IVECO Custom Build

Documentation

Version 1.0

ADVANCED ENGINEERING s.r.o.

Martin Kuklík

Vojtěch Rulc

2024

Contents

Meshing Methodology

The Iveco custom build tool is designed for rapid speed up of preprocessing time. The principle is based on dividing the entire vehicle model into individual parts – include files. The splitting of the whole model is determined by the user, the script ensures and partially controls the correct connection of the individual includes.

The presented tool and methodology were developed and tested by using model in two different codes. Radioss and Optistruct. The picture below shows an example of splitting entire model to include files for Radioss. Basically, include files represents frontal, middle and rear parts of vehicle. This representation allows to create easily two vehicle versions, for example 12 meters long and 10meters long. There is also particular split for frontal axle, faces and chassis.

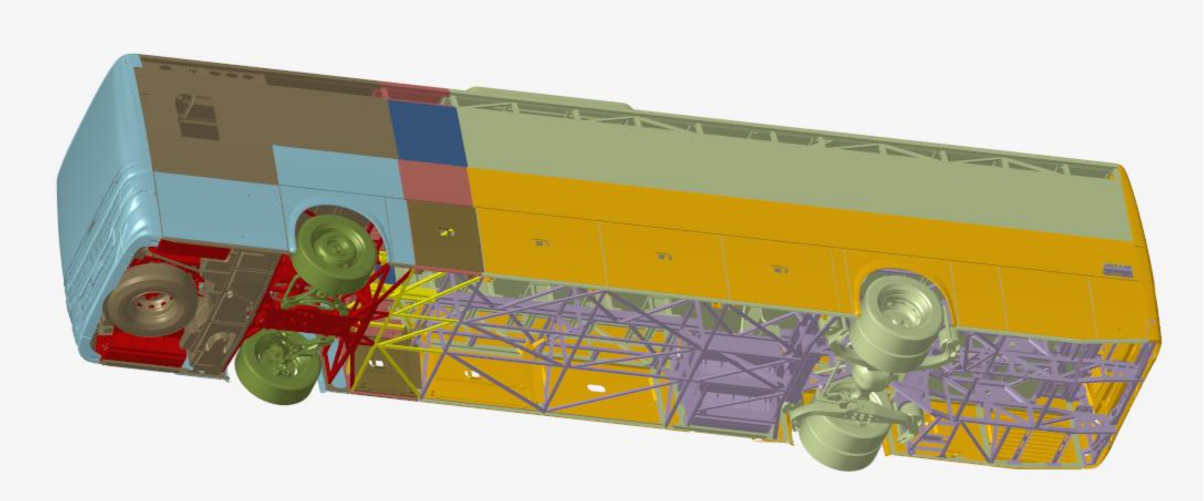


Fig. Model break-up into include files shown by colors.

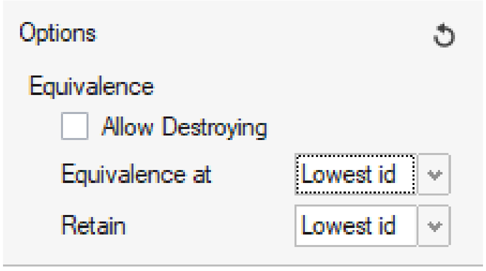
Connection Types

The script connects individual include files to each other in two ways:

1. Node to node connection

This is realized by standard HyperMesh Equivalence feature.

It works with a default tolerance of 0.04mm and the following settings:



Options

Equivalence

□ Allow Destroying

Equivalence at: Lowest id

Retain: Lowest id

Fig. Equivalence settings

It is internally disabled to destroy any element to avoid unexpected changes in the mesh. It also allows equivalence to be applied to zero-length elements with one free end.

The equivalence settings mentioned above are determining requirements for mesh included in include files. The nodes to be connected must correspond in their position, or their position must not differ by more than the tolerance value.

2. Connectors

Equivalence feature is able to fully cover the needs of connecting entities between include files. The limitation is obvious. The user is forced to prepare the 100% correct mesh in such a way that the connecting entities between the include files always must have nodes in the same position. User has to also model manually connecting entities like rigid bodies. This is in case of large model very time consuming.

This disadvantage can be largely eliminated by using connectors. There are 2 connector types introduced for Radioss model and 1 connector type for Optistruct model. Following pictures show typical examples where connectors are very effective to use.

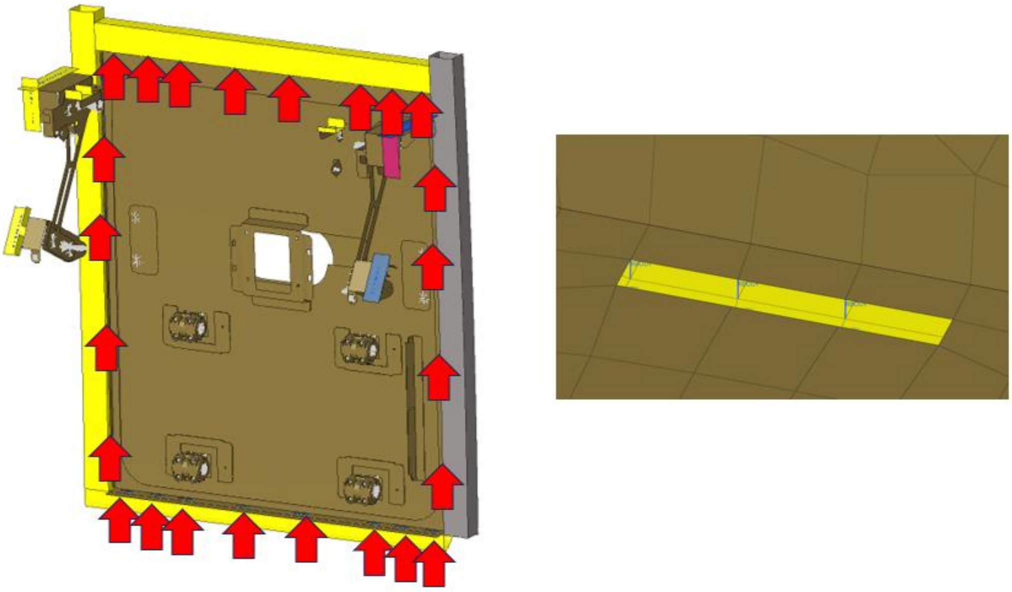


Fig. Connection place suitable for connector application

2A. Connector Rigid Spotweld for Radioss.

STEP1: Make a sure that you have selected correct include file as current. This will place connector into desired include. You can also create connector when only one include file is loaded in Hypermesh. Connection to entities from another include file is realized automatically later. Now select the nodes at desired weld location.

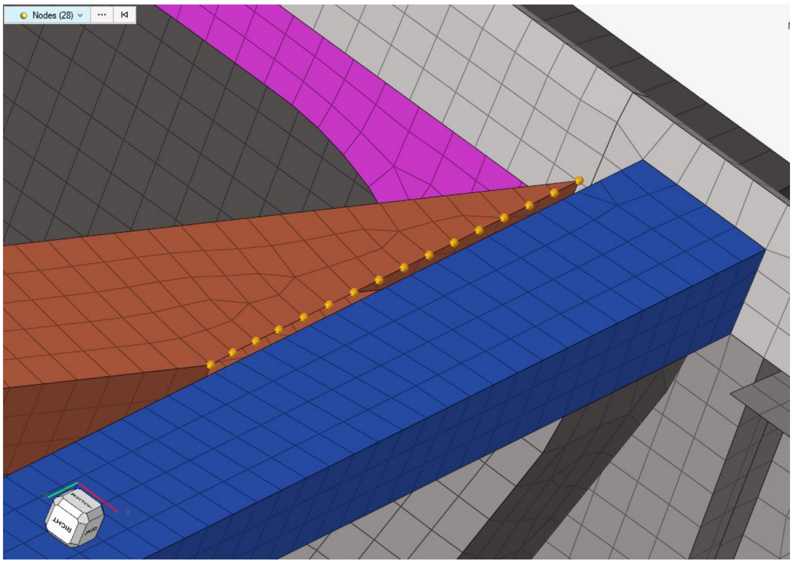
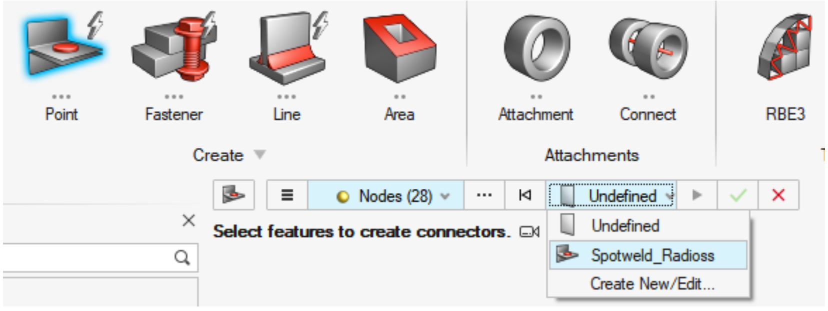


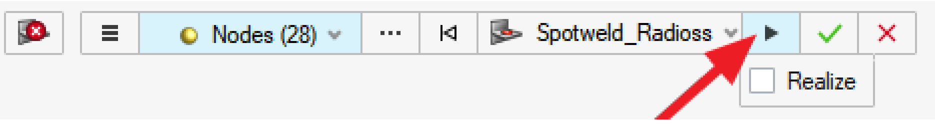
Fig. Connection place suitable for connector application

STEP2: Select Point connector in Connector pull-down menu. The point creation feature is taking over your node selection. Select “Spotweld_Radioss” connector control. Connector control is available as file Spotweld_Radioss.hm in Accessories folder in the extension’s installation folder.

