

Chassis Suspension Simulation (CH9)

BPA Delivery 6 for V5R19 (V5.6)

User Guide

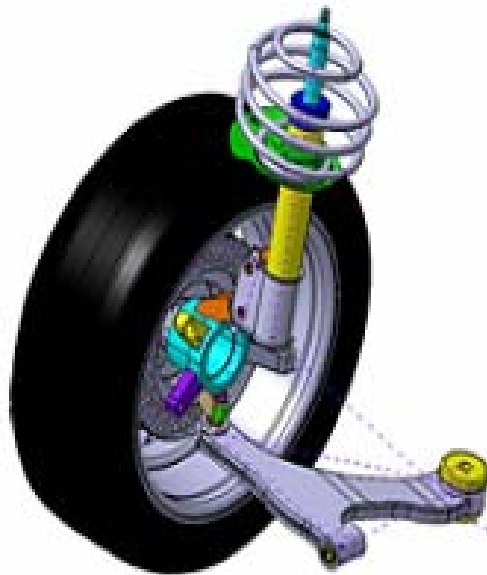


Table of Contents

<i>Chassis Suspension Simulation (CH9)</i>	<i>1</i>
<i>BPA Delivery 6 for V5R19 (V5.6)</i>	<i>1</i>
User Guide	1
Table of Contents	2
Introduction.....	3
Overview of Chassis Suspension Simulation.....	4
1.1 Workbench contents	4
Merge Structure	5
2.1 Purpose.....	5
2.2 Pre-requisite.....	5
2.3 User Operations.....	7
Merge Kinematics	8
3.1 Purpose.....	8
3.1.1 Scenario 1- Create A Common Product	9
3.1.2 Scenario 2- Do Not Create A Common Product	10
3.2 Pre-requisite.....	10
3.3 User Operations.....	11
3.3.1 Scenario 1- Create A Common Product	11
3.3.2 Scenario 2- Do Not Create A Common Product	14
Automatic Dress-up	16
4.1 Purpose.....	16
4.2 Pre-requisite.....	16
4.3 User Operations.....	18

Introduction

“Chassis Suspension Simulation” is an add-in workbench provided over CATIA V5, where advance functionalities which are helpful for suspension study process are provided. It consists of a dedicated workbench containing required commands and functionalities from other existing workbenches as well as newly developed commands specific to this domain.

The purpose of this document is to help readers understand how to use the BPA Chassis Suspension Simulation.

Overview of Chassis Suspension Simulation

This chapter introduces the different toolbars/ functionalities included into the Chassis Suspension Simulation workbench.

1.1 Workbench contents

This workbench is containing all necessary commands required for the development of Chassis Suspension Simulation.

Three tool-bars:

- 1) **CSS Toolbar:** - It contains the specific commands developed for the chassis suspension design. It consists of three commands



- a. Merge Structure
- b. Merge Kinematics
- c. Automatic Dressup

- 2) **DMU Toolbar:** - It contains all necessary commands from the DMU Kinematics workbench.



