

Altair MotionView 2019 Tutorials

MV-1030: Creating a System Definition Using the MotionView GUI

altairhyperworks.com

MV-1030: Creating a System Definition Using the MotionView GUI

In the earlier exercise, you learned about MDL and authoring a definition using the MDL language through a text editor. In general, many of the definitions (such as systems, datasets, and analyses) are created using the MotionView graphical user interface. This tutorial demonstrates how to create a system using the GUI and save its definition to a file, which is an alternate way of creating a definition other than using the text editor.

Exercise: Creating a System Definition Using the GUI.

This exercise will help you learn to:

- Create systems using the MotionView graphical user interface
- Export a system definition to a file
- Reuse the saved definition by instantiating it in the model

Step 1: Creating a system instance.

To create a system, right-click on *Model* in the **Project Browser** and select *Add* > *System/Assembly*.

OR

Right-click on the System/Assembly panel button so on the Container Entity toolbar.

The Add System/Assembly dialog is displayed.

2. Select the **System** radio button and click **Next**.

The **Add System** dialog is displayed.

- 3. Specify sys_pendu as the *Variable*, Pendulum as the *Label*, and def_sys_pendu as the **Definition Name**.
- 4. Click **OK**.

The **Pendulum** system **i** is added to the model and its corresponding panel is displayed.

Step 2: Adding attachments to the system.

1. From the **Attachments** tab, click on the **Add** button (located in the middle of the panel).

The Add an Attachment dialog is displayed.

- 2. Specify the Label as Attachment Point and arg_p for the Variable, select *Point* (from the drop-down menu), and verify that the **Type** is set to *Single only*.
- 3. Click **OK**.



 Add another attachment with the Label as Attachment Body and Variable as arg_b, select Body from the drop-down menu, and specify the Type as Single only.

(یہ چ		sys_pendu	× 🗸	fin			
<u>.</u>	Attachments	Add	Variable	Label	Selection		Attachment Tag
<u> </u>	Options	Dialata	arg_p	Attachment Point	Point	Unresolved	<none></none>
	Initial Conditions	Delete	an h	Attachment Body	Body	Unresolved	(none)
	System Translation	Edit	1. M.C	r maan mark a aay			1101007
	Import/Export						

We have created two attachments to the **Pendulum** system which will be used to attach this system to other entities of a model.

Notice that the both of the newly created attachments are **Unresolved**, which means that the attachments are not yet referring to another entity in the model.

5. Double click on the <u>Point</u> collector.

The **Select a Point** dialog is displayed.

×
Global Origin
OK Cancel

6. Select *Global Origin* from the list on the right side of the dialog and click *OK*.



7. Similarly, click the Body collector, select *Ground Body* from the model tree, and click *OK*.

Step 3: Adding entities to the system.

- 1. Select **Pendulum** in the **Project Browser**.
- 2. Right-click and select *Add* > *Reference Entity* > ^O *Point*.

OR

– Right-click on the **Points** panel button \bigcirc on the **Reference Entity** toolbar).

The Add Point or PointPair dialog is displayed.

3. Specify the Label as $\tt Mass CG$, Variable as $\tt p_cg$, and the Type as Single.

Add Po	oint or PointPa	air		×
Parent:	System	Pendulu	m	
Label:	Mass CG			
Variable:	p_cg			
Type:	le al):			
				<u> </u>
•				
		ОК	Cancel	Apply

4. Click **OK**.

The **Points** panel is displayed with the properties of **Mass CG**.



5. From the **Properties** tab, click in the **X** Coordinate field and click on the **F** button on the **Expression** bar.

