interval object with specification of the lower and upper boundaries for the property value.
Units	Optional list of relationship objects with reference to a TargetSubject UUID of the same kind and a HasPart RelationshipType UUID.
ValueCardinality	Optional list of unit UUIDs.
ContextParameters	Optional list of relationship objects with a TargetProperty and an ContextParameter or SelectedContextParameter RelationshipType identifier.
FunctionDependencies	Optional list of relationship objects with a TargetProperty and an FunctionDependency RelationshipType identifier.
ConnectedProperties	Optional list of other non-standardised relationships to properties.

The RelationShipToProperty is structured as follows:

Metadata/attribute	**Description**
TargetProperties	List of SpecifiedProperty objects (see under Subject and subject relationships) specifying the target properties.
RelationshipType	Identifier for the RelationshipType defining the relationship.

Other entities

Other entities are also included. Most of these are for “information only”, but a few key concepts are worth understanding.

RelationshipType

The RelationshipType provides a conceptual definition for relationships between subjects or between properties.

The different property relationships are defined as:

Relationship	Description
IsKindOf	Generic relationship from a dictionary level subject to a dictionary meta level subject providing the conceptual definition of the subject kind. Possible kinds: Data template, Group of properties, Object type.
IsSubkindOf	Generic relationship between dictionary meta level subjects to define different subject kinds and subkinds. Not in use.
IsSubtypeOf	Generic relationship to a dictionary level subject where the calling subject inherits the content of the target subject. Not in use for other than object types.
HasPart	Partitive relationship to other dictionary level subjects of the same kind.
HasGroupOfProperties	Partitive relationship from a data template to a group of properties.
HasObjectType	Associative relationship from a data template to an object type.
IsSpecialisationOf	Generic relationship to another property, where the value of the calling property shall be a valid value of the generic property.
ProxyProperty	Partitive relationship where the target property is used directly/as a proxy for the calling property. The value of the target property is a valid value of the calling property.
ContextParameter	Partitive relationship where the value of the calling property may change if the value of the target property change. Target properties shall be of enum type.
FunctionDependency	Partitive relationship where the value of the calling property is found from a series of operations including the value of the target property.

Quantity kinds

The quantity kind identifies how a quantity (numerical value) is determined. Is it a length quantity, pressure quantity, or similar.

Quantity kinds are given a name and definition in addition to a physical dimension used for validation with the property and unit dimensions.

Properties refer to a Quantity kind, and the name of the property and quantity kind may seem contrary because the name of the quantity refers to how something is quantified.

Examples:

Quantity	Definition	Comment
length	linear extent in space between any two points	Does not have to be a straight line. Applies to properties describing the length of ropes, along an arbitrary path etc., and may apply to dimensions of objects, distance between two points etc.
width, breadth	minimum length of a straight line segment between two parallel straight lines (in two dimensions) or planes (in three dimensions) that enclose a given geometric shape	A more specific kind of “length”. Applies to object dimensions regardless of the dimension is named length, width or height.
height, depth, altitude,	minimum length of a straight line segment between a point and a reference line or plane	Another specific kind of length. Applies to objects standing on the floor, installed in racks (depth from front plane to back of object) etc. The height of a building board is a width quantity.

Units

The unit defines the unit of measurement. The unit contains all necessary information to validate that it represents the quantity kind of the property, and to convert quantities between different units.

Units

Metadata/attribute	Description
UniqueId	Mandatory UUID of the subject
ExternalIdentifier	Mapping table to external data dictionaries or semantic frameworks. Example: Mapping to UNECE Rec 20 codes MTR, PAS, KGM
Names (FullName, ShortName)	Non-unique, language dependent label. Mandatory FullName and optional ShortName Example: metre, Pascal, kilogram
Definition	Mandatory unique language dependent expression
ReferenceDocument	Optional reference to an ExternalDocument dictionary UUID