- 3) **Catalog Browser:** - This toolbar contains Catalog Browser



Merge Structure

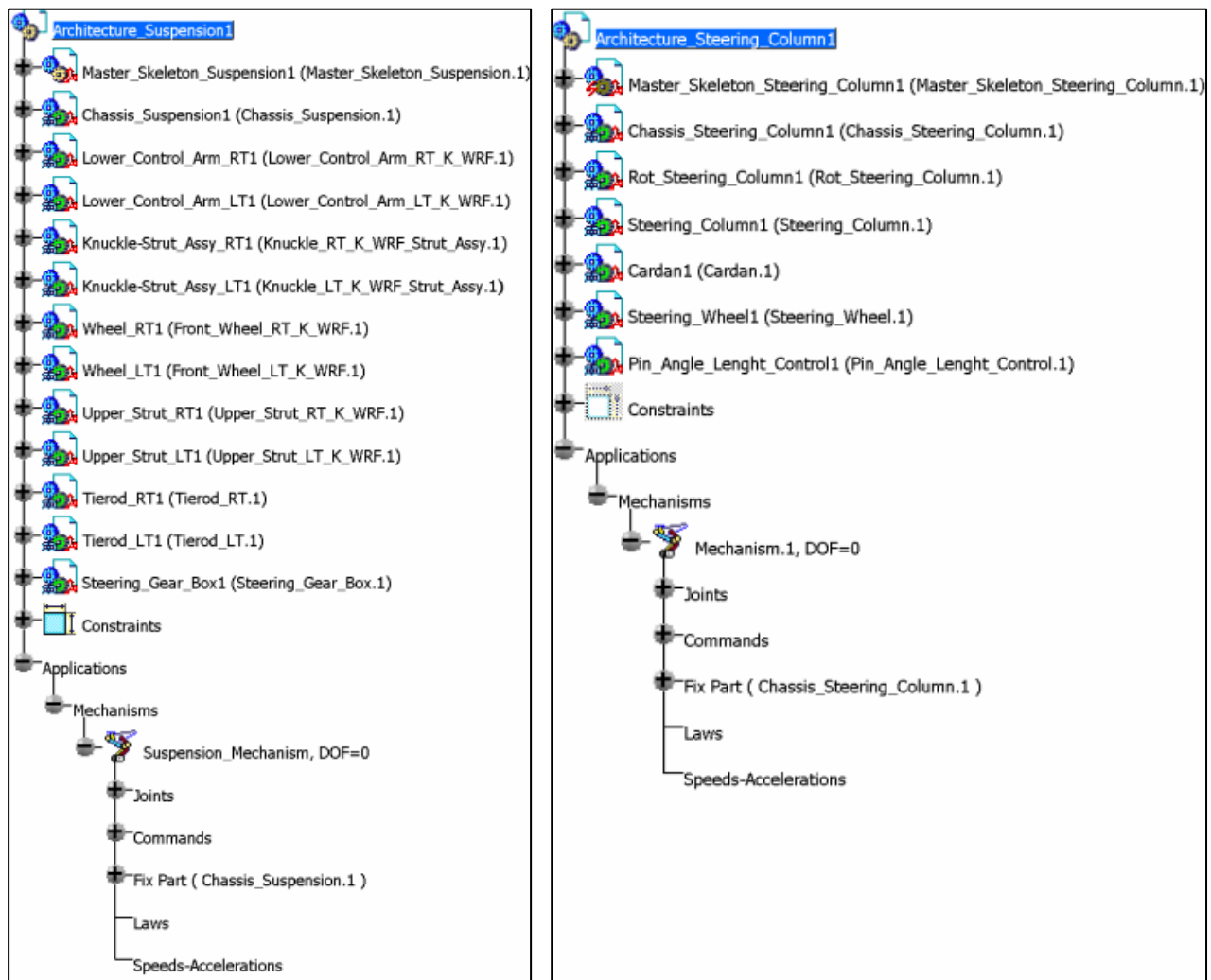
2.1 Purpose

This command merges two product structures.

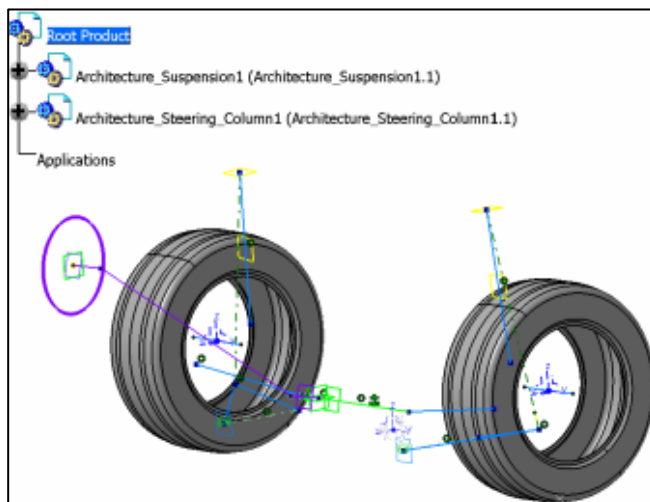
All the parts under both architectures (In the example below, Architecture Suspension and Architecture Steering Column) will get Copy-and-Pasted under the Root Product with Break link format and then the contextual links will be restored.

2.2 Pre-requisite

- 1) Two Architectures with mechanisms related to them.



