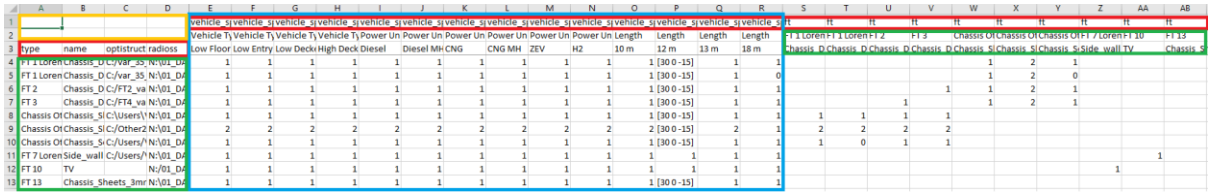
Compatibility CSV

- Csv uses comma as delimiter
- Paths can contain both types of slashes / and \
- Compatibility values:
    - 0 – includes are not compatible
    - Empty – includes are not compatible (recommended for non-subordinate combinations)
    - 1 (or any other integer excluding 0) – includes are compatible
    - [x y z] (e.g. [300 0 0]) – includes are compatible with the specified include offset in the line. Space, comma and semicolon can be used as value separator. In the case that multiple offsets are specified for a given include in a given assembly, the one that is closer to the include in the hierarchy upwards is preferred (i.e. the offset given by the Vehicle Specification has the least priority).
- Structure:



Leave empty
Leave keywords and layout
Definition of includes in database
Compatibility of includes with Vehicle Specification

Hierarchy YAML

- Synopsis of YAML Basic Elements
    - The synopsis of YAML basic elements is given here: Comments in YAML begins with the (**#**) character.
    - Comments must be separated from other tokens by whitespaces.
    - Indentation of whitespace is used to denote structure.
    - Tabs are not included as indentation for YAML files.
    - List members are denoted by a leading hyphen (**-**).
    - List members are enclosed in square brackets and separated by commas.
    - Associative arrays are represented using colon **(:)** in the format of key value pair. They are enclosed in curly braces **{}**.
    - Multiple documents with single streams are separated with 3 hyphens (---).
    - Repeated nodes in each file are initially denoted by an ampersand (**&**) and by an asterisk (*****) mark later.
    - YAML always requires colons and commas used as list separators followed by space with scalar values.

- File must have main item *groups* which must contain two items: *vehicle_spec* and *FT groups*. Rest of the hierarchy is customizable by user.
- Items in *vehicle_spec* have only attribute *name*
- Items in *FT groups* represent groups of FTs and have 3 mandatory and 1 optional attributes:
    - `Name` – *mandatory, string*
    - `Skippable` – *mandatory, boolean* – determines whether the FT group is skippable in the compatibility definition. It therefore governs whether child elements can be

inserted into the assembly if FTs from this group are not selected. If the value is False, no FTs are selected in this group, and no FTs from the child group can be inserted into the assembly.