C Expre	ssion Bu	uilder (N	Mass CO	i-x)								×
Expression:										ОК		
0.0	0.0										Close	
F	ont		App	у		Jndo					Evaluated	I
Pro	perties	M	otion	Forc	e] (ieneral	Lo	cation]			
											Add	
▼ F	ilter ent	ities wi	ithin va	lid scop	e							
8-3	Mode 📄 Fil	el ename	e Data									
	- 📄 Op - 📄 Po	otion D bints	ata.									
	- 📄 Bo - 📄 Ma	dies arkers										
	- 🗋 Ve - 🗎 De	ectors ata Set	s									
۲	- 💽 Pe	endulur	m									
			1	_				Cx.	-		1 1	
7	8	9	+	Ran	ge	sin		Jo	Freq	PSD	Filter	
4	5	6	-	×	У	COS	exp	%ах	IFFTI	[FRF]	Fit	
1	2	3	*	()	tan	log	{	FFT 0	FRF 0	Functions	
+	0	•	1		^	π	alog	}	Window	Hilbert		

The *Expression Builder* is displayed.

6. Delete 0.0 from the **Expression** area (located at the top of the dialog).



 From the model tree, expand the *Pendulum > Attachments > Attachment Point* folders and select *x* (*x* is one of the property attributes of the point entity *Attachment point*).

	Mode 	el lename otion D pints odies arkers ectors ata Set endulur Attack Attack Attack # # # # #	e Data ata s ments achme label state id num idstring X y z data	nt Poin	t						
7	8	9	+	Ran	ge	sin	In	₽°×	Freq	PSD	Filter
4	5	6	-	x	У	cos	exp	ð∕ax	IFFTI	[FRF]	Fit
1	2	3	*	()	tan	log	{	FFT 0	FRF 0	Functions
+	0	•	1		^	π	alog	}	Window	Hilbert	

8. Click the *Add* button (located in the middle of the dialog).

arg_p.x is automatically filled in the **Expression** area.

9. Append +50 to this expression.

The complete expression should now read: arg_p.x+50.

Expres	ssion Bu	ilder (l	Mass CG	i-x)								×
Expres arg_p.	x+50											OK Close
From	ont perties		Appl	y Forc		Undo Seneral		cation	1		🗆 Evaluate	łd
arg_	p.x.	itiae w	ithin va	lid son					0		Add	
		ename tion D ints odies arkers octors ta Set endulu Attack - # - # - # - # - # - # - #	e Data ata ata mments achmei label state id num idstring y y z data	nt Poin	t						_	
7	8	9	+	Ran	ge	sin	In	l°×	Freq	PSD	Filter	
4	5	6	-	x	У	COS	ехр	ð∕ax	IFFTI	[FRF]	Fit	
1						ton	log	1	EET 0	EDE 0		
	2	3	*	_()	lan	log			THUT	Functions	

10. Click **OK** to close the dialog.

Through the above steps the point **Mass CG** is parameterized with regard to the **X coordinate** of the point **Attachment Point** and is placed at a distance of 50 length units in the X direction.



11. Repeat the above steps for the **Y** and **Z** Coordinates, by selecting attribute y and z respectively in the expression bar. Specify the expression for the **Y** Coordinate as arg_p.y and arg_p.z+100 for the **Z** Coordinate.

Alternatively, the expressions in **Y** and **Z** can be filled by copying the $arg_p.x+50$ expression from **X Coordinate** and editing it.

~	>	p_cg	X V fz.				
	Properties Measure	Coordinates: X: 50.0000 Y: 0.0000 Z: 100.0000	Get coordinates from node:	Node X: Y: Z:	Unresolved	Use values	Data Summary

Note The background color of the field changes for parametric expressions.

12. Right-click on the **Pendulum** system in the **Project Browser** and select **Add** > **Reference Entity** > **G Body**.

The Add Body or BodyPair dialog is displayed.

- 13. Enter Mass for the *Label* and b mass for the *Variable*, and click *OK*.
- 14. From the *Properties* tab specify the *Mass* as 1 and the *Inertia properties* as 1000 for *Ixx*, *Iyy* and *Izz* respectively.