- 2) Open these two architectures under one Product (Lets say Root Product)



Note:-

- 1) For successful working of this command, we should have strictly two Products containing kinematics under Root Product.
- 2) Two sub products must have mechanism related to it.
- 3) All the instance names of parts under two sub products MUST be unique. If not, you will have a problem at "Merge Kinematics". It is because of following reasons:
 - At "Merge Structure", if there are same instance names, they will be renamed by the command because you can't have a same instance name at the same level of product structure.
 - At "Merge Kinematics", some Kinematics joints will not be recreated properly because Kinematics joints rely on instance names and it will be lost if the name of pointing instance is modified.

► Hints and Tips: To share a common reference between two sub products

You can have a same reference (i.e., CATPart) between two sub products if instance names are different.

- It might be useful if you share a skeleton part between two products (i.e., Steering and Suspension).

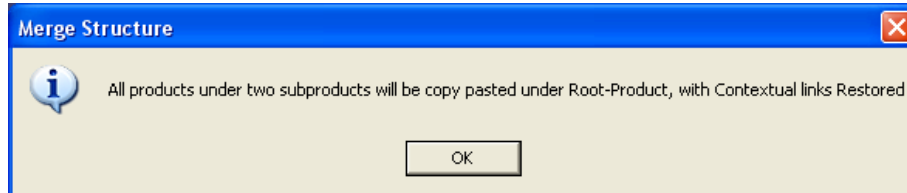
But as of now, at "Merge Kinematics", there is a limitation (you need to choose a Product [not a component] if you choose those two instances as "fixed components"). Please refer to 0.

2.3 User Operations

1) Click the Command.



2) Select Root Product and validate OK.



The result is illustrated in the adjoining screenshot.



Merge Kinematics

3.1 Purpose

This command recreates the mechanism for the “Merged Structure” (Result of command “Merge Structure”) at Root level.

► Hints and Tips: In case you need to define compound joints (Rack/Gear) later on

In some cases, you may want to add new compound joints (Rack/Gear) after merging to connect your two kinematics assemblies. For instance, suppose you merge steering kinematics and suspension kinematics. In that case, typically you need a Rack joint between the two. But obviously that Rack joint didn't exist in the original two kinematics models. So you need to add it by yourself.

The scenario will be something like following.

1. Create i) Suspension Kinematics, and ii) Steering Kinematics.
2. Merge those kinematics models.
3. Define a Rack joint which “connects” suspension and steering.

At step 3, you need to choose “Prismatic” joint from suspension side and “Revolute” joint from steering, because Rack joint is a compound joint. At this moment, one product must be commonly referred to by those two joints.

Figure 1 gives an example. P1 – P4 stand for products. R1 is Revolute Joint. Pris1 and Pris2 are Prismatic Joints. We can create Rack1, because R1 and Pris1 share P2. On the other hand, we can't define Rack2 because R1 and Pris2 don't share a product.

