

Making a seven seg display with RWIE

Sonelec-Musique > Logiciels > Librairies pour Proteus / RWIE

Last update : 24/03/2007

Preview

In this tutorial, we'll see how to create a 3D model of a seven seg display, with RealWorld Icon Editor software (RWIE can be downloaded on <http://www.rw-designer.com>). Once display object created with RWIE, we'll see how export it in a 3DS file format that can be used in Ares, that is the PCB creation software included in Proteus suite.

From the unique model made with RWIE, we'll also see that it is possible to define the number to display and the color of the Led segments. With that, it will be possible to export a 3DS model file that show the number 2 in Red, and another 3DS model file that show the number 7 in gree, for example.

Note : it is possible to create a simpler object than one that will be created here, and that should be capable of only display a fixed number and a unique color, but i think it's more interesting to show how to use internal mechanism included in RWIE that allow some very interesting things.

3D model structure definition

We want to create a 3D object that looks like a seven segment display model HP-5082-76xx, that can be used in Ares 3D viewer. We'll définie three objects groups (containers) that make display, once assembled :

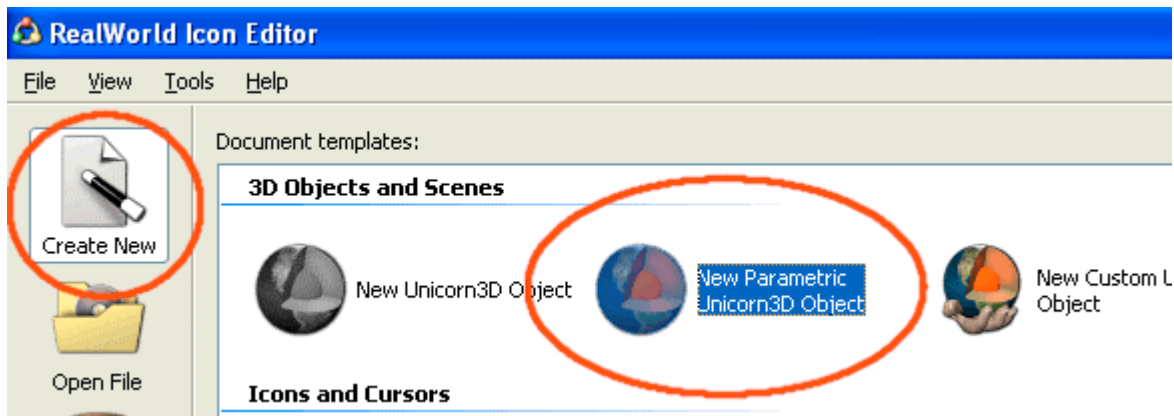
- An objects group for the body of the component,
- An objects group for the seven led segments,
- An objects group for the wire.

Creation of the main 3D model structure

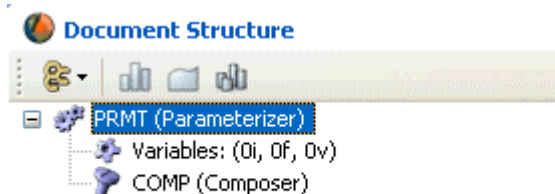
RWIE allow to work with primitives that can stand on an unique hierarchical level, or with primitives that are arranged on several hierarchical levels. It's possible to let all primitives on a same level, but it's better to organize them in several levels to classify them, as you should do with your Word documents and electronic pictures, to avoid mix them on your hard drive. So we decide to construct a base structure as following.

```
| - HP-5087-76xx (Level 0)
  | - Body (Level 1)
    | - Part_01 (Level 2)
  | - Segments (Level 1)
    | - Seg_01 (Level 2)
    | - Seg_02 (Level 2)
    | - ...
    | - Seg_07 (Level 2)
  | - Wires (Level 1)
    | - Wire_01 (Level 2)
    | - Wire_02 (Level 2)
    | - ...
    | - Wire_14 (Level 2)
```

Names are arbitrary choosen, I decided to use english names to allow tutorial be written in several languages. Start create this base structure, begining with creation of a new Parametric Unicorn3D object project. For this, select Create New, and double-clc on New Parametric Unicorn3D object icon.