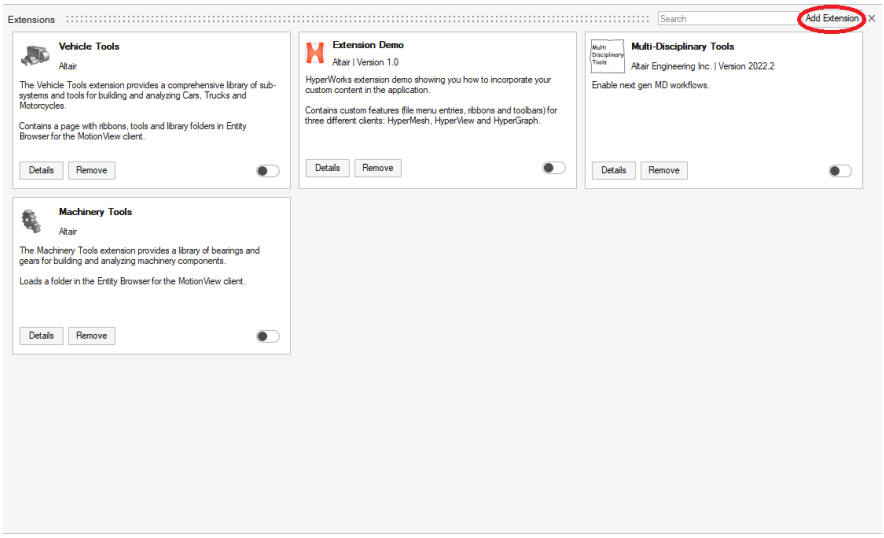
- `FTs` – *mandatory* – list of FTs in this group. Every FT has 2 attributes:
  - `Name` – *mandatory, string*
  - `Multiselection` – *mandatory, boolean* – determines whether multiple includes can be selected for the buld-up from the given FT, or only one
- `Groups` – *optional* – can list all groups directly subordinated to current group

```yaml
groups:
  vehicle_spec:
      - name: Vehicle Type
      - name: Power Unit Type
      - name: Length
  FT groups:
      - name: Chassis Structure Assembly
        skippable: True
        FTs:
          - name: FT 1 Lorem ipsum
            skippable: False
            multiselection: False
          - name: FT 2
            skippable: False
            multiselection: False
          - name: FT 3
            skippable: False
            multiselection: False
        groups:
          - name: Chassis Equipment
            skippable: True
            FTs:
              - name: FT 4
                skippable: False
                multiselection: False
              - name: FT 5
                skippable: False
                multiselection: False
              - name: FT 6
                skippable: False
                multiselection: False
              - name: FT 5-1
                skippable: False
                multiselection: False
              - name: FT 6-1
                skippable: False
                multiselection: False
              - name: Chassis Others
                skippable: False
                multiselection: True
            groups:
              - name: Next group
                skippable: False
                FTs:
                  - name: FT 44
                    skippable: False
                    multiselection: False
                  - name: FT 55
                    skippable: False
                    multiselection: False
      - name: Left Lateral Structure Assembly
        skippable: True
        FTs:
          - name: FT 7 Lorem ipsum
            skippable: False
```

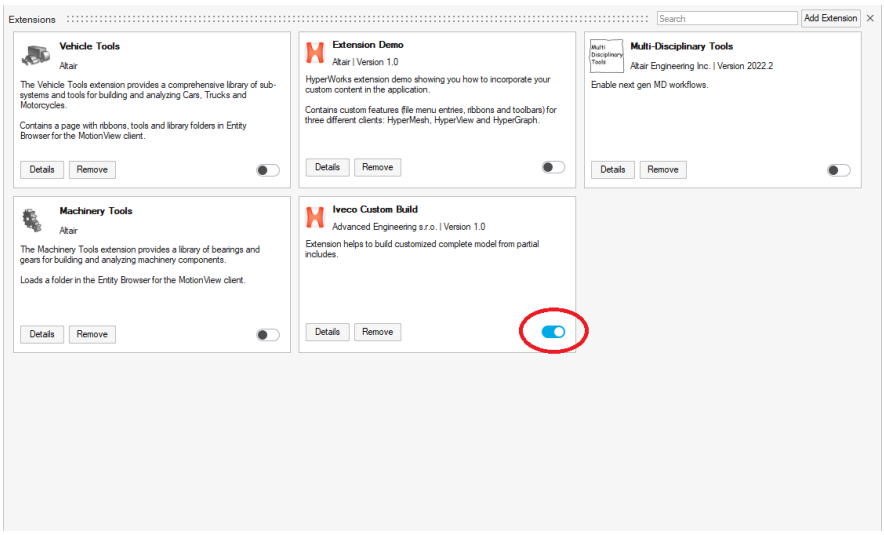
Installation

1) Copy main folder where you want to store it
2) Open config.yaml in main folder and edit paths to compatibility csv and hierarchy yaml.

3) Open HyperMesh, select File – Extensions – Add Extension

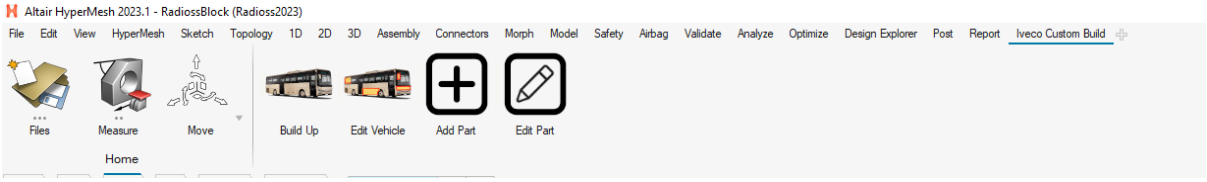


4) Select folder with the extension
5) Switch the extension ON



Using extension

The extension is placed as a new ribbon in HyperMesh and contains 4 buttons.

