

IVECO Custom Build

Documentation

Version 1.0



Martin Kuklík

Vojtěch Rulc

2024

Contents

Meshing Methodology	3
HyperMesh Extension	4
Compatibility CSV	5
Hierarchy YAML	6
Installation.....	7
Using extension	9
Build-Up	9
Edit Vehicle.....	10
Add Part	10
Edit Part.....	11

Meshing Methodology

Martin

HyperMesh Extension

Work with assembling and with database is realized by HyperMesh Extension.

The current trend in HyperMesh customization is the increasing use of Extensions, which are modular add-ons that extend the core functionalities of the software. Extensions can be developed to provide new capabilities, automate complex processes, and integrate HyperMesh more seamlessly with other tools and systems in the engineering workflow. This trend reflects a broader move towards modular and scalable customization, where users can easily add, update, or remove specific functionalities as needed. By leveraging Extensions, users can keep their HyperMesh environment up-to-date with the latest tools and innovations, ensuring that their simulation capabilities continue to evolve alongside industry advancements. This approach not only enhances the flexibility and power of HyperMesh but also fosters a collaborative ecosystem where users can share and benefit from community-developed Extensions.





The approach with Extensions was chosen due to its sustainability in maintaining customization across successive versions of HyperMesh. Unlike traditional customization methods, which often require significant rework when software updates are released, Extensions offer a modular and adaptable framework. This ensures that custom functionalities remain compatible with new versions of HyperMesh, reducing the need for extensive modifications and minimizing disruptions to the workflow. By adopting Extensions, users can enjoy a stable and continuous customization experience, allowing them to focus on innovation and efficiency without worrying about the compatibility issues that typically accompany software upgrades. This sustainable approach to customization ensures that investments in developing custom tools and processes continue to yield benefits over the long term.

In following chapters, let's see which prerequisites must be prepared before using the extension, how to install it and how to use its GUI.

Compatibility CSV

- Csv uses comma as delimiter
- Paths can contain both types of slashes / and \
- Compatibility values:
 - o 0 – includes are not compatible
 - o Empty – includes are not compatible (recommended for non-subordinate combinations)
 - o 1 (or any other integer excluding 0) – includes are compatible
 - o [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:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1					Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle
2					Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty
3	Type	Name	Opt	Structure	radios	Low Floor	Low Entry	Low Deck	High Deck	Diesel	Diesel MH	CNG	CNG MH	ZEV	H2	10 m	12 m	13 m	18 m	FT 1	FT 1	FT 1	FT 1	FT 1	FT 1	FT 1	FT 1	FT 1
4	FT 1	Loren	Chassis	D C/VAR_35	N/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	FT 1	Loren	Chassis	D C/VAR_35	N/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	FT 2	Chassis	D C/FT2	var N/01_DA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	FT 3	Chassis	D C/FT4	var N/01_DA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	Chassis	Of	Chassis	SI C/USERS/	N/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	Chassis	Of	Chassis	SI C/Other2	N/01_DA	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
10	Chassis	Of	Chassis	SI C/USERS/	N/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	FT 7	Loren	Side	_wall	C/USERS/	N/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	FT 10	TV		N/01_DA		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	FT 13	Chassis	Sheets	3m/	N/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