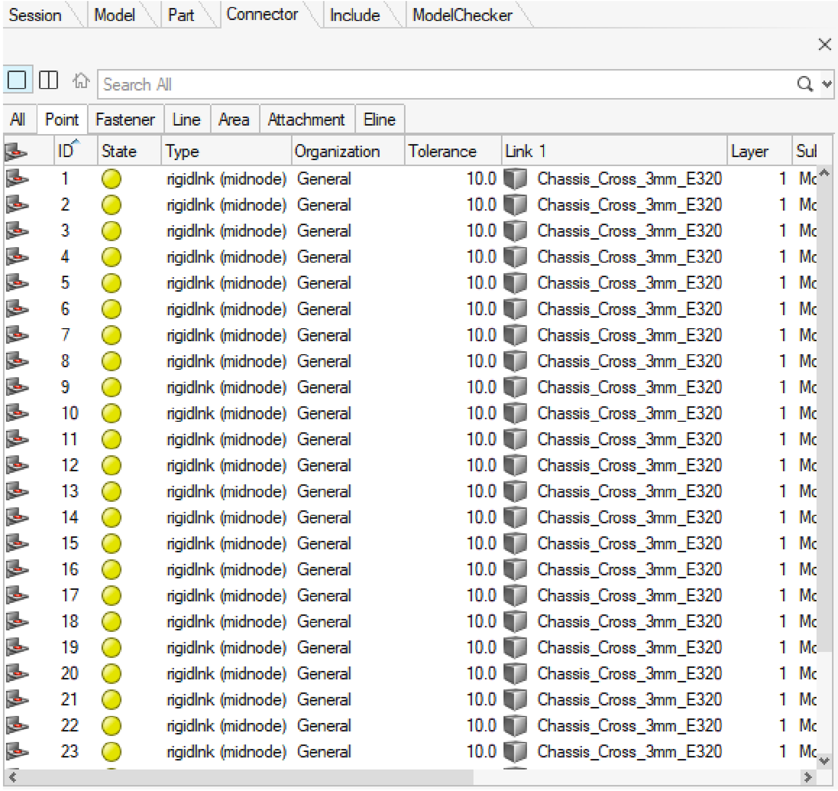


Select features to create connectors.

STEP3: Confirm connector realization by click on run button.

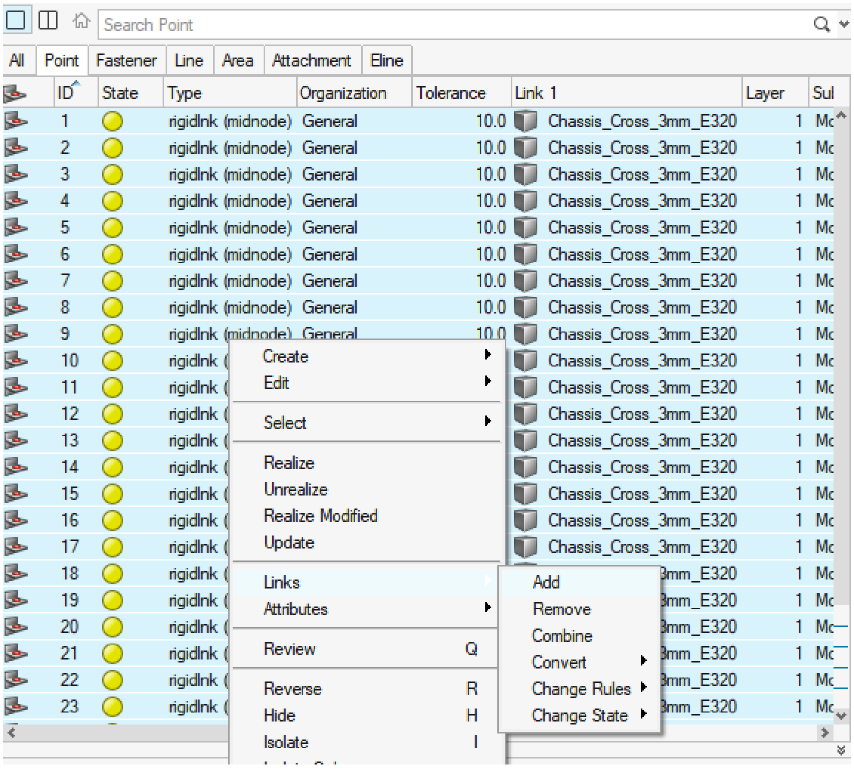


The connectors should be created. The best overview is available in connector browser



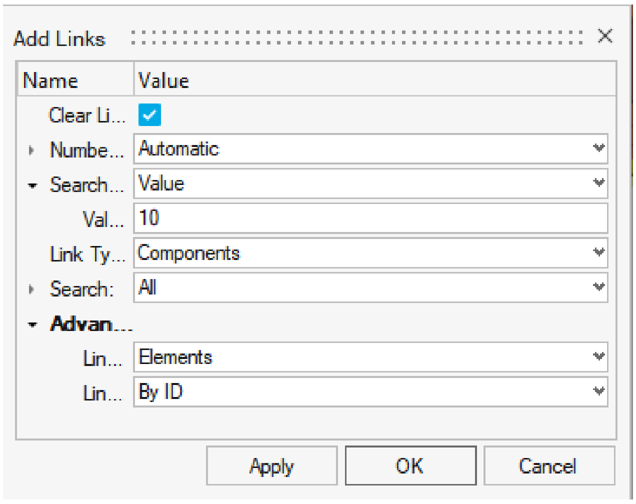
	ID	State	Type	Organization	Tolerance	Link 1	Layer	Sul
	1	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	2	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	3	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	4	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	5	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	6	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	7	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	8	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	9	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	10	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	11	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	12	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	13	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	14	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	15	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	16	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	17	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	18	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	19	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	20	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	21	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	22	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
	23	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc

STEP4: Connector settings update. Select newly created connectors. Right mouse button, select Links and Add.



ID	State	Type	Organization	Tolerance	Link 1	Layer	Sul
1	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
2	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
3	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
4	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
5	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
6	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
7	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
8	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
9	●	rigidlnk (midnode)	General	10.0	Chassis_Cross_3mm_E320	1	Mc
10	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
11	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
12	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
13	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
14	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
15	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
16	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
17	●	rigidlnk (			Chassis_Cross_3mm_E320	1	Mc
18	●	rigidlnk (			mm_E320	1	Mc
19	●	rigidlnk (			mm_E320	1	Mc
20	●	rigidlnk (			mm_E320	1	Mc
21	●	rigidlnk (			3mm_E320	1	Mc
22	●	rigidlnk (			3mm_E320	1	Mc
23	●	rigidlnk (			3mm_E320	1	Mc

Fullfill the table as follows:



Name	Value
Clear Li...	☑
Numbe...	Automatic
Search...	Value
Val...	10
Link Ty...	Components
Search:	All
Advan...	
Lin...	Elements
Lin...	By ID

STEP6: Confirm by Apply + OK. Export/save an include

KNOWN LIMITATION: Connectors do not require precise pitch distance between connected include files. In general connectors allows relatively large deviations of nodes position. However, it is necessary to avoid the situation where a node position in one of the includes lies between nodes in the opposite include file.

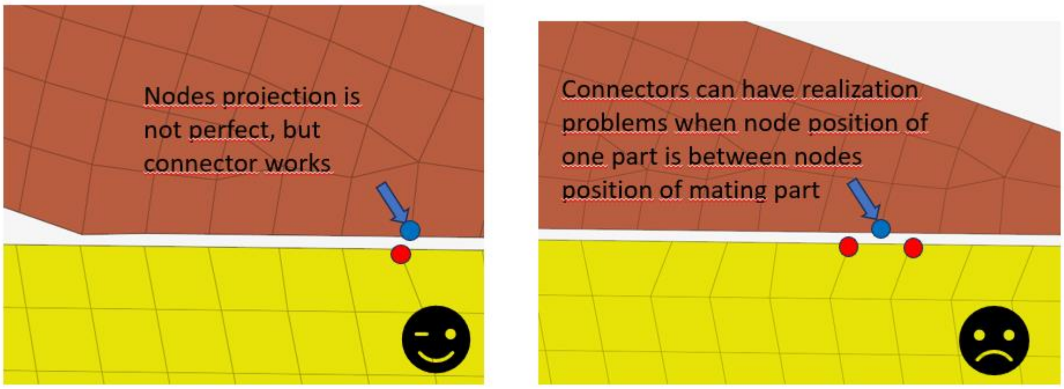

Nodes projection is not perfect, but connector works
Connectors can have realization problems when node position of one part is between nodes position of mating part

Fig. Known limitation for spotweld connector

2B. Connector Spring for Radioss

The spring connector is proposed for joint connections. Typical example is connection between front Axle and Chassis. Connector itself is going to realize only spring element connecting prepared rigid elements. It means user has to prepare two rigid bodies in two different include files. Each rigid body has to have one free end. Nodes of both free ends must have identical position, because the final spring should have zero length.