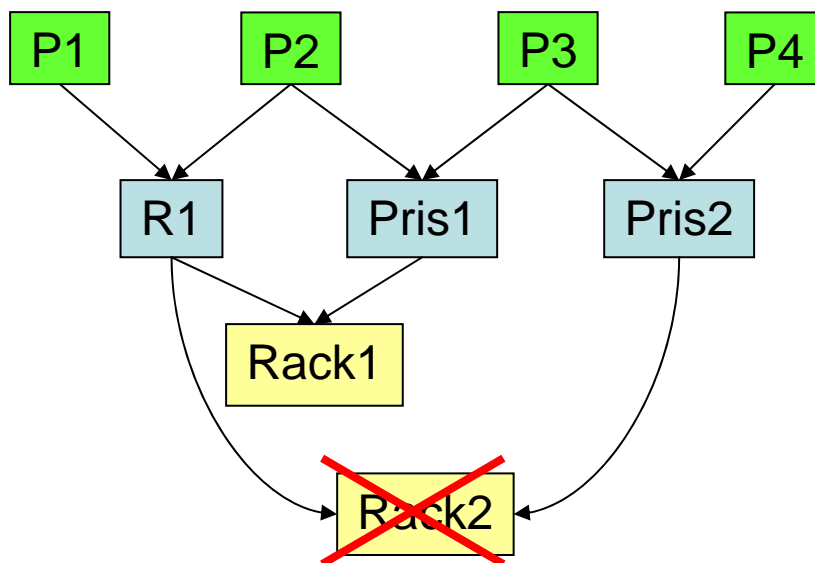


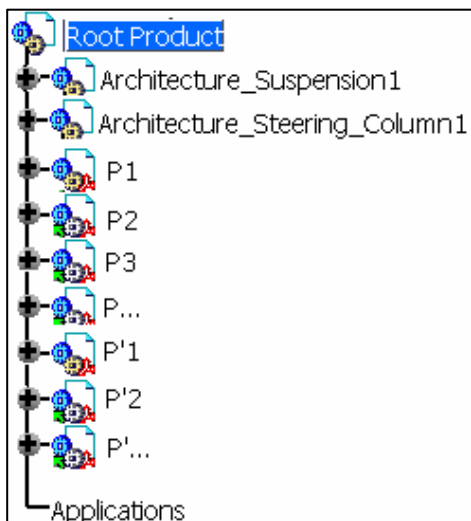
Figure 1 Prerequisite for Rack Joint Creation

Therefore, in case that you want to add a compound joint such as Rack that “connects” two kinematics definitions, you need to modify product structure and existing kinematics.

To deal with the case above, the command supports following scenarios.

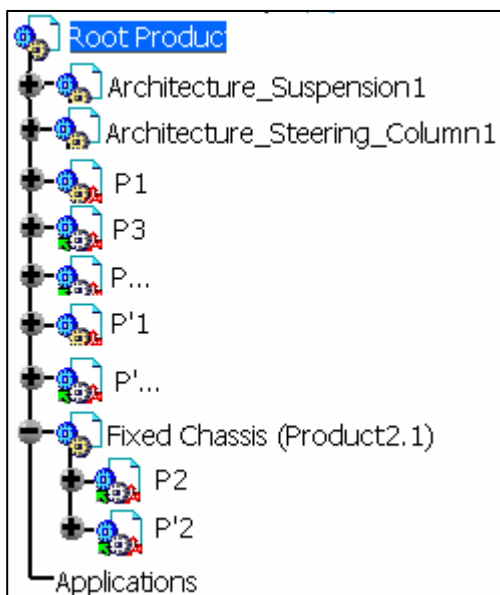
3.1.1 Scenario 1- Create A Common Product

1. Having Root Product with two architectures under it and all sub-products under each architecture pasted at Root Level.



P1 , P2 , P3 ... are products from Architecture Suspension
P'1 , P'2 , P'3... are products from Architecture Suspension

2. Executing the command for creating a new product, a dialog will ask to select two products coming from two different architectures which will be then moved under the new product created. (Here let's say user have selected P2 and P'2)



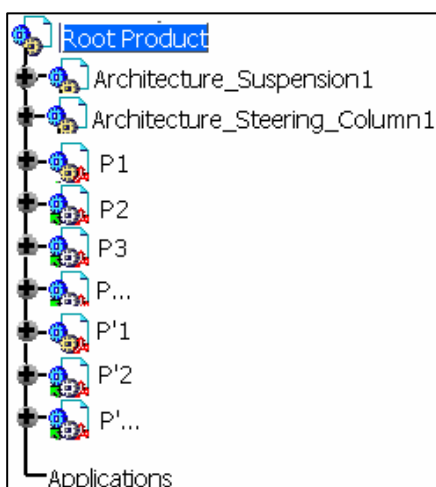
Fixed Chassis is a new product created.

Here P2 and P'2 are two products coming from two different architecture

3. Select Product (Lets say Fixed Chassis) to fix the mechanism:-
 - Mechanism created need to be fixed with reference to some product. Command allows you to choose the product with which user want to fix the mechanism.
4. Name the newly created mechanism.
 - A dialog will get pop-up to allow user to name the newly created mechanism.
5. Command ends with creating a Mechanism, reproducing the Joints and Commands from two architectures at Root Level

3.1.2 Scenario 2- Do Not Create A Common Product

1. Having Root Product with two architectures under it and all sub-products under each architecture pasted at Root Level.



2. Executing the command for not creating a new product, mechanism which will be the result of the merge of the two mechanisms from two architectures will get created.
3. Select Product to fix the mechanism.
4. Name the newly created mechanism.
5. Command ends with Creating a Mechanism, Reproducing the Joints and Commands from two architectures at Root Level.