-  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 build-up from the given FT, or only one
 - Groups – *optional* – can list all groups directly subordinated to current group

```

0          10          20          30          40
1 groups:
2   vehicle_spec:
3     - name: Vehicle Type
4     - name: Power Unit Type
5     - name: Length
6   FT groups:
7     - name: Chassis Structure Assembly
8       skippable: True
9       FTs:
10        - name: FT 1 Lorem ipsum
11          skippable: False
12          multiselection: False
13        - name: FT 2
14          skippable: False
15          multiselection: False
16        - name: FT 3
17          skippable: False
18          multiselection: False
19      groups:
20        - name: Chassis Equipment
21          skippable: True
22          FTs:
23            - name: FT 4
24              skippable: False
25              multiselection: False
26            - name: FT 5
27              skippable: False
28              multiselection: False
29            - name: FT 6
30              skippable: False
31              multiselection: False
32            - name: FT 5-1
33              skippable: False
34              multiselection: False
35            - name: FT 6-1
36              skippable: False
37              multiselection: False
38            - name: Chassis Others
39              skippable: False
40              multiselection: True
41          groups:
42            - name: Next group
43              skippable: False
44              FTs:
45                - name: FT 44
46                  skippable: False
47                  multiselection: False
48                - name: FT 55
49                  skippable: False
50                  multiselection: False
51            - name: Left Lateral Structure Assembly
52              skippable: True
53              FTs:
54                - name: FT 7 Lorem ipsum
55                  skippable: False

```

Installation

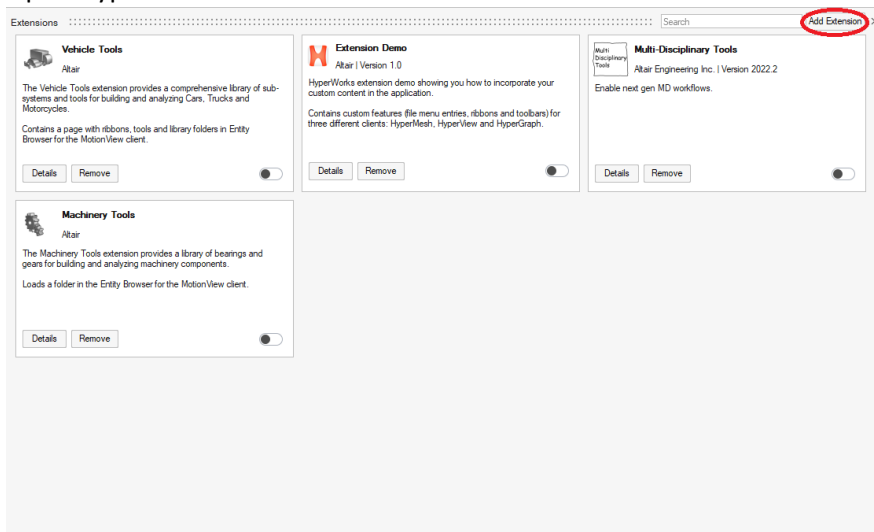
- 1) Copy main folder of extension where you want to store it
- 2) Open config.yaml in main folder and edit paths to compatibility csv and hierarchy yaml.

```
0 10 20 30 40 50 60 70 80 90 100
1 paths:
2   csv: "N:/01_DATA/01_PROJECTS/103_Iveco_Model_Buildup/01_data/01_python/compatibility.csv"
3   hierarchy: "N:/01_DATA/01_PROJECTS/103_Iveco_Model_Buildup/01_data/01_python/types_hierarchy.yaml"
4   log: ""
```

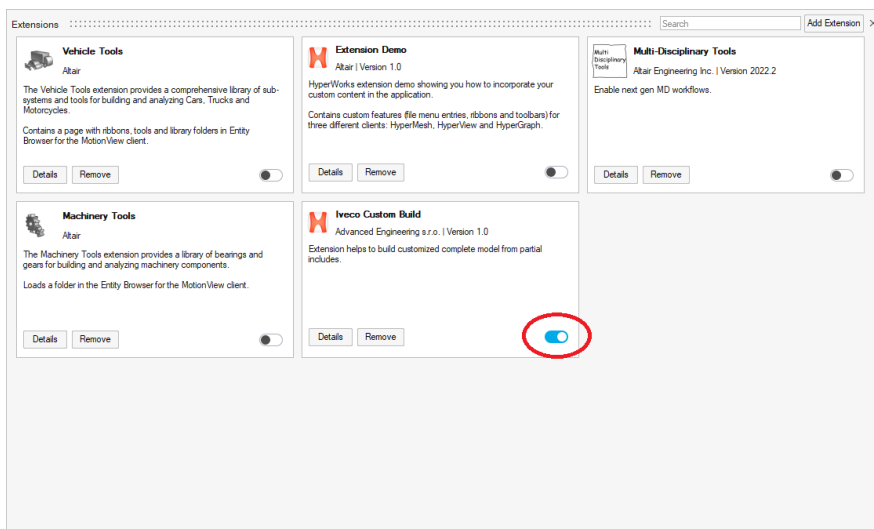
Name of the files is not prescribed and can be any. *csv* and *hierarchy* are required, *log* can be empty. In that case, the log file will be created as