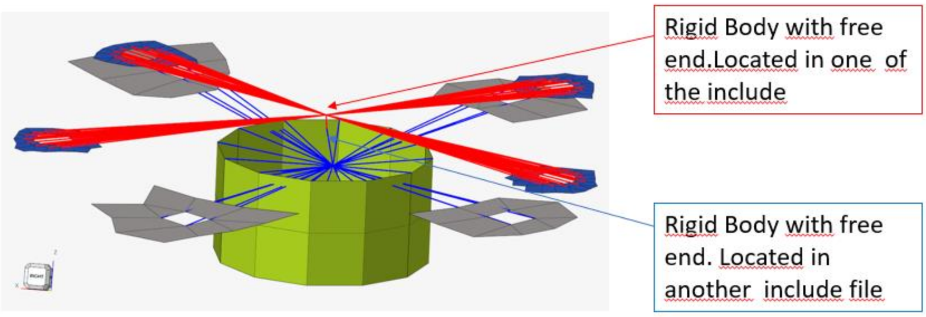
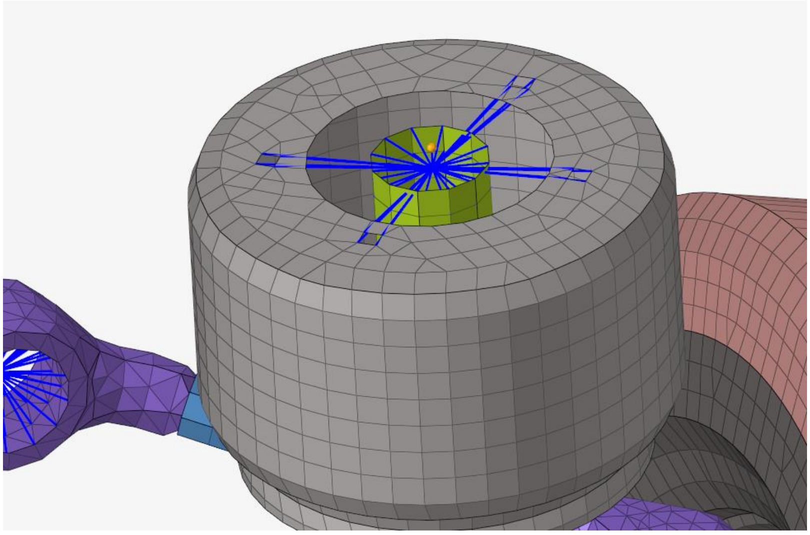
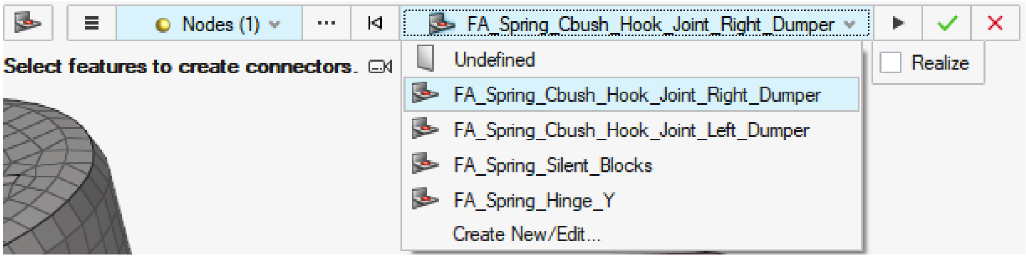

Rigid Body with free end.Located in one of the include
Rigid Body with free end. Located in another include file

Fig. Connection place suitable for spring connection

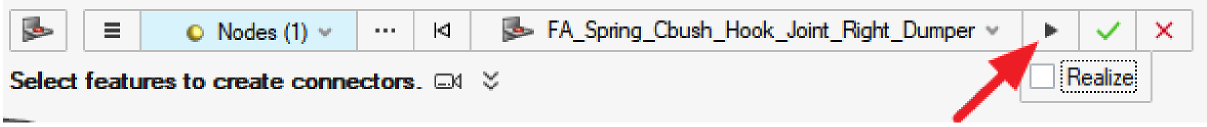
STEP1: Make a sure that you have correct include file set as current include. This will place connector into desired include. You can also create connector when only one include file is loaded in Hypermesh. Connection to entites from another include is realized automatically later. Now select the nodes at desired weld location.



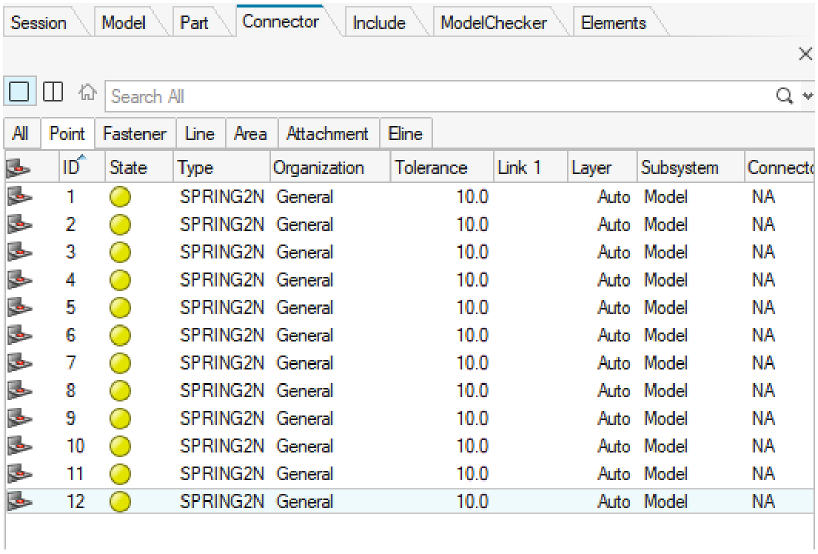
STEP2: Select Point connector in Connector pull-down menu. The point creation feature is taking over your node selection. Select “FA_Spring_Cbush_Hook_Joint_Right_Dumper” connector control. Connector control is available as file Radioss_Joints.hm in Accessories folder.


Nodes (1)
FA_Spring_Cbush_Hook_Joint_Right_Dumper
Select features to create connectors.
Undefined
FA_Spring_Cbush_Hook_Joint_Right_Dumper
FA_Spring_Cbush_Hook_Joint_Left_Dumper
FA_Spring_Silent_Blocks
FA_Spring_Hinge_Y
Create New/Edit...
□ Realize

STEP3: Confirm connector realization by click on run button.

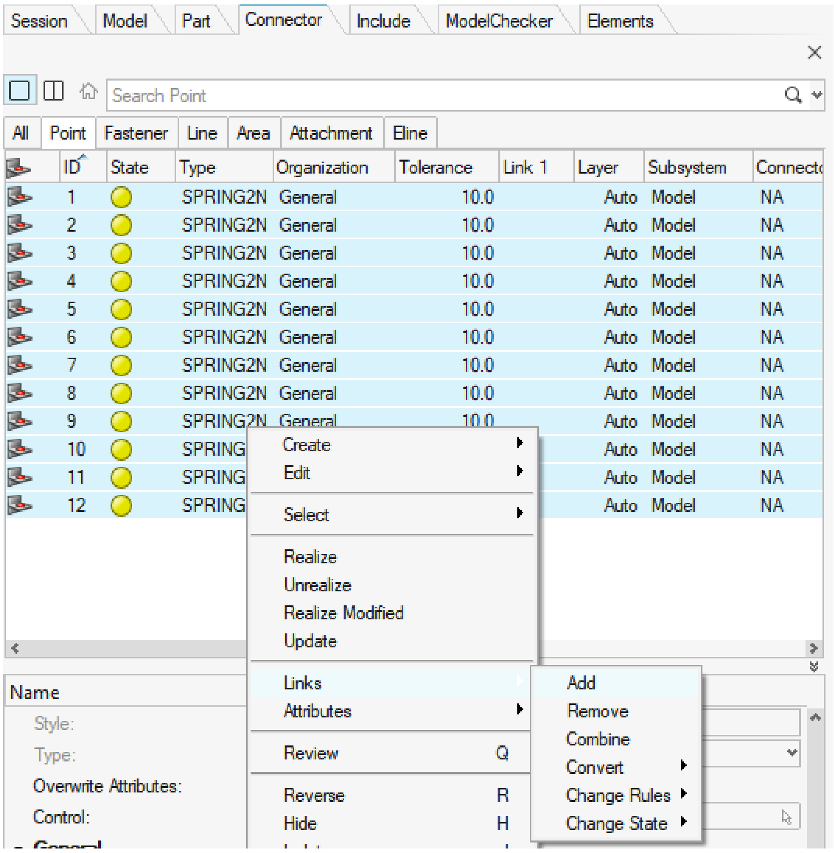

Nodes (1)
FA_Spring_Cbush_Hook_Joint_Right_Dumper
Select features to create connectors.
□ Realize

The connectors should be created. The best overview is available by connector browser



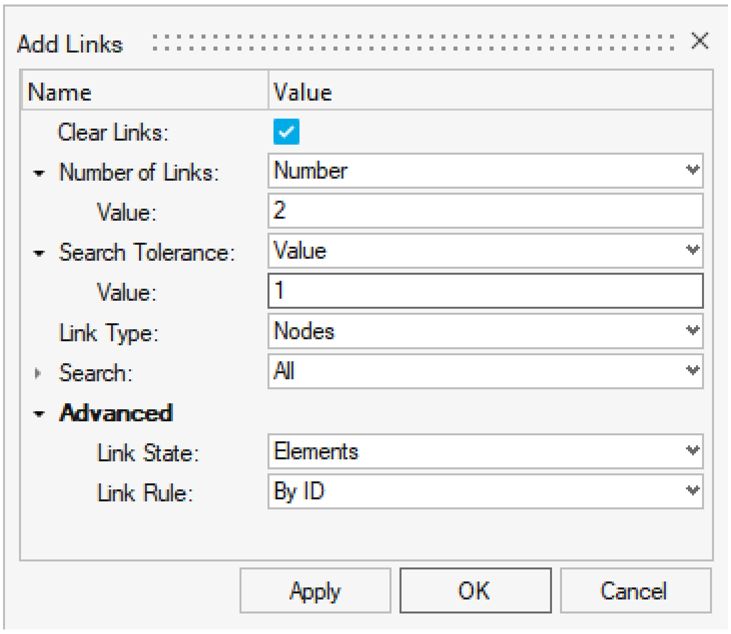
	ID	State	Type	Organization	Tolerance	Link 1	Layer	Subsystem	Connecto
	1	●	SPRING2N	General	10.0		Auto	Model	NA
	2	●	SPRING2N	General	10.0		Auto	Model	NA
	3	●	SPRING2N	General	10.0		Auto	Model	NA
	4	●	SPRING2N	General	10.0		Auto	Model	NA
	5	●	SPRING2N	General	10.0		Auto	Model	NA
	6	●	SPRING2N	General	10.0		Auto	Model	NA
	7	●	SPRING2N	General	10.0		Auto	Model	NA
	8	●	SPRING2N	General	10.0		Auto	Model	NA
	9	●	SPRING2N	General	10.0		Auto	Model	NA
	10	●	SPRING2N	General	10.0		Auto	Model	NA
	11	●	SPRING2N	General	10.0		Auto	Model	NA
	12	●	SPRING2N	General	10.0		Auto	Model	NA

STEP4: Connector settings update. Select newly created connectors. Right mouse button, select Links and Add.