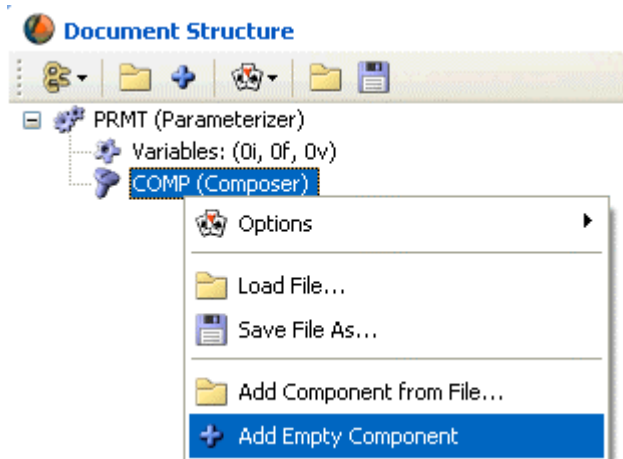


A new base structure, called PRMT (Parameterizer) is created. This new structure contains a variables container called Variables and a new components container called COMP (Composer).

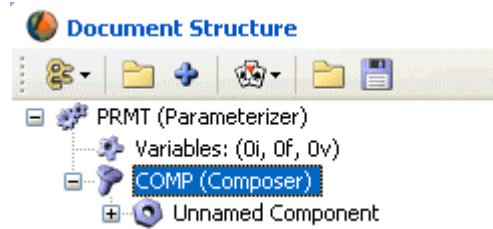


At the moment, let the Variables container, and make interest on the COMP container. In the COMP container, we'll create a new container that will contain three objects groups (three other containers). We don't create object directly in the COMP container to later allow more manipulation on the final 3D composed full object.

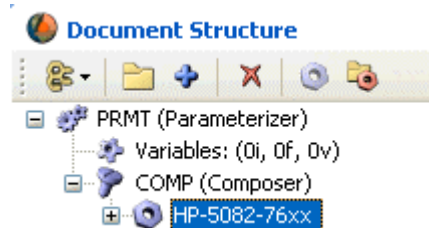
Right-click on the COMP container to show its contextual menu, and select Add Empty component command.



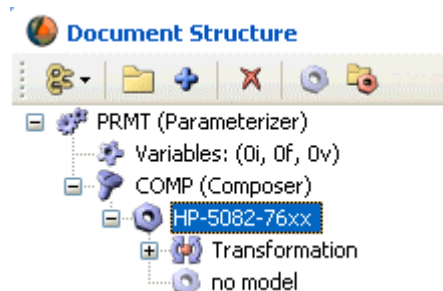
A new component called Unnamed Component is added. To see it, clic on the + sign on the left of COMP.



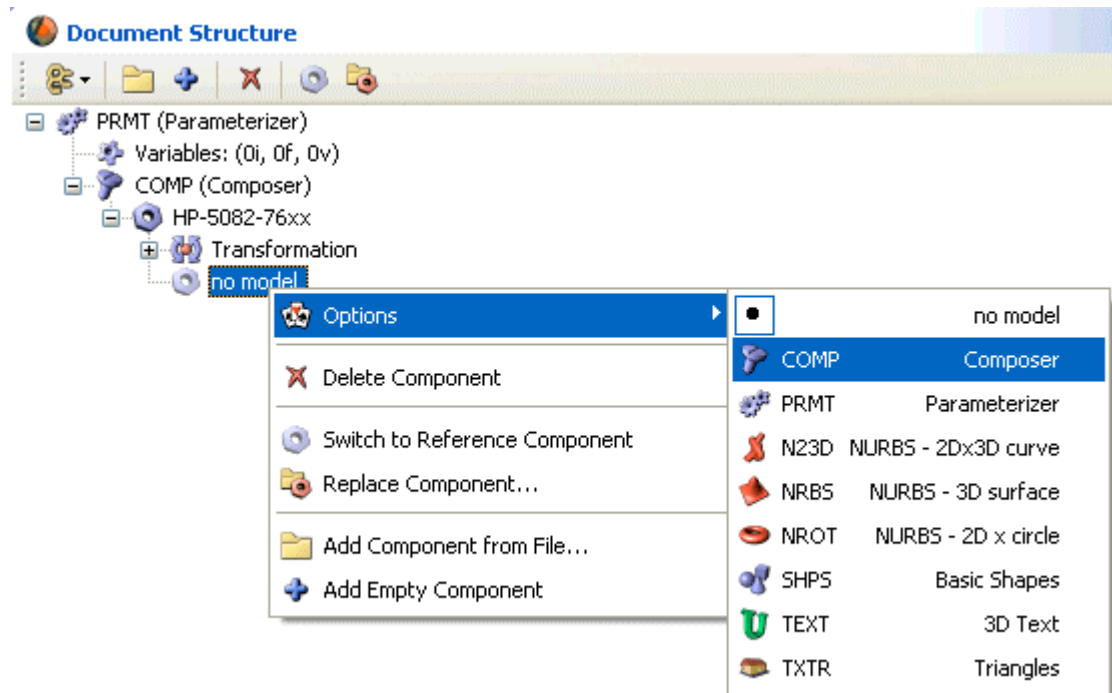
Rename this new component, from Unnamed Component to HP-5082-76xx. Clic one time on the name Unnamed Component, wait one seconde, and clic a second time on it (or clic once time and hit F2 function key of your keyboard), change actual name to HP-5082-76xx, and press Enter key to validate new entered name.