"C:\Users\Username\Documents\Altair\iveco_log.log"

- 3) Open HyperMesh and select: File – Extensions – Add Extension




- 4) Select folder of the extension
- 5) Switch the extension ON




- 6) Then all details, including link to this documentation can user find in extension description

Extensions

**Multi-Disciplinary Tools**
Altair Engineering Inc. | Version 2022.2
Enable next gen MD workflows.


Details Remove

☐

**Machinery Tools**
Altair
The Machinery Tools extension provides a library of bearings and gears for building and analyzing machinery components.
Loads a folder in the Entity Browser for the MotionView client.


Details Remove

☐

**Iveco Custom Build**
Advanced Engineering s.r.o. | Version 1.0
Extension helps to build customized complete model from partial includes.

Details Remove

☒

**Iveco Custom Build**
Advanced Engineering s.r.o. | Version 1.0

Directory
[N:\01_DATA\01_PROJECTS\103_Iveco_Model_Buildup\01_data\01_python\](#)

Minimum required version
2023.1

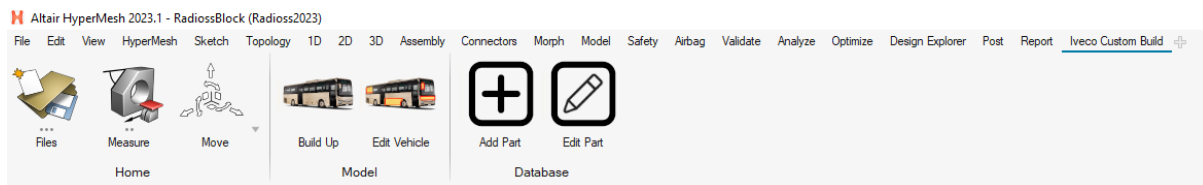
Description
Extension helps to build customized complete model from partial includes.
[View Documentation](#)