	ID	State	Type	Organization	Tolerance	Link 1	Layer	Subsystem	Connecto
	1	●	SPRING2N	General	10.0		Auto	Model	NA
	2	●	SPRING2N	General	10.0		Auto	Model	NA
	3	●	SPRING2N	General	10.0		Auto	Model	NA
	4	●	SPRING2N	General	10.0		Auto	Model	NA
	5	●	SPRING2N	General	10.0		Auto	Model	NA
	6	●	SPRING2N	General	10.0		Auto	Model	NA
	7	●	SPRING2N	General	10.0		Auto	Model	NA
	8	●	SPRING2N	General	10.0		Auto	Model	NA
	9	●	SPRING2N	General	10.0		Auto	Model	NA
	10	●	SPRING				Auto	Model	NA
	11	●	SPRING				Auto	Model	NA
	12	●	SPRING				Auto	Model	NA

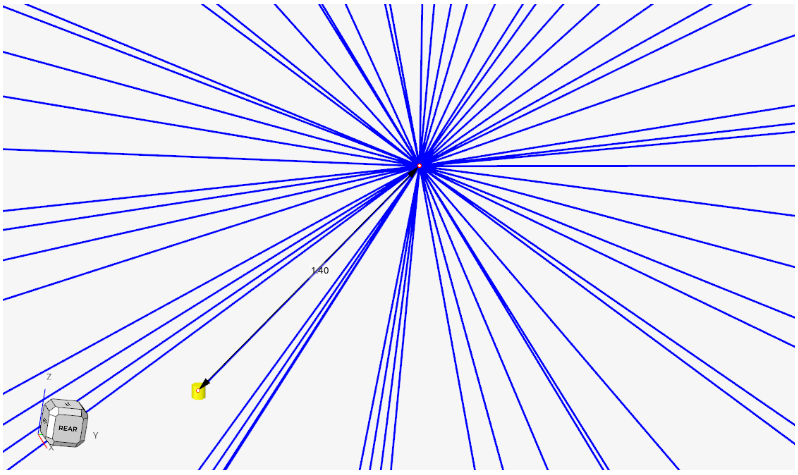
Fullfill the table as follows:



Add Links

Name	Value
Clear Links:	☑
▾ Number of Links:	Number
Value:	2
▾ Search Tolerance:	Value
Value:	1
Link Type:	Nodes
▸ Search:	All
▾ **Advanced**	
Link State:	Elements
Link Rule:	By ID

Note: Search tolerance value must be lower than distance between master and slave node in rigid body. For example, distance between the master node and secondary node is 1.4mm as shown below. Search tolerances is setup as 1.0mm.



STEP6: Confirm by Apply + OK. Export/save an include

2C. Connector Rigid Spotweld for Optistruct

STEP1: Make a sure that you have correct include file set as current include. This will place connector into desired include. You can also create connector when only one include file is loaded in Hypermesh. Connection to entites from another include is realized automatically later. Now select the nodes at desired weld location.

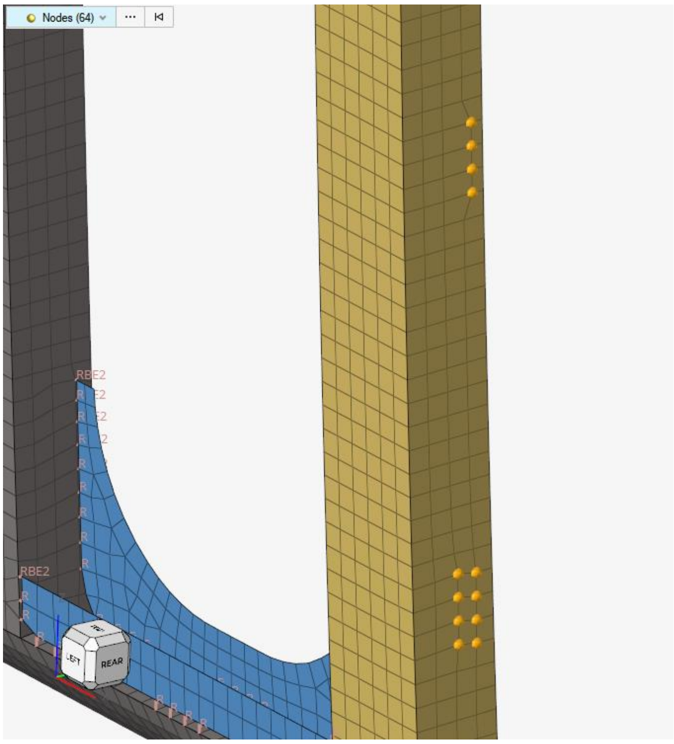
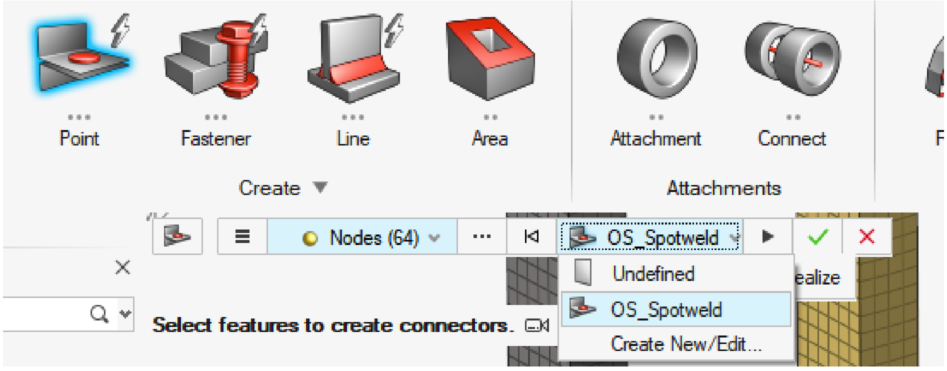


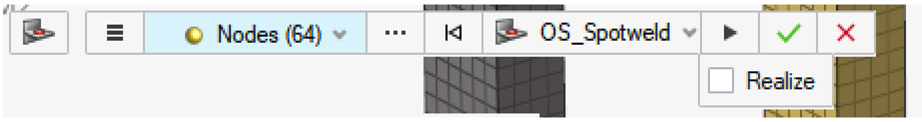
Fig. Connection place suitable for connector application

STEP2: Select Point connector in Connector pull-down menu. The point creation feature is taking over your node selection. Select “OS_spotweld” connector control. Connector control is available as file OS_spotweld.hm in Accessories folder.

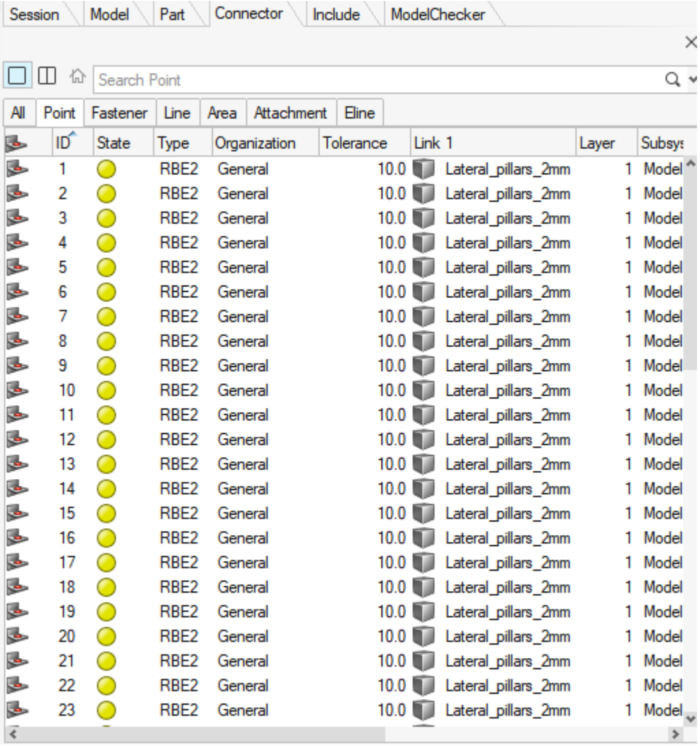


Select features to create connectors.

STEP3: Confirm connector realization by click on run button.

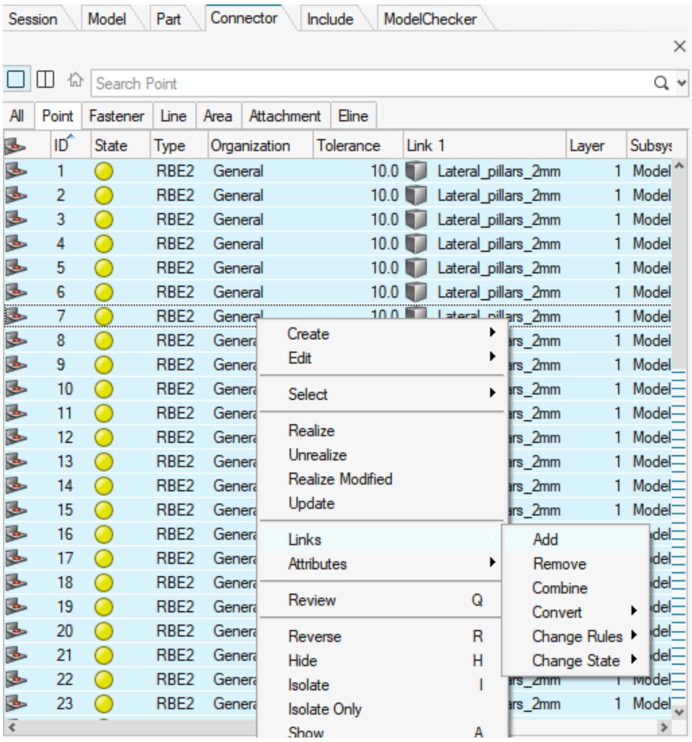


The connectors should be created. The best overview is available by connector browser