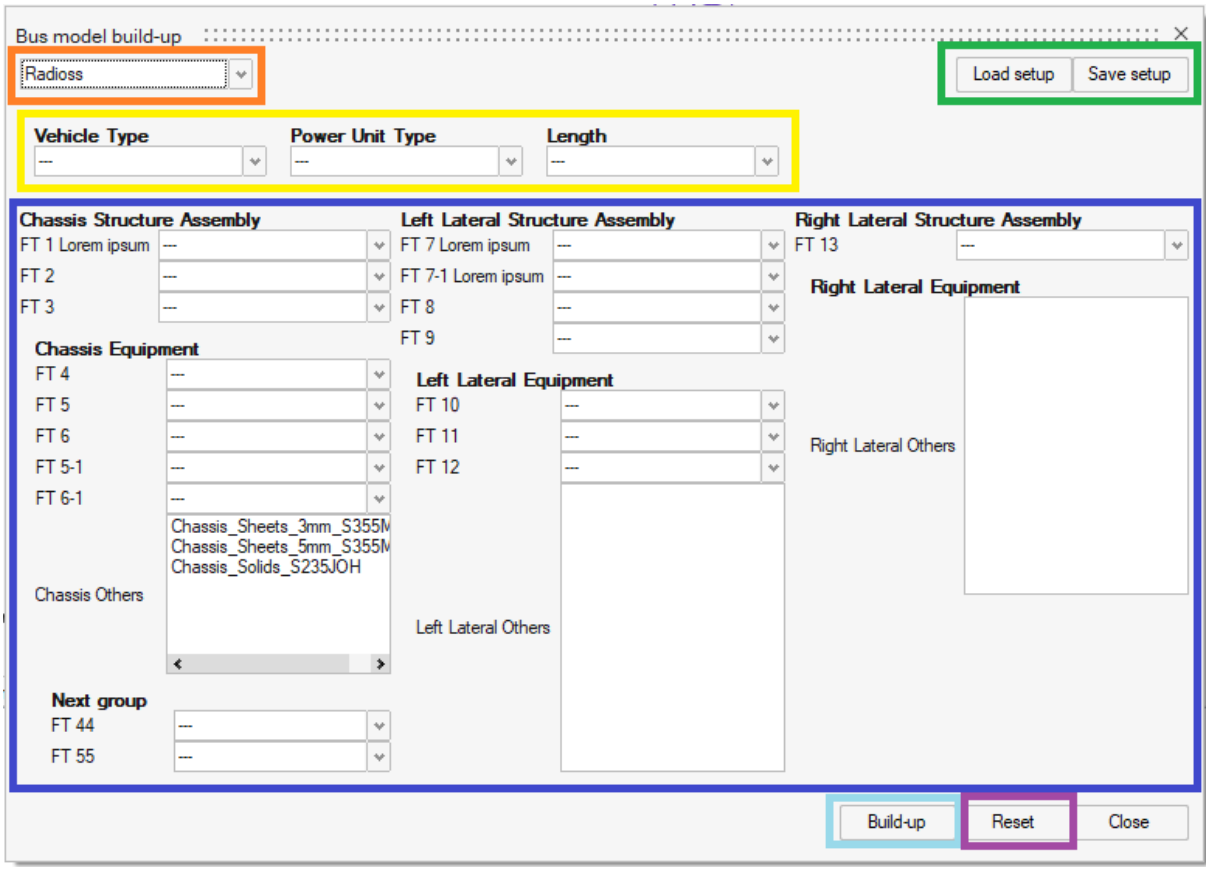


First two buttons serve for creating and editing model of vehicle, rest two serve for maintenance of Part (Includes) database.

It is not necessary to GUI, for managing database. User can also manage database as usual CSV in any editor (e.g. Excel), especially in case of extensive changes. But in case of adding or editing single parts, using of GUI is more effective.

Build-Up

Serves for creating new model and for importing new includes into currently opened model.



Select solver interface in which will the model be created
You can use the Save setup button to save the currently selected FTs and Vehicle specification as a template that you can return to at any time using the Load setup button. This template uses the yaml format for saving.

Selection of Vehicle specification

Selection of FTs. Structure of this table depends on your hierarchy definition in Hierarchy yaml. Notice that first level of FT groups is divided to columns. Other levers of FT groups are then under superior group and is visually indented from the left side. Every group with bold title contains relevant FTs

Build-up button imports a vehicle model in selected configuration. When there is already opened model in HyperMesh, this model is preserved and the selected FTs are imported into it.

