| To (department acronym name) | | For information (department acronym name) | |
| --- | --- | --- | --- |
|  |  |  |  |

Business Rules

# Edit

To perform an edit the user has to be in a workspace and select a SOP-date that has change requests with status open and a SOP-date that is less than or equal to the selected SOP-date

## Edit Entity

All entities have to be valid before they can be saved to the database.  
The validation rules are described in another document.

## Edit Message Signal

When adding a signal to a message the signal will get the first available position in the message. The user can then move the signal to a specific location if they which to and it is possible to do so until the signal has been committed.

If we for some reason need to move a signal that has already been committed then we have to do it through the database.

## Edit Direct Wire Port Data

The user can create a Direct Wire link by specifying a transmitting allocation element and a receiving allocation element. Specifying only one side is not valid for Direct Wire.

## Edit Internal Port Data

The user can create an Internal link by specifying a transmitting allocation element and a receiving allocation element. Specifying only one side is not valid for Internal.

## Edit Can Communication

The user can either choose to create the receiving, the transmitting or both sides when creating the communication.

### Segments

When selecting a transmitting or receiving allocation element the user also has to select a segment. The list of segments represents the segments that the component, which the allocation element is allocated on, can communicate on. This can be changed by adding/removing Segment relations for a component.

If the transmitting segment is different from the receiving segment the user is requested to create a gateway unless there already exists one.

### User Function tagging

When creating a receiver the user is also asked to tag the receiver to a specific UF but this is not mandatory.

### Destination Address

If any of the message signals selected are connected to a message that is either P2P or exceeds 8 byte in length, the destination address is required.

## Create Gateway

When creating a new gateway, the user is presented with a suggested path which will be the shortest path between the starting segment and the ending segment.

If the user is not satisfied with the suggested path, then they can alter it.

In order to choose a component as Gateway it needs to be able to communicate on both the From and To segment. This can be changed by adding/removing Segment relations for a component.

A gateway is only valid if the following conditions are satisfied:

* The gateway contains no duplicate segments or components.
* The gateway starts on the transmitting segment.
* The gateway ends on the receiving segment.
* An identical gateway does not currently exist for this communication.

The gateway algorithm then combines the possible gateway paths to display all the possible routes to the user.

Example:

We currently have the following gateways:

Red –> Comp1 –> Green  
Green –> Comp 2 –> Orange

The possible gateway paths are:  
Red –> Comp1 –> Green  
Green –> Comp 2 –> Orange

Red –> Comp1 –> Green –> Comp2 –> Orange

If we then decide to add the following gateway path:

Red –> Comp3 –> Green

We end up with the following possible gateway paths:

Red –> Comp1 –> Green  
Green –> Comp 2 –> Orange

Red –> Comp3 –> Green

Red –> Comp1 –> Green –> Comp2 –> Orange

Red –> Comp3 –> Green –> Comp2 –> Orange

# Dependencies

## Active

An entities active state is sometimes dependent upon other entities active state.

An entity can also sometimes not have their own active state and instead rely solely on other entities active state.

If the dependency is not always present (Not required) then the relation can be marked as optional and it will not be considered when calculating IsActive state.

The active state is currently calculated as following:

**If the entity has their own active state:**

If the entity is active and its dependent entities are active then the entity is active.

If the entity is active and one of its dependent entities are inactive then the entity is inactive.

If the entity is inactive and its dependent entities are active then the entity is inactive.

**if the entity does not have their own active state:**

If the dependent entities are active then the entity is active.

If one of the dependent entities are inactive then the entity is inactive.

## Commit

The commit functionality in Sesamm Tool 2 allows for partial commits.

When committing an entity you have to also commit all of the parents that this entity depends on and currently only exists in this workspace.

**Example:**

Create a new Function Category.  
Create a new User Function and connect it to the previously created Function Category.

Commit User Function.

Result: Both the User Function and the Function Category will be selected for commit.

## Revert

The revert functionality in Sesamm Tool 2 also allows for partial reverts.

When reverting an entity you have to also revert all of the children that have a dependency towards this entity and currently only exists in this workspace.

**Example:**

Create a new Function Category.  
Create a new User Function and connect it to the previously created Function Category.

Revert Function Category.

Result: Both the Function Category and the User Function will be selected for revert.

### Exception

One exception exists though and that is the Can Port Data and its relations to Ae Port and Gateway.

If we revert an Ae Port we also revert the Can Port Data if no other Ae Ports or Gateways are connected to it.

If we revert a Can Port Data we also have to revert the Ae Port if any and the gateways connected to this Can Port Data.

# Can Specification

## Filters

When generating a Can Specification a number of filters can be applied to control what will be included in the Can Specification.

The following choices are possible:

* SOP
* Test Week
* Workspace
* Commit Date
* Include/Exclude specific Change requests

**SOP**

Includes active communication up to that specific point.

If SOP has been selected, test week cannot be used.

**Test Week**

Includes active communication up to that specific point.

If Test Week has been selected, SOP cannot be used.

**Workspace**

Include things from a specific workspace. Useful when you want to see how the changes in your workspace affects a component.

**Commit Date**

Only include changes up to this time.

## Can Identifiers

Every message in the Can Specification has an Id that corresponds with its pgn, source address, destination address and prio.

If a message is larger than 8 byte it has to be sent with the transport protocol.  
This will cause the message to have a slightly different Can Identifier than it would otherwise have.

**Example of a standard Can Identifier:**

Source Address: 27

Destination Address: 1D

PGN: 0x0EF00

Prio: 3

Results in the following identifier:

0CEF1D27

Prio is left shifted 26 position which will result in:

C000000

The PGN is then left shifted 8 position which will result in:

EF0000

These two values and the source address are then added together to form:

CEF0027

If the message is P2P (Which this message is since it has a destination address) the destination address will then be left shifted 8 positions and added to the value:

1D00

CEF1D27

The number is then padded with a zero if it does not exceed 7 digits resulting in:

0CEF1D27

**Example of a TP Can Identifier:**

Source Address: 0x10

PGN: 0x0FEE1

Prio: 6

Results in the following identifier:

18ECFF10

18EBFF10

(FEE1)

This identifier exists of two specific identifiers, one for TP\_CM and one for TP\_DT and then the PGN of the original message.

The identifiers are calculated in the same way as the standard identifier but instead using the PGN of the message TP\_CM(0x0EC00) and TP\_DT(0x0EB00).

If the message has no destination address then it will be FF (Global).

## Change notes

A user can include change notes in the Can Specification by adding a reference filter (The same way as in 3.1).

The entities are then diffed and the result is presented at the end of the Can Specification.

## Excluded entities

* Entities that are inactive are not included in the Can Specification.
* Value Descriptions that have been excluded are not part of the Can Specification