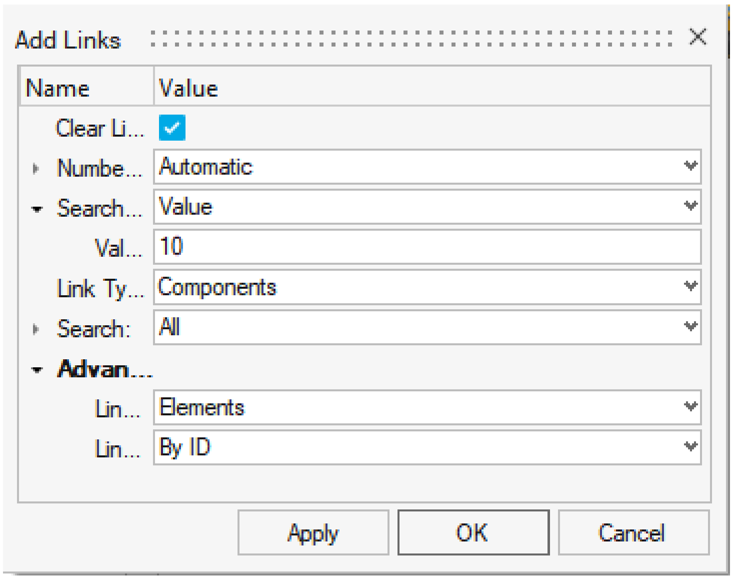
ID	State	Type	Organization	Tolerance	Link 1	Layer	Subsys
1	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
2	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
3	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
4	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
5	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
6	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
7	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
8	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
9	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
10	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
11	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
12	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
13	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
14	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
15	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
16	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
17	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
18	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
19	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
20	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
21	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
22	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
23	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model

STEP4: Connector settings update. Select newly created connectors. Right mouse button, select Links and Add.



ID	State	Type	Organization	Tolerance	Link 1	Layer	Subsys
1	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
2	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
3	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
4	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
5	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
6	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
7	●	RBE2	General	10.0	Lateral_pillars_2mm	1	Model
8	●	RBE2	Genera		rs_2mm	1	Model
9	●	RBE2	Genera		rs_2mm	1	Model
10	●	RBE2	Genera		rs_2mm	1	Model
11	●	RBE2	Genera		rs_2mm	1	Model
12	●	RBE2	Genera		rs_2mm	1	Model
13	●	RBE2	Genera		rs_2mm	1	Model
14	●	RBE2	Genera		rs_2mm	1	Model
15	●	RBE2	Genera		rs_2mm	1	Model
16	●	RBE2	Genera				del
17	●	RBE2	Genera				del
18	●	RBE2	Genera				del
19	●	RBE2	Genera				del
20	●	RBE2	Genera				del
21	●	RBE2	Genera				del
22	●	RBE2	Genera		rs_2mm	1	Model
23	●	RBE2	Genera		rs_2mm	1	Model

Fullfill the table as follows:



Add Links

Name	Value
Clear Li...	☑
Numbe...	Automatic
Search...	Value
Val...	10
Link Ty...	Components
Search:	All
Advan...	
Lin...	Elements
Lin...	By ID

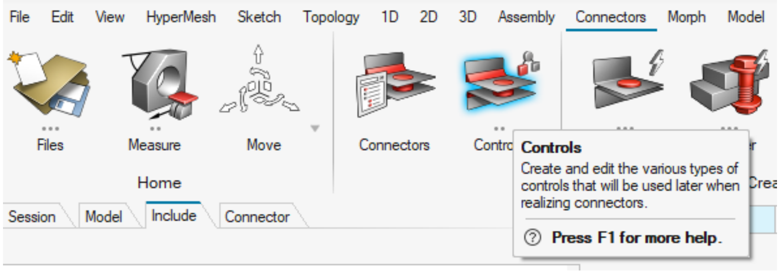
STEP6: Confirm by Apply + OK. Export/save an include

2D. Connector Control

The above procedures for creating connectors use a total of three sets of settings, called connector controls.

This chapter describes how to create and edit connector controls, because modification is essential, especially for spring-type connections.

STEP1: Open HyperMesh v2023.1 or higher and make sure that correct user profile is selected. Go to Connectors menu and select Controls ribbon button.

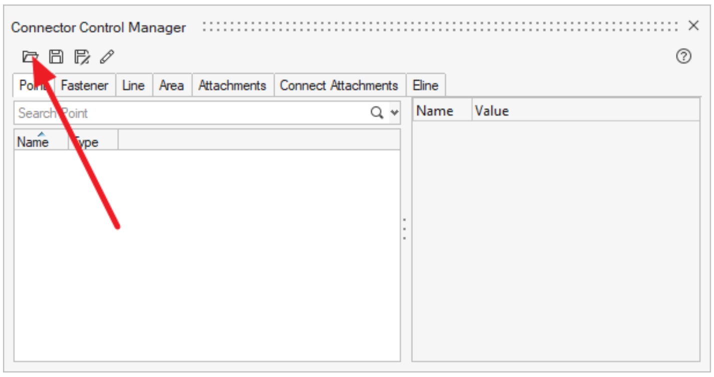


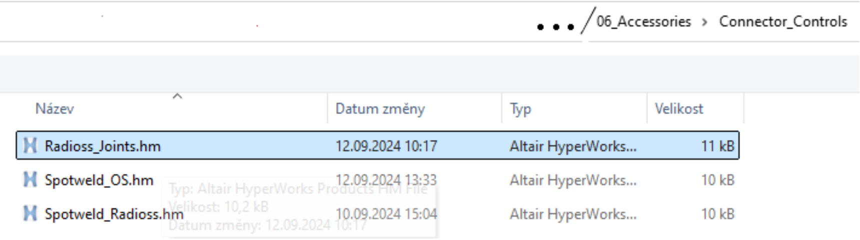
Controls

Create and edit the various types of controls that will be used later when realizing connectors.

Press F1 for more help.

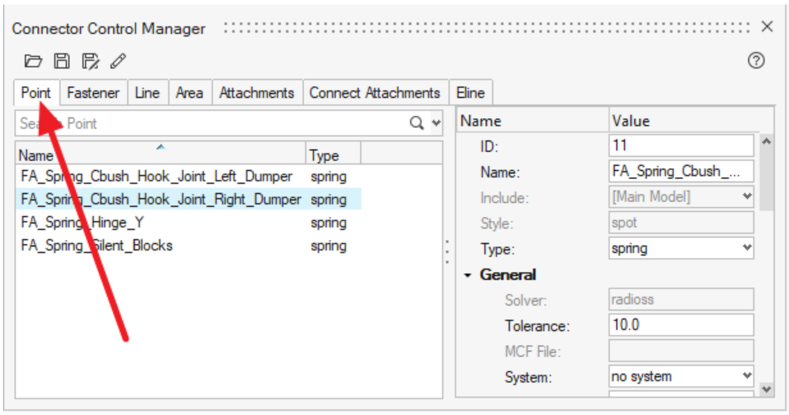
STEP2: The Connector Control Manager panel will pop up. To edit existing connector control, for example Radioss spring, open relevant hm file Radioss_Joints.hm in accessories folder.




• • • / 06_Accessories › Connector_Controls
Typ: Altair HyperWorks Products HM File
Velikost: 10,2 kB
Datum změny: 12.09.2024 10:17

Název	Datum změny	Typ	Velikost
Radioss_Joints.hm	12.09.2024 10:17	Altair HyperWorks...	11 kB
Spotweld_OS.hm	12.09.2024 13:33	Altair HyperWorks...	10 kB
Spotweld_Radioss.hm	10.09.2024 15:04	Altair HyperWorks...	10 kB

STEP3: Go again to the connectors menu and select Controls ribbon button. The connector Control Manager panel will pop up. Make a sure that Point connectors tab is selected.