3.2 Pre-requisite

For successful working of this command:

- Two Architectures (Products) under a Root product having their mechanisms defined and with their sub-products pasted at Root Level.

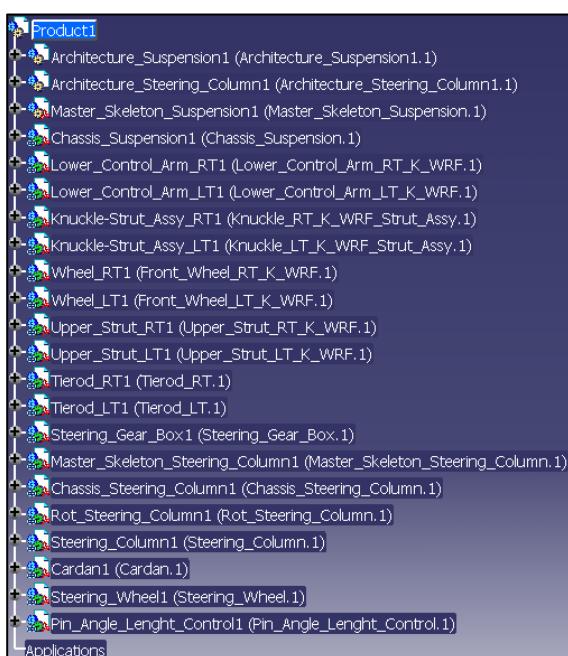
- Each architecture must have mechanism related to it.

Command will work on the result of Merge Structure command.

3.3 User Operations

3.3.1 Scenario 1- Create A Common Product

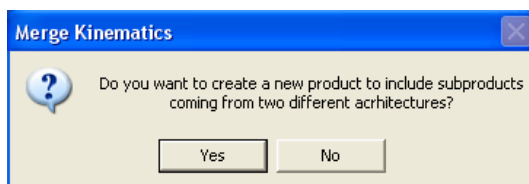
1. Actual data before executing the command.



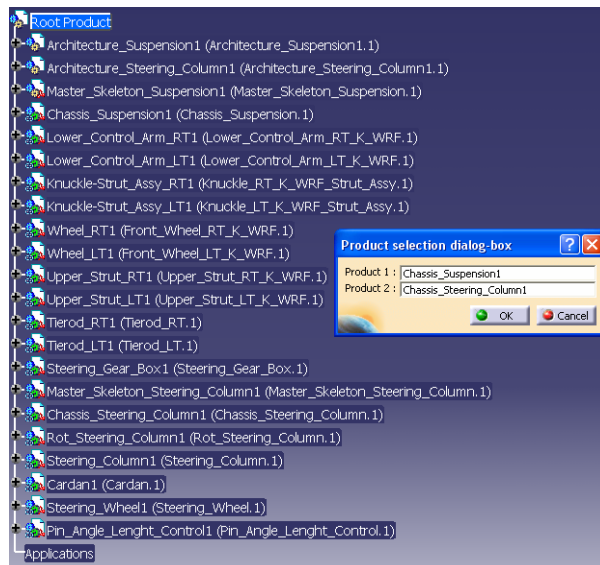
2. Click the command



3. Click “Yes” to create new product, which will contain two products coming from two different architectures (Products).



4. Select two products which will get moved under the new product getting created.



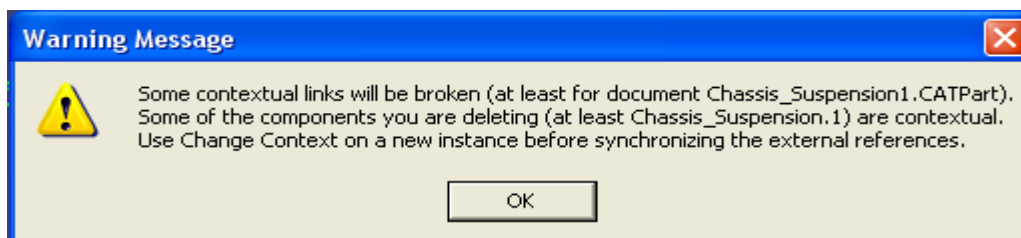
5. Choose to create product or component and name the newly created product/component.