Reset button reverts all changes in selection and gives the table to default empty setup

In the setup table there are shown only relevant parts, not all from database. Relevant parts are those that

- have an existing include file for the currently selected solver interface (if Radioss include file is missing for a part and Radioss interface is selected in the window, the part is not shown in selection boxes)
- are compatible with selected superior parts

Edit Vehicle

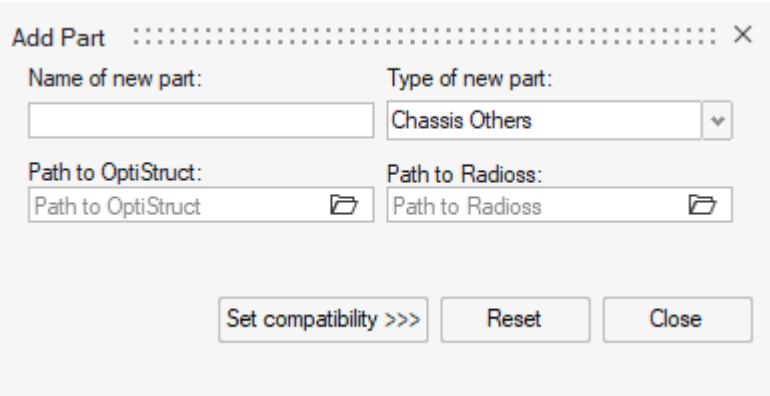
Serves for editing currently loaded model of vehicle. It loads all current includes into setup table and user can deselect them to remove it, change the selection to replace them and select new to add new.

The GUI is similar as in Build Up, but solver interface selection is removed. When user wants to change solver interface of current setup, it is recommended to Save setup via button in Edit Vehicle, then open New (empty) model in HyperMesh, open Build Up, load previously saved setup, change solver interface and Build-up this new model.