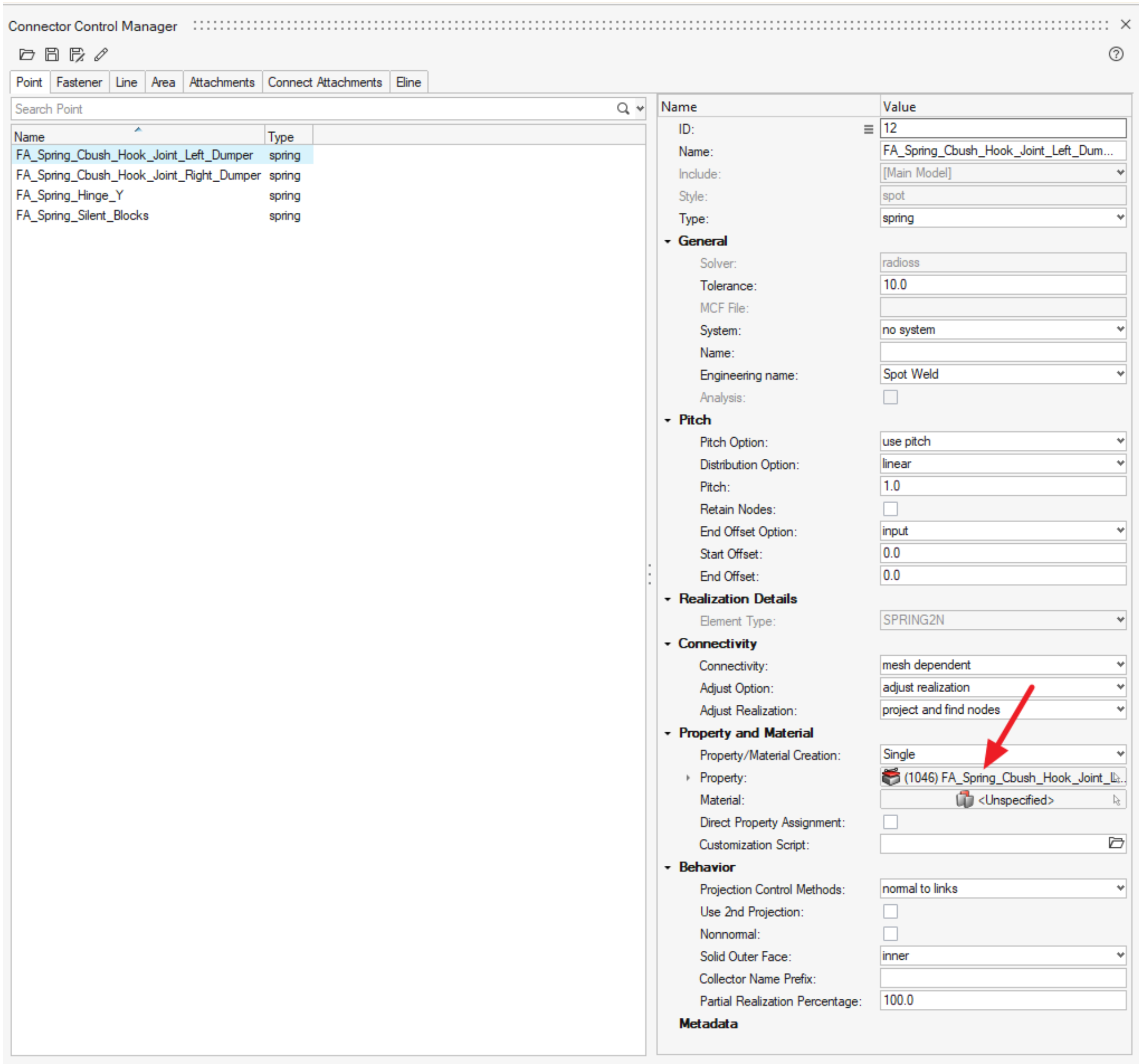


Name	Type
FA_Spring_Cbush_Hook_Joint_Left_Dumper	spring
FA_Spring_Cbush_Hook_Joint_Right_Dumper	spring
FA_Spring_Hinge_Y	spring
FA_Spring_Silent_Blocks	spring

Name	Value
ID:	11
Name:	FA_Spring_Cbush_...
Include:	[Main Model]
Style:	spot
Type:	spring
General	
Solver:	radioss
Tolerance:	10.0
MCF File:	
System:	no system

Notice that 4 connector controls were imported.

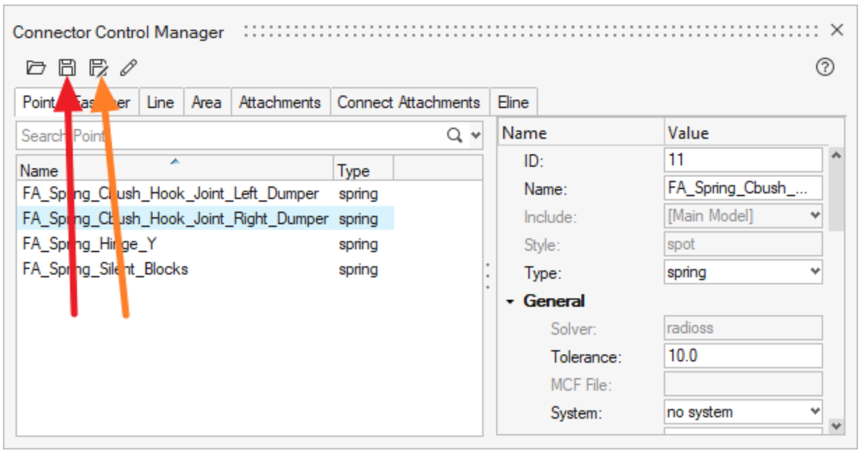
STEP4: Select one of the connector control. Right side of manager shows editable settings. The most important settings for spring connection are property assignment. Property must be presented in include files containing connecting structure.



Name	Type
FA_Spring_Cbush_Hook_Joint_Left_Dumper	spring
FA_Spring_Cbush_Hook_Joint_Right_Dumper	spring
FA_Spring_Hinge_Y	spring
FA_Spring_Silent_Blocks	spring

Name	Value
ID:	12
Name:	FA_Spring_Cbush_Hook_Joint_Left_Dum...
Include:	[Main Model]
Style:	spot
Type:	spring
General	
Solver:	radioss
Tolerance:	10.0
MCF File:	
System:	no system
Name:	
Engineering name:	Spot Weld
Analysis:	□
Pitch	
Pitch Option:	use pitch
Distribution Option:	linear
Pitch:	1.0
Retain Nodes:	□
End Offset Option:	input
Start Offset:	0.0
End Offset:	0.0
Realization Details	
Element Type:	SPRING2N
Connectivity	
Connectivity:	mesh dependent
Adjust Option:	adjust realization
Adjust Realization:	project and find nodes
Property and Material	
Property/Material Creation:	Single
Property:	(1046) FA_Spring_Cbush_Hook_Joint_L...
Material:	<Unspecified>
Direct Property Assignment:	□
Customization Script:	
Behavior	
Projection Control Methods:	normal to links
Use 2nd Projection:	□
Nonnormal:	□
Solid Outer Face:	inner
Collector Name Prefix:	
Partial Realization Percentage:	100.0
Metadata	

STEP5: Save or Save as your connector control



Name	Type
FA_Spring_Cbush_Hook_Joint_Left_Dumper	spring
FA_Spring_Cbush_Hook_Joint_Right_Dumper	spring
FA_Spring_Hinge_Y	spring
FA_Spring_Silent_Blocks	spring

Name	Value
ID:	11
Name:	FA_Spring_Cbush_...
Include:	[Main Model]
Style:	spot
Type:	spring
General	
Solver:	radioss
Tolerance:	10.0
MCF File:	
System:	no system

HyperMesh Extension

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

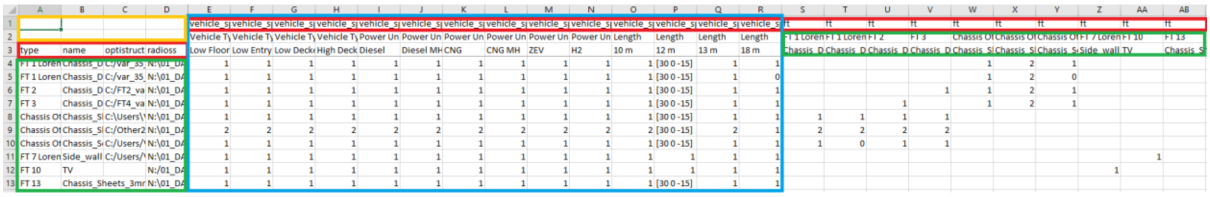
The current trend in HyperMesh customization is the increasing use of Extensions, which are modular add-ons that extend the core functionalities of the software. Extensions can be developed to provide new capabilities, automate complex processes, and integrate HyperMesh more seamlessly with other tools 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:
    - 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:



	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_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	vehicle_s	ft	ft	ft	ft	ft	ft	ft	ft	ft	ft
2					Vehicle Ty	Vehicle Ty	Vehicle Ty	Vehicle Ty	Power Un	Power Un	Power Un	Power Un	Power Un	Power Un	Length	Length	Length	Length	FT 1 Loren	FT 1 Loren	FT 2	FT 3	Chassis Of	Chassis Of	Chassis Of	FT 7 Loren	FT 10	FT 13
3	type	name	optistruct	radioss	Low Floor	Low Entry	Low Deck	High Deck	Diesel	Diesel MH	CNG	CNG MH	ZEV	H2	10 m	12 m	13 m	18 m	Chassis_D	Chassis_D	Chassis_D	Chassis_D	Chassis_S	Chassis_S	Chassis_S	Side_wall	TV	Chassis_S
4	FT 1 Loren	Chassis_D	C:/var_35_	N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	1					1	2	1			
5	FT 1 Loren	Chassis_D	C:/var_35_	N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	0					1	2	0			
6	FT 2	Chassis_D	C:/FT2_va	N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	1				1	1	2	1			
7	FT 3	Chassis_D	C:/FT4_va	N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	1			1		1	2	1			
8	Chassis Of	Chassis_Sl	C:\Users\	N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	1	1	1	1	1						
9	Chassis Of	Chassis_Sl	C:/Other2	N:\01_DA	2	2	2	2	2	2	2	2	2	2	2	[30 0 -15]	2	1	2	2	2	2						
10	Chassis Of	Chassis_S	C:/Users/	N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	1	1	0	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	
12	FT 10	TV		N:/01_DA	1	1	1	1	1	1	1	1	1	1	1	1	1	1								1		
13	FT 13	Chassis_Sheets_3mr		N:\01_DA	1	1	1	1	1	1	1	1	1	1	1	[30 0 -15]	1	1										

🟨 Leave empty

🟥 Leave keywords and layout

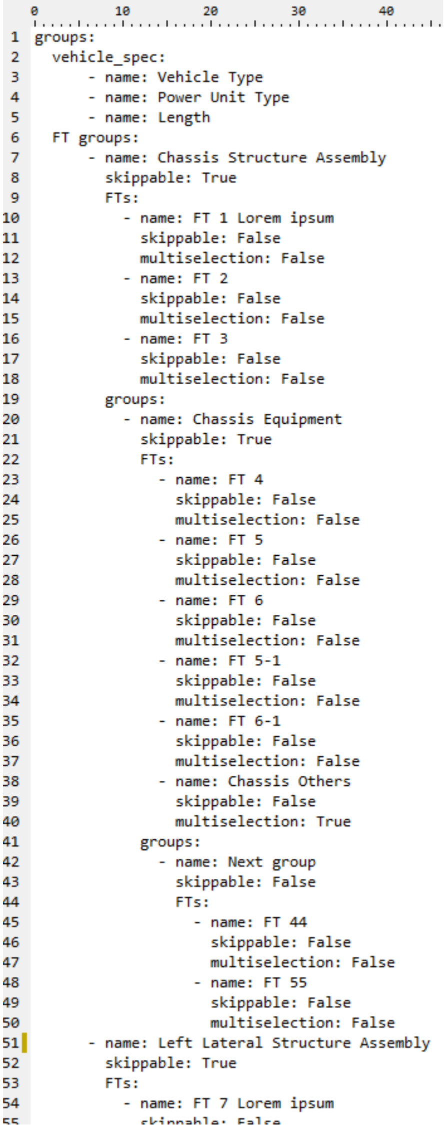
🟩 Definition of includes in database

🟦 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.