Symbol	Optional language dependent string representing the unit symbol used in mathematical expressions Example: m, Pa, kg
Dimension	Designation of the physical dimension of the quantity expressed in terms of the exponents of the seven base quantities length (m), mass (kg), time (s), electric current (A), thermodynamic temperature (K), amount of substance (mol) and luminosity (cd).
Coefficient	Rational number specifying the conversion coefficient of a certain unit in terms of base SI units. Example: m = 1/1 m, mm = 1/1 000 m, g = 1 000/1 kg, N/mm2 = 1 000 000/1 Pa.
Offset	Rational number specifying the offset from a pure scaling. Relavant for certain units. Example: [°C] = 1/1 [K] – 27315/100.
UnitScale	Designation of whether the unit denotes a linear or logarithmic quantity.
UnitBase	Designation of the base of the unit. • Linear units: 1, π or 1/π. • Logarithmic: 10 (log10), 2 (log2) or *e* (ln)

External documents

External documents play an important role in distinguishing properties or subject from one another. As many properties ay have the same name, there is no strict naming convention.

The External document entity have an optional attribute DocumentReference. This contains a short identifier or reference to the document itself. Examples are:

Document title	**Document reference**
Construction product regulation (CPR)	3110/2024
Byggteknisk forskrift	TEK17
Classification of reaction to fire…	EN 13501-1:2018
Bestandig betong med alkalireaktivt tilslag	NB 21

Many properties have names “<name> acc. to <documentReference>”, enabling search for the standard reference and recognition of the “correct” property if the “according to…” is the important identifier for this. This is a simple approach but introduce some issues.

The recommendation is to keep any reference to documents outside the name of the property. The reason is partially because names become unnecessary long, mot most important, it introduce additional governance load and increase the risk of mismatches.

An example of the latter is where a property is defined according to a reference document.

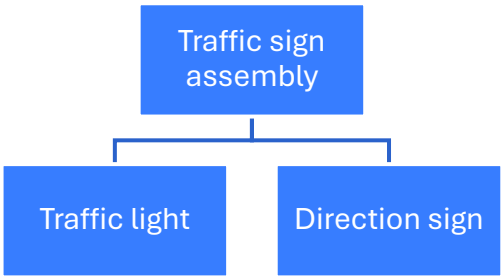
- When the document is revised, the property may or may not change. In the case where the property changes, the both the ReferenceDocument attribute and the name has to be updated.
- In the case where the property does not change, the property is according to both versions of the document and a reference to only one in the name becomes misleading.

When the reference disappear from the name, searches should also traverse the ReferenceDocument attribute of the property and the following DocumentReference. The display of search results could display only parts of the property name along with the document reference as:

- *Compressive streng… [EN 1168:2005+A3:2…]*
- *Compressive streng… [EN 14843:2007]*
- *Compressive streng… [EN 12843:2004]*
- *Compressive streng… [EN 15037-4:2010+A1:..]*

The traffic sign assembly template bundle

The traffic sign assembly information is modelled using bundled data templates. A top level data template for traffic sign assemblies comprises the two part templates for traffic lights and direction signs.



Each template contains the properties relevant to the information exchange. The properties are organized into groups of properties to provide the possibility to provide a narrower context or to fulfil a specific purpose.

The properties relevant to an assembly of traffic lights and directions signs are:

Property	**Groups of properties**			**Data templates**		
	Traffic sign assembly properties	Traffic light properties	Direction sign properties	Traffic sign assembly template	Traffic light template	Direction sign template
Name	X	X	X	X	X	X
Lane id	X	X	X		X	X
Height	X		X	X		X
Free clearance height	X		X	X		X
Lane centre position	X			X		
Intended traffic behaviour	X	X		X	X	
Lane traffic light signal colour	X			X		
Traffic light signal colour	X	X			X	
Traffic light behaviour indicator		X			X	
Colour			X			X

The data type, possible values and relationships to other properties for each property are:

Property	Data type	Possible values	Unit	Property relationship
Name	String			
Lane id	Integer			
Height	Decimal		any length unit	
Free clearance height	Decimal		m	ISO: Height
Lane centre position	Decimal		m	CP: Lane id
Intended traffic behaviour	Enum	Left-turn only; Left-turn or ahead; Straight ahead; Right-turn or ahead; Right-turn only; All ways		CP: Lane id
Lane traffic light signal colour	Enum	Red; Yellow; Green		CP: Lane id PP: Traffic light signal colour
Traffic light signal colour	Enum	Red; Yellow; Green		ISO: Colour
Traffic light behaviour indicator	Enum	Left arrow; Ahead arrow; Right arrow; Circle		FD: Intended traffic behaviour
Colour	Enum	Yellow; White; Blue		
ISO: Is specialization of, CP: Context parameter, PP: Proxy property, FD: Function dependency.				