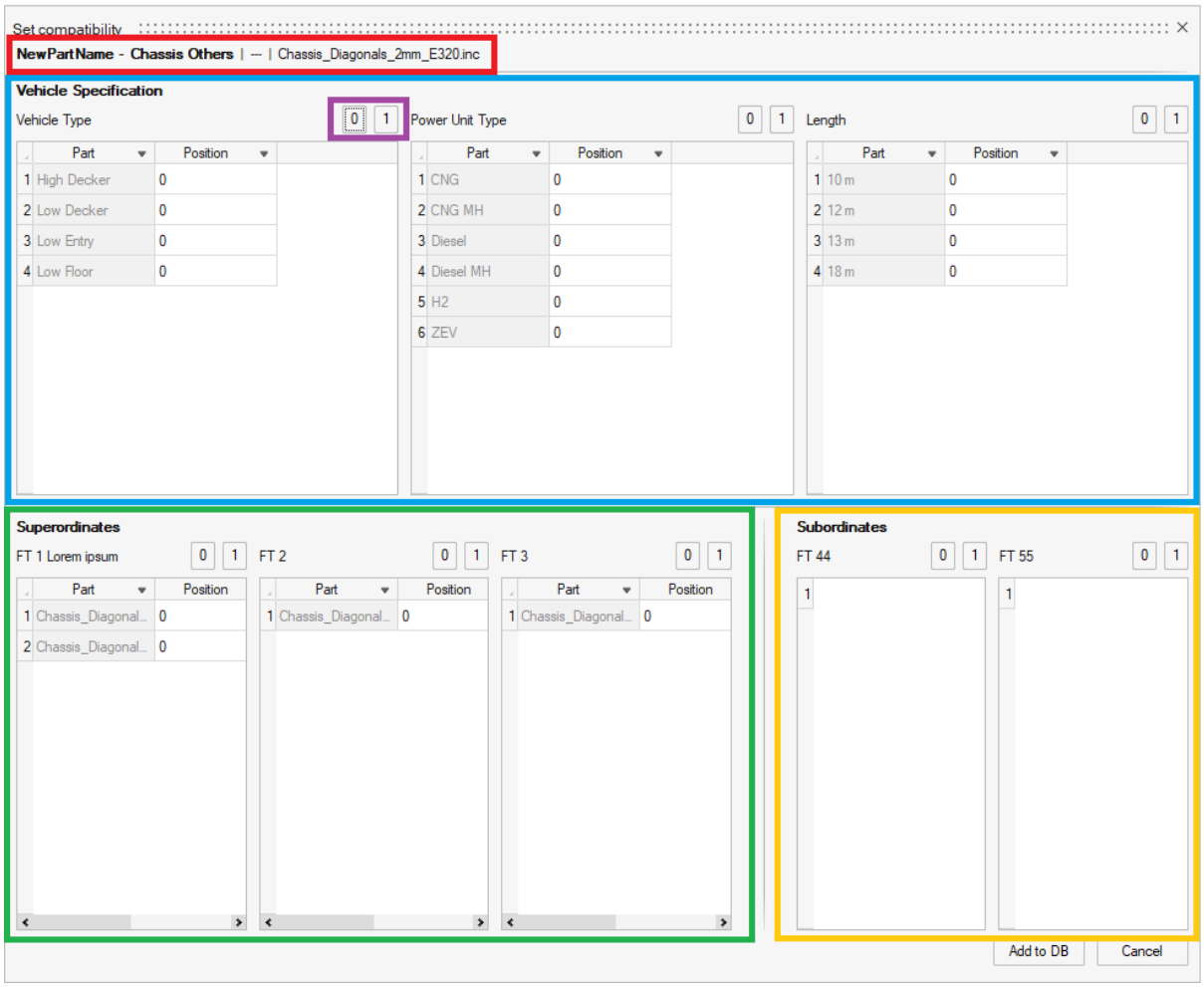
Add Part

Serves as GUI for adding new includes to database which is saved in compatibility CSV.

It is necessary to fill name of new part (must be unique in database), select type (FT) the part and fill in at least one path to include file. The file must exist. It is not necessary to fill in paths to both files – it is assumed that the part may not have a twin for the second solver interface.



After selecting necessary data, continue by clicking button Set compatibility which will open new window.



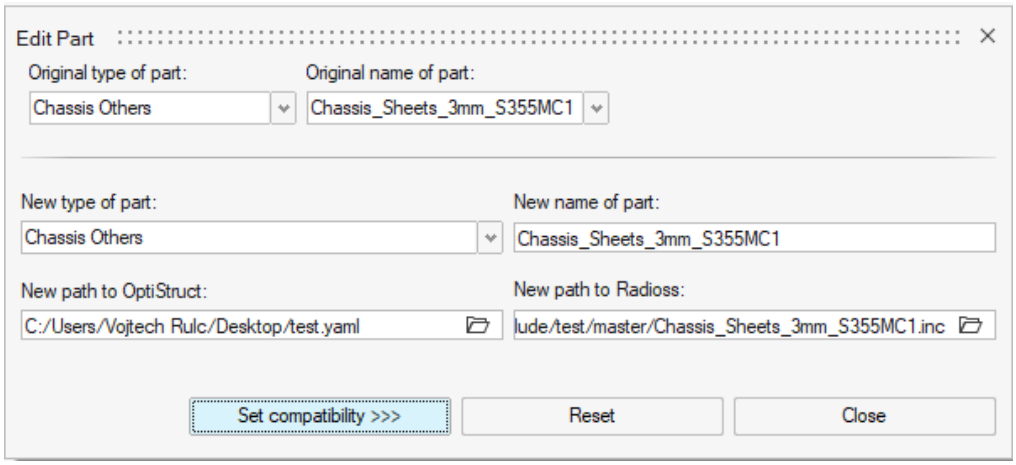
Information about currently being added part for check. Name – Type | OS file | Radios file
Compatibility with Vehicle Specification
Compatibility with superordinant FTs
Compatibility with subordinant FTs
These buttons quickly assign selected value to all parts in the FT

The setup is confirmed by Add to DB button.

For relevant compatibility values, which is possible to fill, please see *Compatibility CSV* chapter in this document.

Edit Part

Serves for editing part (include) which is already in database or duplicating that part with some changes. Unlike the Add Part dialog, it also contains the selection boxes Original type of Part and Original name of part. With these, the user selects which part he wants to edit or use as default for duplication with editing.



After editing the data, user continues by clicking Set compatibility. That opens again new window same as in Add Part dialog, but with one difference: For confirmation there not only one button, but two of them:

- **Edit in DB** – it edits the selected part
- **Add as new** – it leaves original part untouched, and uses all filled data (including data in previous window) for creating new part. This is very effectively way to add new part which has same or similar compatibility as any already existing part in DB