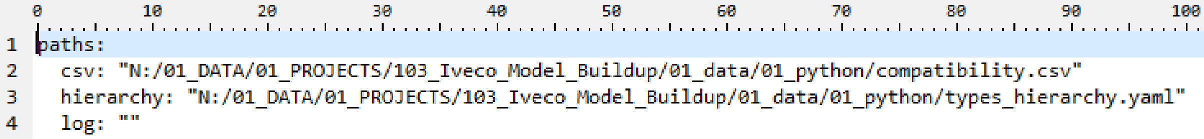


```yaml
groups:
  vehicle_spec:
      - name: Vehicle Type
      - name: Power Unit Type
      - name: Length
  FT groups:
      - name: Chassis Structure Assembly
        skippable: True
        FTs:
          - name: FT 1 Lorem ipsum
            skippable: False
            multiselection: False
          - 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
```

- 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

Installation

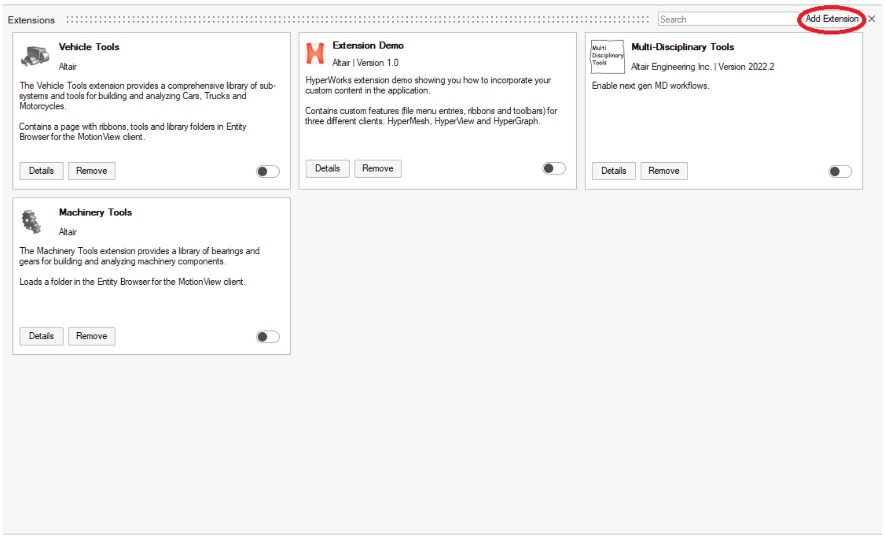
1) Copy main folder of extension where you want to store it
2) Open config.yaml in main folder and edit paths to compatibility csv and hierarchy yaml.



```
paths:
  csv: "N:/01_DATA/01_PROJECTS/103_Iveco_Model_Buildup/01_data/01_python/compatibility.csv"
  hierarchy: "N:/01_DATA/01_PROJECTS/103_Iveco_Model_Buildup/01_data/01_python/types_hierarchy.yaml"
  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



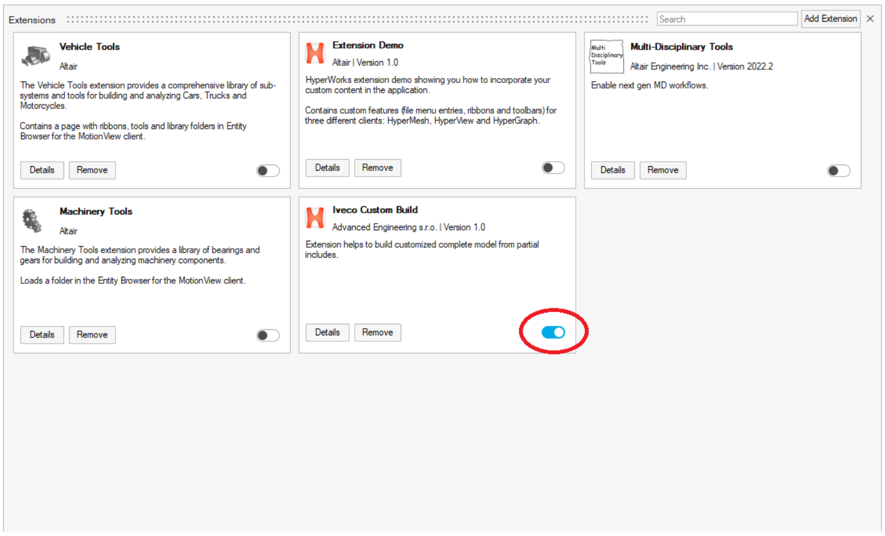
Vehicle Tools
Altair
The Vehicle Tools extension provides a comprehensive library of sub-systems and tools for building and analyzing Cars, Trucks and Motorcycles.
Contains a page with ribbons, tools and library folders in Entity Browser for the MotionView client.

Extension Demo
Altair | Version 1.0
HyperWorks extension demo showing you how to incorporate your custom content in the application.
Contains custom features (file menu entries, ribbons and toolbars) for three different clients: HyperMesh, HyperView and HyperGraph.

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

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.

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



Vehicle Tools
Altair
The Vehicle Tools extension provides a comprehensive library of sub-systems and tools for building and analyzing Cars, Trucks and Motorcycles.
Contains a page with ribbons, tools and library folders in Entity Browser for the MotionView client.

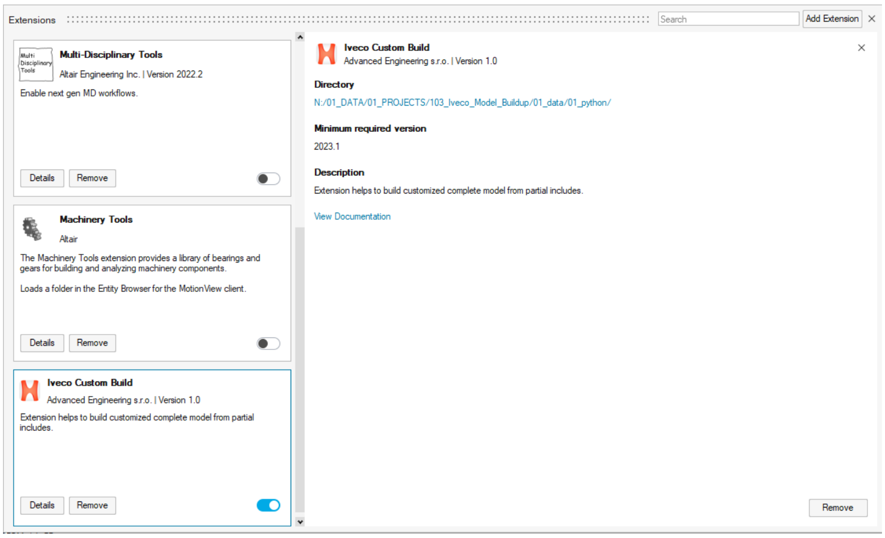
Extension Demo
Altair | Version 1.0
HyperWorks extension demo showing you how to incorporate your custom content in the application.
Contains custom features (file menu entries, ribbons and toolbars) for three different clients: HyperMesh, HyperView and HyperGraph.

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

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.

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

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



Multi-Disciplinary Tools

Altair Engineering Inc. | Version 2022.2

Enable next gen MD workflows.