(-	b	_mass	× V fr
9	Properties	Elex Body (CMS)	Get Properties from associated Graphic(s)
<u> </u>	CM Coordinates	, in the body (onito)	Data Summary
	Inertia Coordsys		
	Body Coordsys		Inertia properties:
	Initial Conditions		bxc 1000.0000 by: 0.0000
			lyy: 1000.0000 lxz: 0.0000
		Mass: 1.0000	lzz: 1000.0000 lyz: 0.0000

15. Click on the *CM Coordinates* tab and check the *Use center of mass coordinate system* option. Pick the point *Mass CG* as the **Origin**.

(-	1	b_mass X V fie	1
•	Properties	V Use center of mass coordinate system)
	CM Coordinates	15 of concernment years	l
2	Inertia Coordsys		l
	Body Coordsys	Orient two axes	
	Initial Conditions	Origin: Z Axis 💌 ZX Plane 💌	l
		Point Mass CG	

16. Right-click on the **Pendulum** system in the **Project Browser** and select **Add** > **Reference Entity** > **Graphic**.

The Add "Graphic" dialog is displayed.



17. Specify the Label as Rod, the Variable as gcyl_rod, the Type as *Cylinder*, and click *OK*.

The **Graphics** panel is displayed.

18. From the *Connectivity* tab; double click on the *Body* collector for **Parent** and pick *Mass* in the *Pendulum* system, click on the *Point* collector and pick *Mass CG* as *Origin*, and for **Direction** double click the *Point* collector and select the attachment to the system *Attachment Point*.

	gcyl_rod 🛛 🗙	🕻 📝 J.n.	
Connectivity Properties Visualization	Parent: Body Mass	Direction:	Cylinder Graphic
	Origin: Point Mass CG		

- 19. Go to the *Properties* tab and change the value of **Radius 1** to 2.
- 20. Next, add a *Sphere* graphic by right-clicking on the *Pendulum* system in the **Project Browser** and selecting *Add* > *Reference Entity* > *Graphic*.
- 21. Specify the Label as Mass, the Variable as gsph_mass, the Type as Sphere, and click OK.
- 22. Pick *Mass* for the **Parent** body and *Mass CG* as the **Origin**.

	gsph_mass 🔀 🗸 Jr.
Connectivity Properties Visualization	Parent: Sphere Graphic Body Mass
	Origin:
	Point Mass CG

23. From the *Properties* tab, specify 25 for the *Radius*.

(gsph_mass 🔀 🧹 🎉	-
 Image: A state Image: A state<th>Connectivity Properties Visualization</th><th>Radius: 25.0000</th><th></th>	Connectivity Properties Visualization	Radius: 25.0000	
		Material inside	

24. Right-click on the **Pendulum** system in the **Project Browser** and select **Add** > **Constraint** > **Joint**.

The **Add Joint or JointPair** dialog is displayed.



25. Select **Revolute Joint** from the drop-down menu, specify the **Label** as <code>Pivot</code>, the **Variable** as <code>j_pivot</code>, the **Type** as **Single**, and click **OK**.

The **Joints** panel is displayed.

26. From the **Connectivity** tab, double click on Body 1 and select **Mass** from the

Select a Body dialog (model tree). For the second body, click on the collector and browse through the model tree (*Model > Pendulum > Attachments*) and select *Attachment Body*.

C Select a Body	x
arg_b ✓ Only show entities within valid scope → Model → Filename Datas → Option Datas → Points → Bodies → Data Sets → Data Sets → Option Datas → Data Sets → Option Datas → Data Sets → Option Datas → Attachments → Attachment Body → Option Datas → Option Datas → Option Datas → Option Datas → Options → Options → Options	Attachment Body (THIS)
	OK Cancel

- **Note** Alternatively, you can click on the *Global Triad* (at the bottom left of the triad) to pick *Ground Body* via the **Attachment Body**.
- 27. Similarly, for Origin select Attachment Point (located under Model > Pendulum > Attachments). Use the Alignment axis drop-down menu to change from Point to Vector and select Global Y for the Alignment axis.

	i_pivot	🗙 🗸 fr.	
Connectivity Connectivity Initial Conditions	Revolute Joint	Body 1 Mass Body 2 Ground Body	
		Origin: Point Global Origin	Alignment axis:
			— 🛆 Alta

28. Save the model k to your < working directory> as pend_gui.mdl.

Step 4: Exporting the system definition.