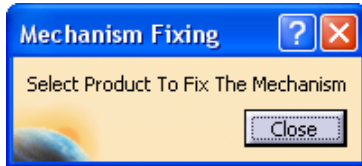
► Limitation: In case you share an identical reference between two sub products.

In case you choose two parts which had originally been the same reference (CATPart), you have to choose "Product" here (Do not choose "Component").

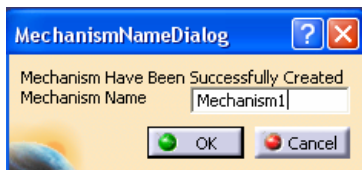
6. You will see the following Warning Message. Click OK



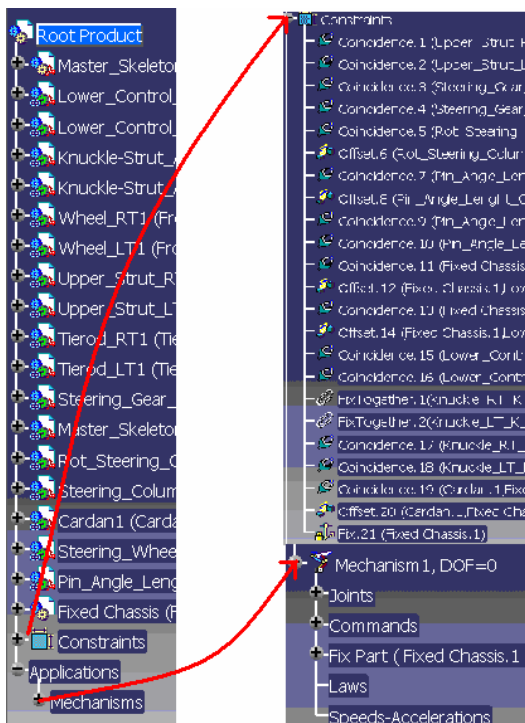
7. Click Close button and select the product to fix the mechanism.



8. Please enter the name for a newly created mechanism.



9. Result can be seen



All the Joints and commands from previous mechanisms (One from each architecture) are created under this architecture.

3.3.2 Scenario 2- Do Not Create A Common Product

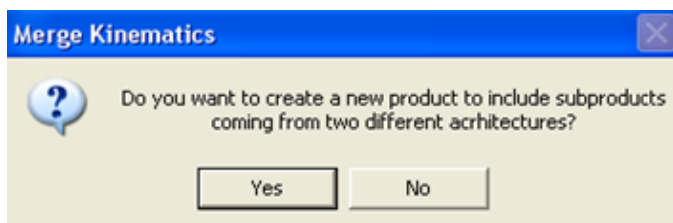
1. Actual data before executing the command.