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.

Iveco Custom Build

Advanced Engineering s.r.o. | Version 1.0

Extension helps to build customized complete model from partial includes.

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

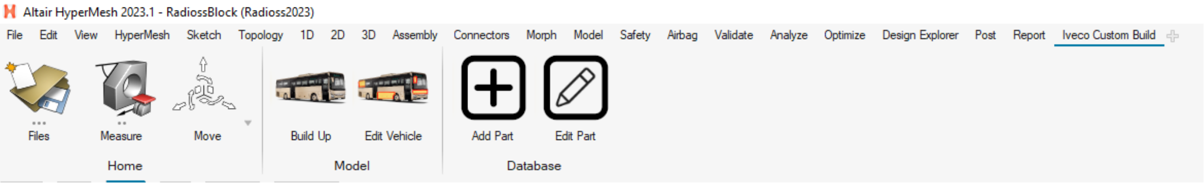
2023.1

Description

Extension helps to build customized complete model from partial includes.

Using extension

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

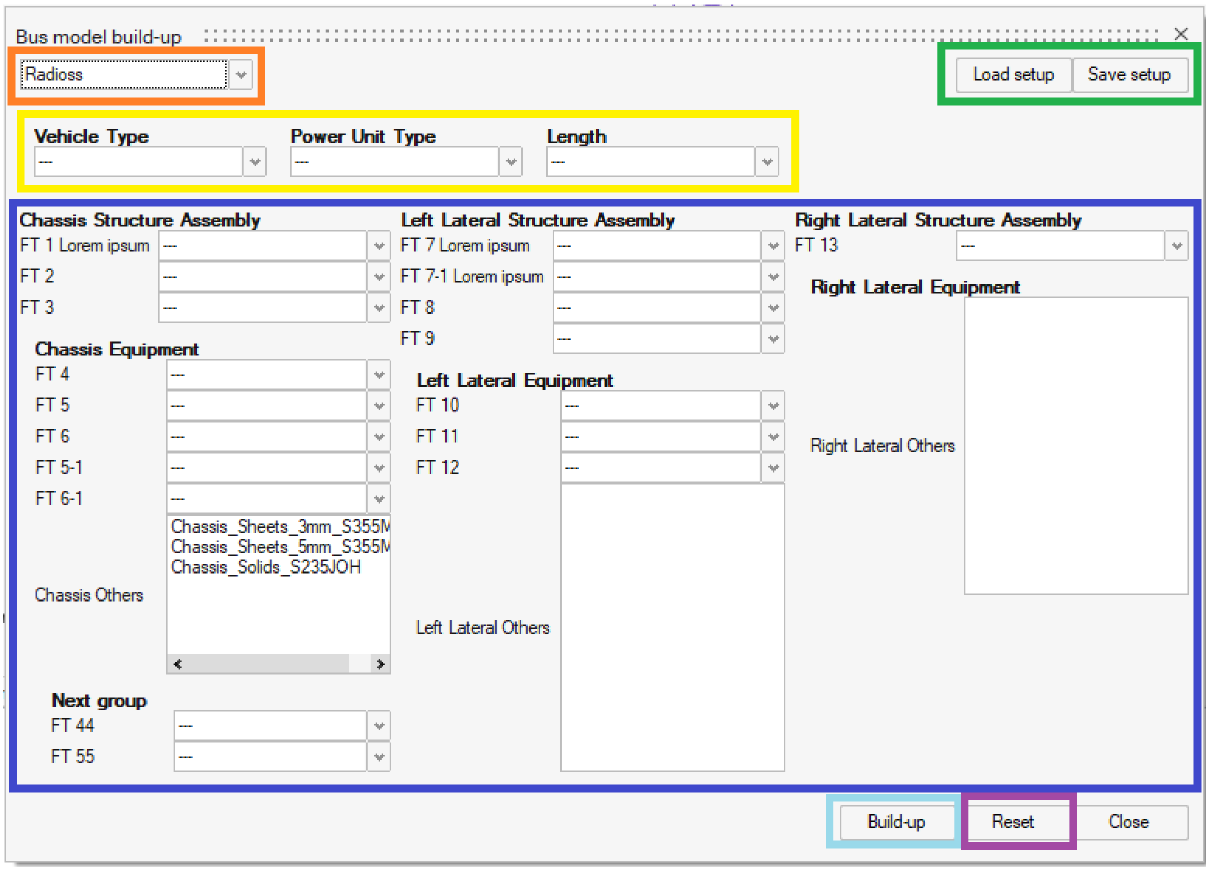

Altair HyperMesh 2023.1 - RadiossBlock (Radioss2023)
File Edit View HyperMesh Sketch Topology 1D 2D 3D Assembly Connectors Morph Model Safety Airbag Validate Analyze Optimize Design Explorer Post Report Iveco Custom Build
Files
Measure
Move
Home
Build Up
Edit Vehicle
Model
Add Part
Edit Part
Database

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

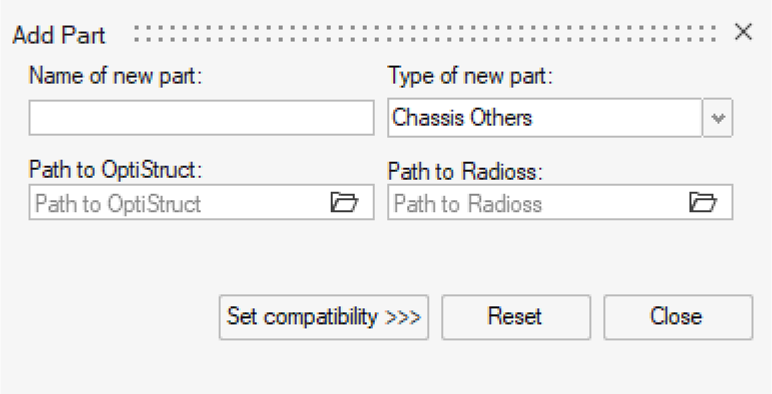
Serves for editing currently loaded model of vehicle. It loads all current includes into setup table and user can deselect them to remove it, change the selection to replace them and select new to add new.

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

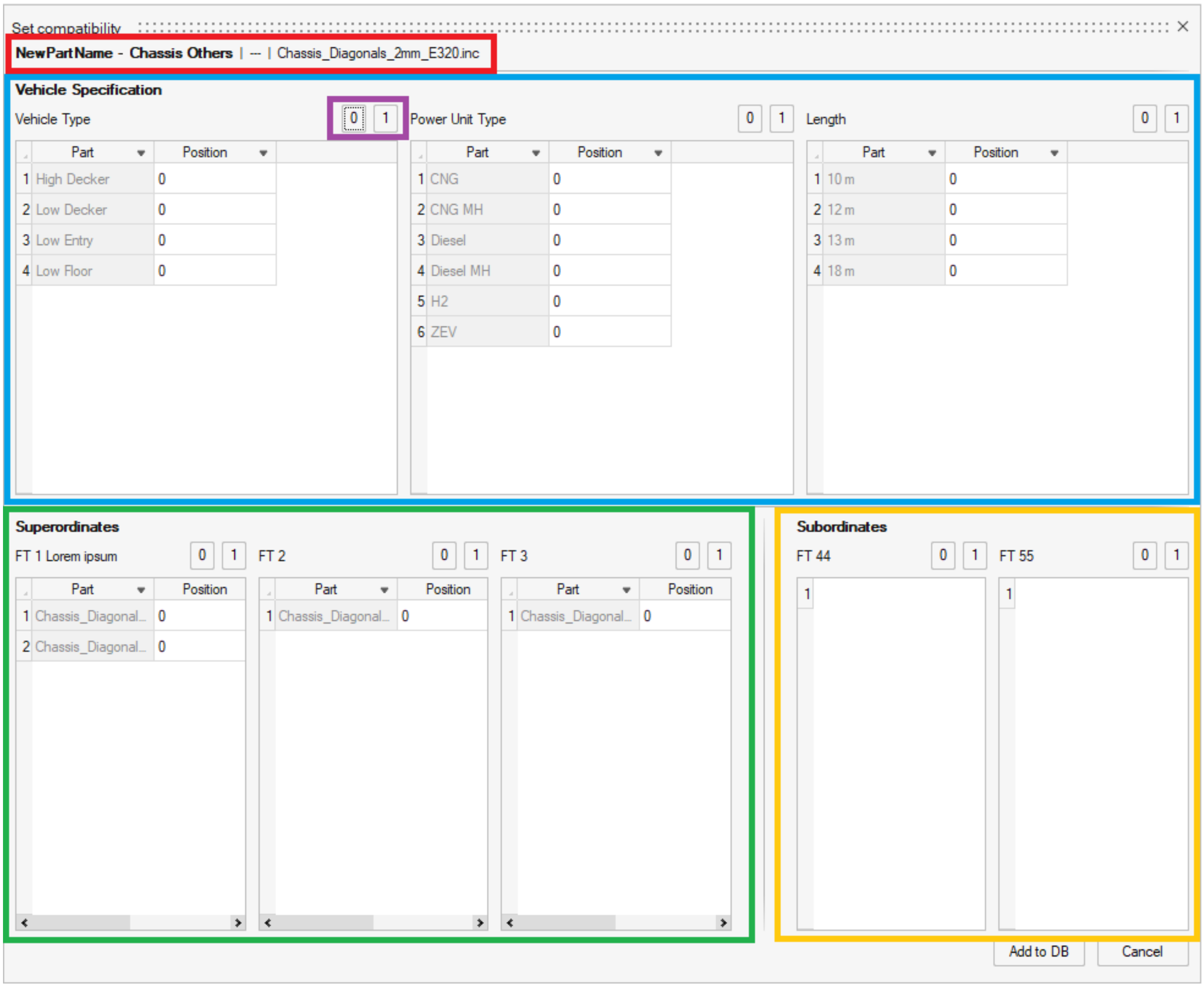
Name of new part:

Type of new part: Chassis Others

Path to OptiStruct: Path to OptiStruct

Path to Radioss: Path to Radioss

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



NewPartName - Chassis Others | --- | Chassis_Diagonals_2mm_E320.inc

Vehicle Specification

Vehicle Type

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

Power Unit Type

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

Length

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

Superordinates

FT 1 Lorem ipsum

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

FT 2

	Part	Position
1	Chassis_Diagonal...	0

FT 3

	Part	Position
1	Chassis_Diagonal...	0

Subordinates

FT 44

1

FT 55

1

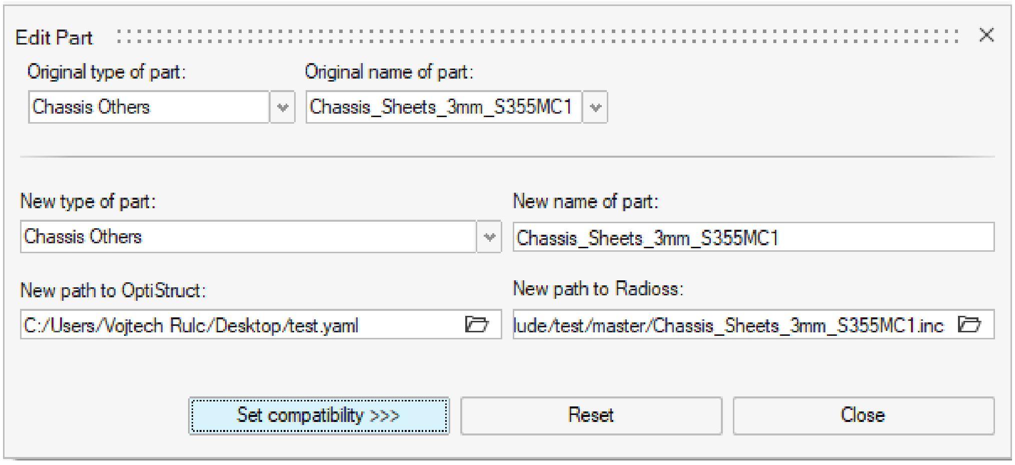
🟥	Information about currently being added part for check. Name – Type | OS file | Radios file
🟦	Compatibility with Vehicle Specification
🟩	Compatibility with superordinant FTs
🟨	Compatibility with subordinant FTs
🟪	These buttons quickly assign selected value to all parts in the FT

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

Original type of part: Chassis Others

Original name of part: Chassis_Sheets_3mm_S355MC1

New type of part: Chassis Others

New name of part: Chassis_Sheets_3mm_S355MC1

New path to OptiStruct: C:/Users/Vojtech Rulc/Desktop/test.yaml

New path to Radioss: lude/test/master/Chassis_Sheets_3mm_S355MC1.inc

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