- 1. Select the **Pendulum** system in the **Project Browser** and click on the **Import/Export** tab in the **Systems/Assembly** panel.
- 2. Select the *Export* option.

sys_p		sys_pendu	🗙 🗸 fiz
	Attachments Options	◯ Import ⊙ Export	Export topology as: MDL statements Property Data
	Initial Conditions System Translation	Select file:	C:/Altair/tutorials/working directory/sys_pend_gui.mdl
	Import/Export		Quick Export

- 3. Click on the **Select file** file browser $\stackrel{\frown}{=}$ and browse to your <working directory>.
- 4. Specify the name of the file as sys_pend_gui.mdl by clicking on the folder icon and clicking on **Save**.
- 5. Open the above file in a text editor.

The system definition content will look as displayed below:

```
*DefineSystem( def sys pendu, arg_p, arg_b )
  *Attachment( arg p, "Attachment Point", Point, "Select attachment.",
P Global Origin,
                 )
  *Attachment( arg b, "Attachment Body", Body, "Select attachment.",
B Ground, )
  *SetDefaultSystemInstance( sys pendu, "Pendulum" )
  *Point( p cg, "Mass CG" )
  *Body( b mass, "Mass", p_cg, , , , )
  *Graphic(gcyl rod, "Rod", CYLINDER, b mass, p cg, POINT, arg p, 2.0,
gcyl rod.r1, , 0.0, CAPBOTH )
  *Graphic(gsph mass, "Mass", SPHERE, b mass, p cg, 25.0)
  *RevJoint( j pivot, "Pivot", b_mass, arg_b, arg_p, VECTOR, V_Global_Y
  *SetPoint( p_cg,
                                          arg p.x+50, arg p.y,
arg p.z+100 )
  *SetBodyInertia( b mass,
                                                1.0, 1000.0, 1000.0,
1000.0, 0.0, 0.0, 0.\overline{0})
  *Set( b mass.usecm, true )
*EndDefine()
```

Note The **Export** option is only available for Systems and Analyses. For other definitions like Datasets or Templates, the definition can be copied from the model .mdl file.



- 1. Select the *Model* system in the **Project Browser**.
- 2. Click the *Import/Export* tab in the **Systems/Assembly** panel.
- 3. Click on the **Select file** file browser $\stackrel{\frown}{=}$ and browse to your <working directory>.
- 4. Select the sys pend gui.mdl file and click **Open**.

		the_model	🗙 🥑 fra
1	Properties	Import	
	Options	C Export	
C	Initial Conditions		
	System Translation	Select file:	C:/Altair/tutorials/working directory/sys_pend_gui.mdl
	Import/Export		

5. Click the *Import* button.

The **Import System/Analysis Definition** dialog is displayed.

Import System/Analysis Definition			×
Select a definition:	System	•	ОК
def_sys_pendu			Cancel
Label:			
Pendulum			
Variable:			
sys_pendu			
Note:			
		_	

- 6. Select def_sys_pendu.
- 7. Change the Label to Pendulum 2 and the Variable to sys_pendu_2.
- 8. Click **OK**.

The system definition is instantiated.

9. Select the *Pendulum 2* system in the **Project Browser**.



- 10. Go to the **Attachments** tab and resolve the attachments in the following manner:
 - Double click on the <u>Point</u> collector. In the model tree that appears, click on the **Pendulum** system, select **Mass CG** from the list on the right, and click **OK**.

V Select a Point	x
sys_pendu.p_cg Only show entities within valid scope Becommended Entities Global Origin Global Origin </td <td>Attachment Point Mass CG</td>	Attachment Point Mass CG
	OK Cancel

- Click on the Body collector. In the model tree that appears, click on the **Pendulum** system, select **Mass** from the list on the right, and click **OK**.
- 11. Save the model k to your <working directory> as pend_2_gui.mdl.

The same system definition can be reused to instantiate several times either within the same model or in a different model.