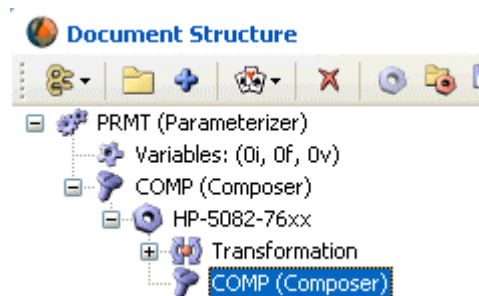
Clic on the + sign on the left of HP-5082-76xx item, and you'll see two elements : one called Transformation (we'll see later for what it can be used), and another called No model.



Right-clic on No model to show its contextual menu, and select Options command, that allow developp another sub-menu that list all base components that can be used here. In this sub-menu, select COMP.

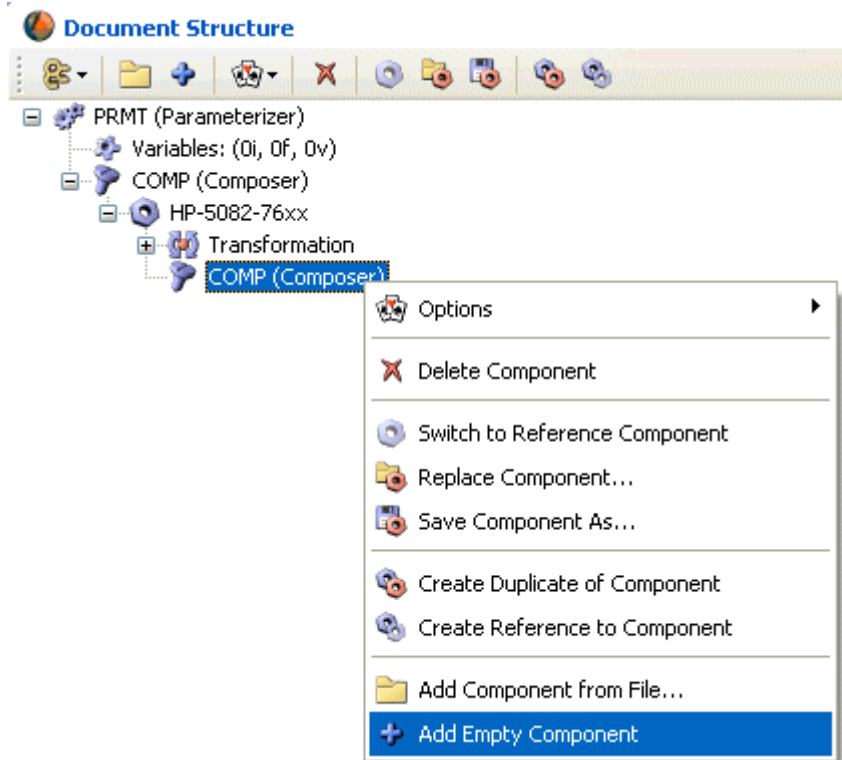


You should see the following :