2. Click the command.

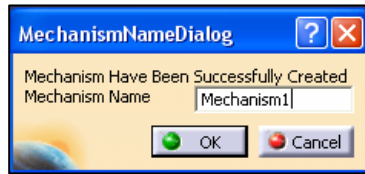


3. Click "NO"

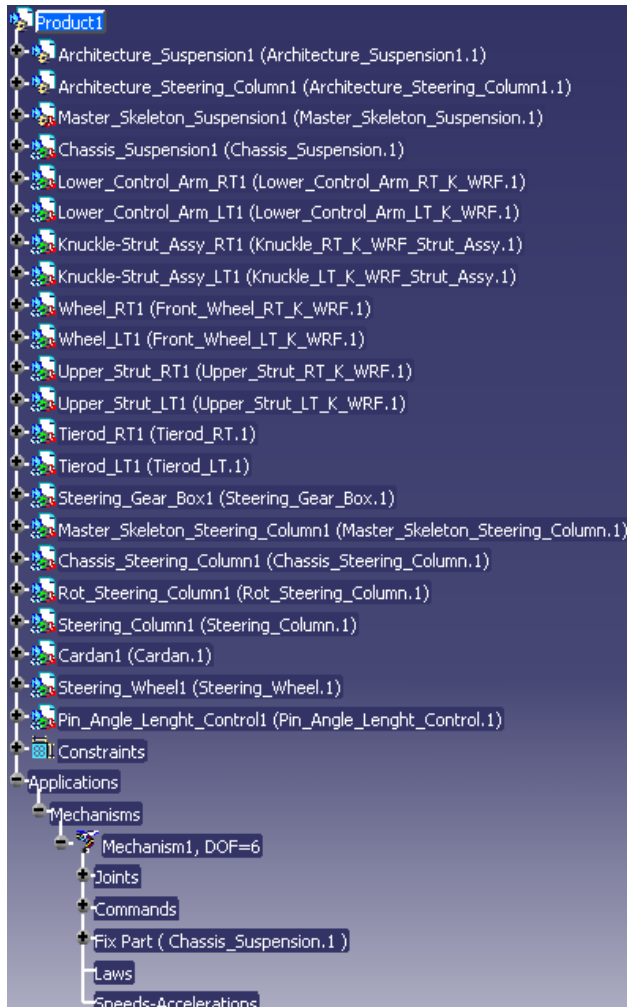


4. Mechanism has got created which will be result of merging of two mechanisms coming from two different architectures. Select a product to fix this mechanism.

5. Name the mechanism.



6. Result is indicated in the screenshot below



All the commands and joints from the two mechanisms from two architectures are now under this newly created mechanism.

Automatic Dress-up

4.1 Purpose

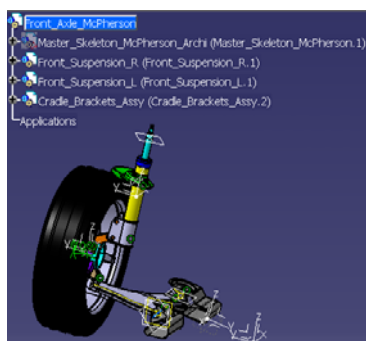
It automates Dress-up (link between kinematics models and corresponding 3D detailed parts) definition.

To find a relationship between kinematics models and 3D detailed parts, the command relies on the naming explained at 0.

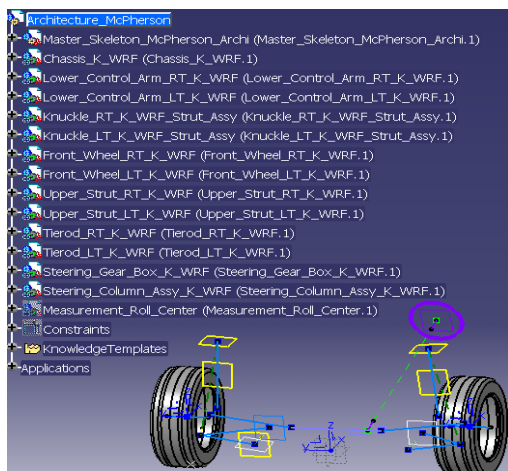
4.2 Pre-requisite

There need to exist an architecture with 3D parts and one architecture containing wire frame parts.