Top-level traffic sign assembly template

The top-level traffic sign assembly template includes the following properties:

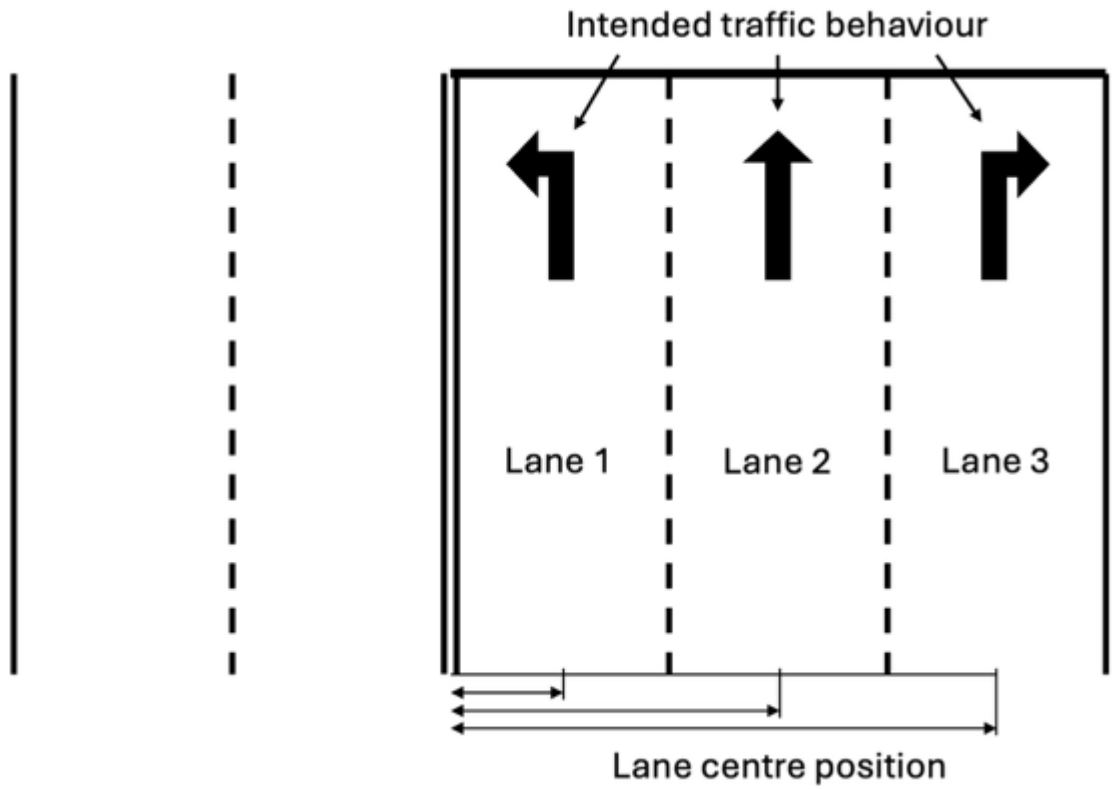
Property	Permitted values	Specified unit	Comment on use of property
Name			The name or designation of the assembly
Height		m	The total height of the traffic sign assembly

Free clearance height		m	The required free clearance height underneath the assembly
Lane centre position		m	The centre position for each lane entering a junction
Intended traffic behaviour	Left-turn only; Left-turn or ahead; Straight ahead; Right-turn or ahead; Right-turn only; All ways		Indication of the intended traffic behaviour of each lane
Lane traffic light signal colour	Red; Yellow; Green		The colour of the traffic light for each lane of traffic at a given time

Users are intended to provide information for the included properties.

In the payload, two additional properties – *lane id* and *traffic light signal colour* – are included because they are referred to through property relationships.