Remove

8

Using extension

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

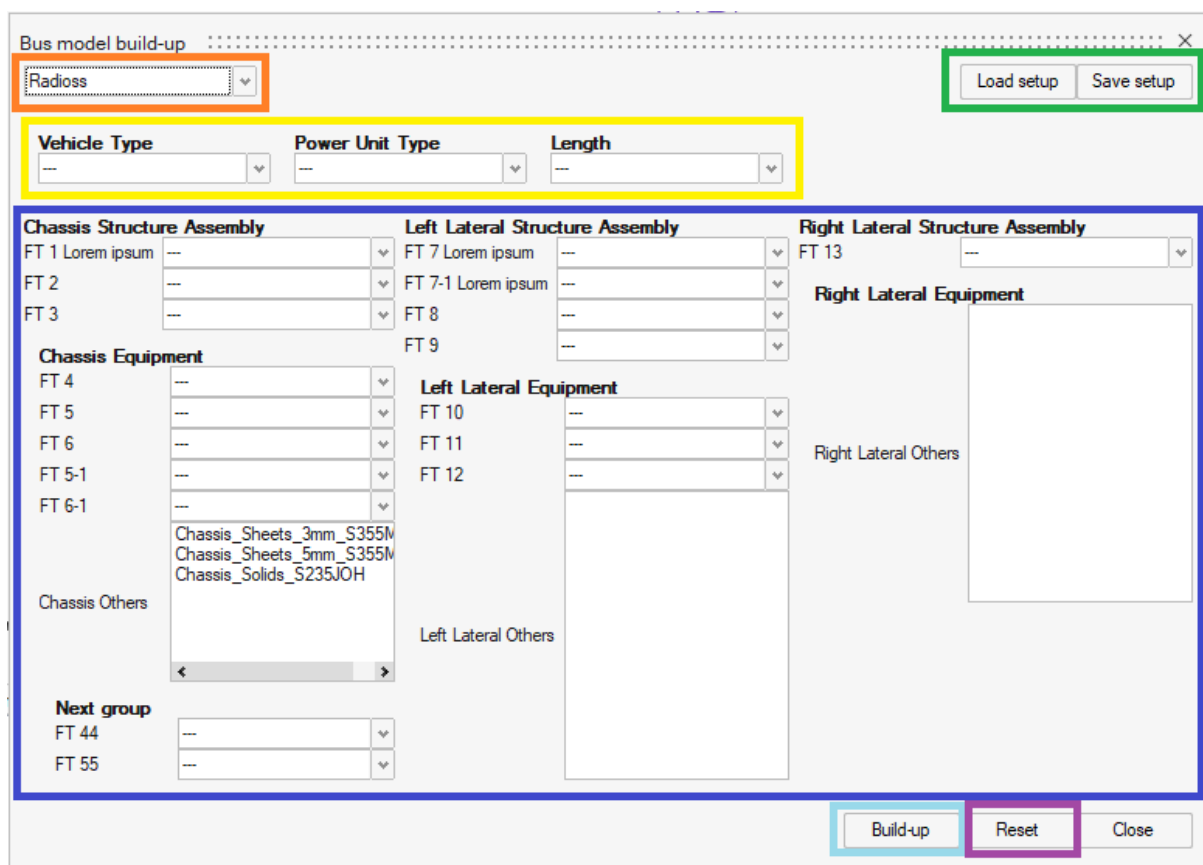






First two buttons serve for creating and editing HyperMesh model of vehicle, rest two serve for maintenance of Part (Includes) database.

It is not necessary to use 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.

Add Part : X

Name of new part: Type of new part:

Path to OptiStruct: Path to Radioss:

Set compatibility >>> Reset Close

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

Set compatibility X

NewPartName - Chassis Others | -- | Chassis_Diagonals_2mm_E320.inc

Vehicle Specification

Vehicle Type ☐ 0 ☒ 1 Power Unit Type ☐ 0 ☒ 1 Length ☐ 0 ☒ 1

Part	Position
1 High Decker	0
2 Low Decker	0
3 Low Entry	0
4 Low Floor	0

Part	Position
1 CNG	0
2 CNG MH	0
3 Diesel	0
4 Diesel MH	0
5 H2	0
6 ZEV	0

Part	Position
1 10 m	0
2 12 m	0
3 13 m	0
4 18 m	0

Superordinates

FT 1 Lorem ipsum ☐ 0 ☒ 1 FT 2 ☐ 0 ☒ 1 FT 3 ☐ 0 ☒ 1

Part	Position
1 Chassis_Diagonal...	0
2 Chassis_Diagonal...	0

Part	Position
1 Chassis_Diagonal...	0

Part	Position
1 Chassis_Diagonal...	0






Subordinates

FT 44 ☐ 0 ☒ 1 FT 55 ☐ 0 ☒ 1

Part	Position
1	

Part	Position
1	

Add to DB Cancel

-  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.

Edit Part X
 Original type of part: Original name of part:
 Chassis Others Chassis_Sheets_3mm_S355MC1
 New type of part: New name of part:
 Chassis Others Chassis_Sheets_3mm_S355MC1
 New path to OptiStruct: New path to Radioss:
 C:/Users/Vojtech Rulc/Desktop/test.yaml lude/test/master/Chassis_Sheets_3mm_S355MC1.inc
 Set compatibility >>> Reset Close

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