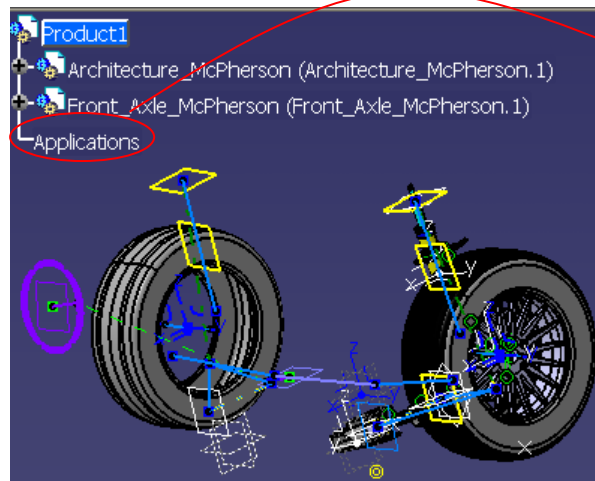
1. 3D Parts:-



2. Wire frame Parts:-

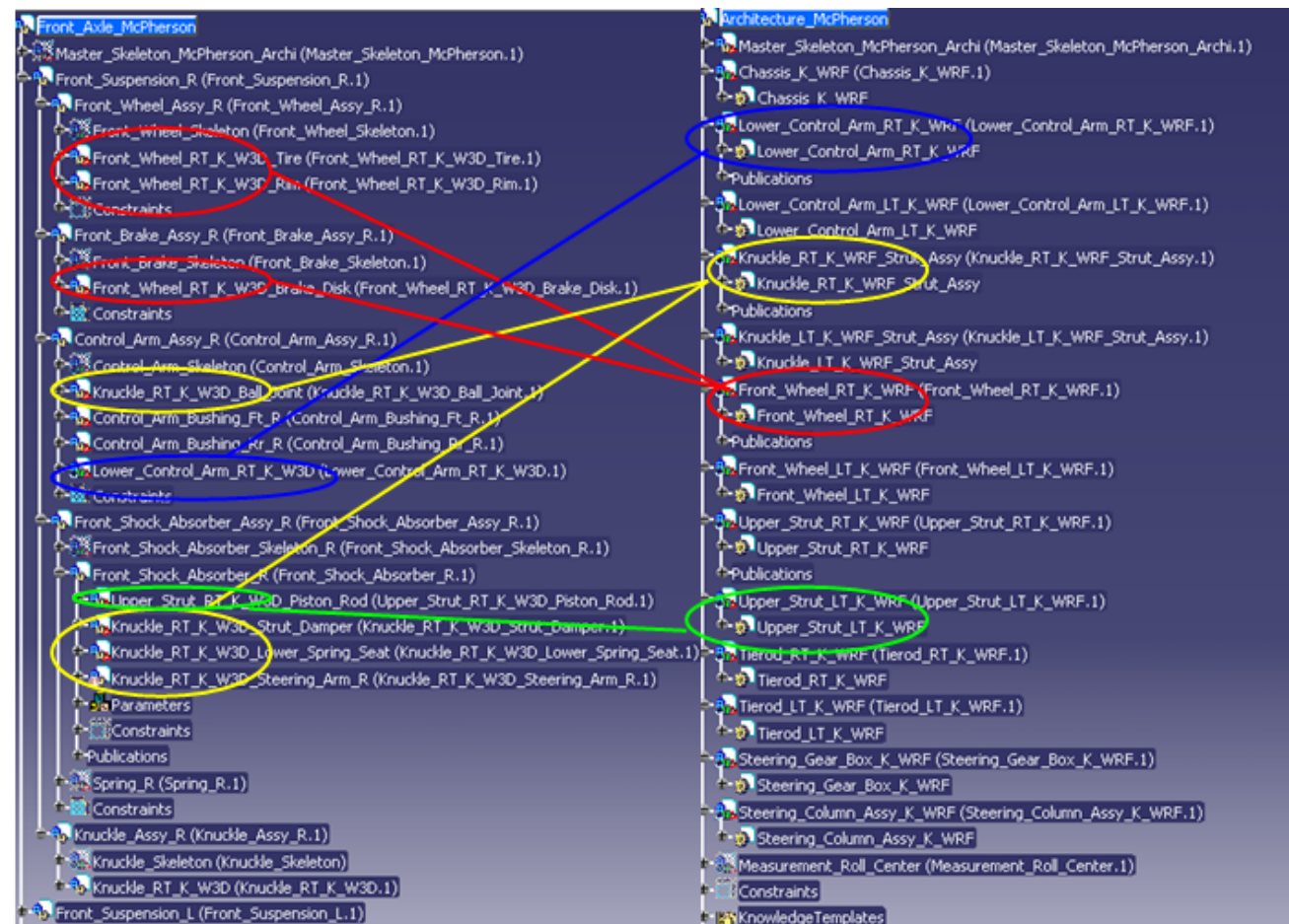


3. Open 3D parts and Wire frame under so as to have one common Father Product (here it is Product1)



Check that there is NO any imported mechanism.
If present then delete that one.

4. Check for the existence of respective match with appropriate Suffix (Like here, _K_WRF for wire frame and _K_W3D for 3D parts)

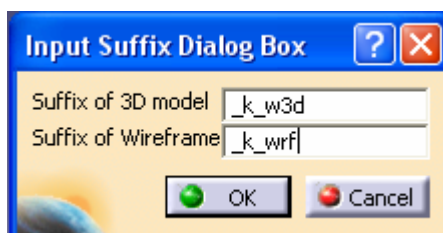


4.3 User Operations

5. Click the command-



6. Select an architecture containing wireframe parts.
7. Enter the appropriate suffix



8. Dressup Successfully created



9. Dressup is created.