Lane id provides context to other properties as a context parameter. In the figure below, this is exemplified for the properties Intended traffic behaviour and Lane centre position. One value for the Lane id is not provided by the user when populating the template, but all possible values are provided as context specifiers.



Traffic light signal colour is referenced by Lane traffic light signal colour as a proxy property. The reason is that the traffic light colour property itself does not know or care about which

lane it applies to. To contextualise this such that the assembly level can hold four separate values, a separate layer must be created providing this context.

Direction sign template

The direction sign template includes the following properties:

Property	Permitted values	Specified unit	Comment on use of property
Name			The name of the direction sign (constructed from the destination names)
Lane id			The indication of which lane the direction sign is placed above
Height		mm	The height of the direction sign
Free clearance height		m	The free clearance height underneath the direction sign
Colour	Yellow; White; Blue		The colour of the direction sign background

Users are intended to provide information for the included properties.

The property Height is included both directly to describe the height of the sign and as a generic property for Free clearance height, but the user shall not provide two different values for the property Height.

The value assigned to Height represents the height of the sign. The other height value is assigned to Free clearance height, and represents the free clearance underneath the sign. The specialising relation is for “information purposes only”, helping users to understand that the free clearance height is a height measure.

Traffic light template

The traffic light template includes the following properties:

Property	Permitted values	Specified unit	Comment on use of property
Name			The name or designation of the assembly
Lane id			The indication of which lane the information pertains to
Traffic light signal colour	Red; Yellow; Green		The colour of the traffic light at a given time
Traffic light behaviour indicator	Left arrow; Ahead arrow; Right arrow; Circle		The shape of the traffic light signal reflecting the intended traffic behaviour
Intended traffic behaviour	*Left-turn only; Left-turn or ahead; Straight ahead; Right-turn or ahead; Right-turn only; All ways*		*Property used as input to the traffic light behaviour indicator*

Users are intended to provide information for the included properties.

The property Intended traffic behaviour indicate which behaviour the traffic light shall regulate. The way of regulating this, is by lighting the Traffic light behaviour indicator with the correct light colour. The Traffic light behaviour indicator has a function dependency to the Intended traffic behaviour.

The number of possible values for the traffic light behaviour indicator is less numerous than the possible values for the Intended traffic behaviour, but there is a unique relation between each Inteded traffic behaviour value and each Traffic light behaviour indicator value as:

Intended traffic behaviour	Traffic light behaviour indicator
Left-turn only	Left-arrow
Left-turn or ahead	Circle
Straight ahead	Circle or Ahead arrow
Right-turn or ahead	Circle
Right-turn only	Right-arrow
All ways	Circle

Part 3 – Exchange of data/values

The values of a property with context parameters are visualized as tables. The table below is presentation of a data sheet for the assembly level.

Property	**Value**		
Height	6,2 m		
Free clearance height	4,9 m		
	Context parameter: Lane id		
	1	2	3
Lane centre position	1,6 m	4,8 m	8,0 m
Intended traffic behaviour	Left-turn only	Straight ahead	Right-turn only
Lane traffic light signal colour	Red	Green	Green
Traffic light signal colour			

The values for the four direction signs are:

Property	**Direction sign templ.**	**Direction sign templ.**	**Direction sign templ.**	**Direction sign templ.**
Name	Majorstuen – Sentrum	Fornebu	Røa – Holmenkollen	Bogstad
Lane id	1	2	3	3
Height	594 mm	594 mm	594 mm	342 mm
Free clearance height	4,9 m	4,9 m	5,24 m	4,9 m
Colour	Yellow	Yellow	Yellow	White

The values for the four traffic light signals are:

Property	Traffic light template	Traffic light template	Traffic light template	Traffic light template
Name	Left-turn signal	Ahead signal, left	Ahead signal, right	Right-turn signal
Lane id	1	2	2	3
Intended traffic behaviour	Left-turn only	Straight ahead	Straight ahead	Right-turn only
Traffic light signal colour	Red	Green	Green	Green
Traffic light behaviour indicator	Left arrow	Circle	Circle	Right arrow