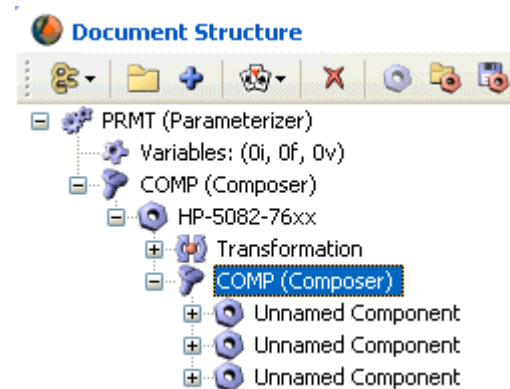
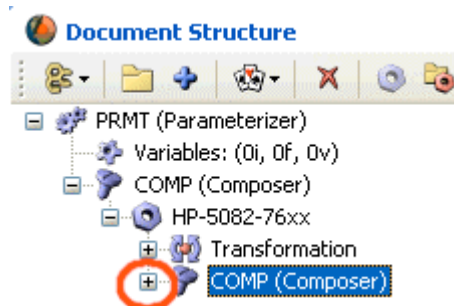


We have now a second COMP container that is included in the first COMP container. This second COMP container will contain our three groups Body, Segments et Wires.

Now, add our three containers. Right-click on COMP, and select Add Empty Component command. Repeat twice this operation (right-click on COMP and Add Empty Component).

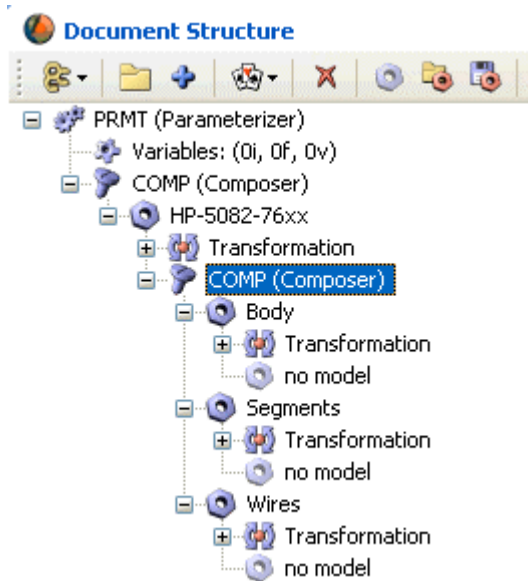


A + sign appear on the left of COMP, that indicate that container is no more empty. Clic on this + sign to developp it.



As the same manner, click on the + sign on the left of each new Unnamed Component, to expand respective branch. As you can see, we have, for each new element, two items called Transformation and No model.

Now, rename each new created element with previously decided names, Body, Segments and Wires.



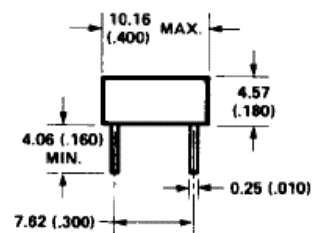
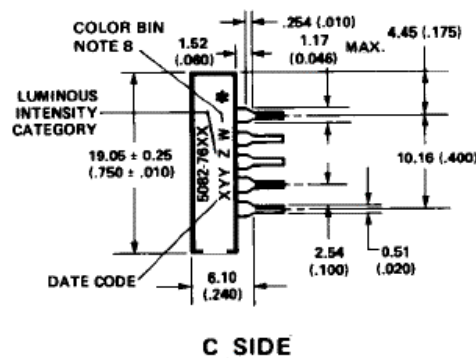
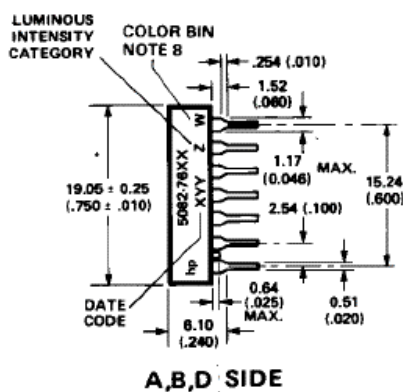
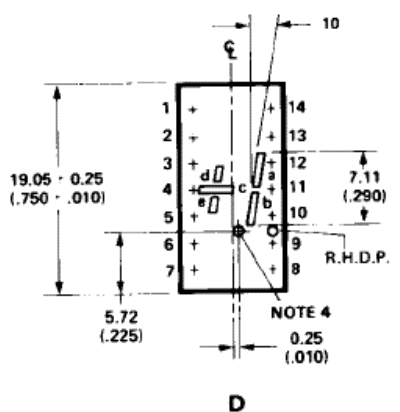
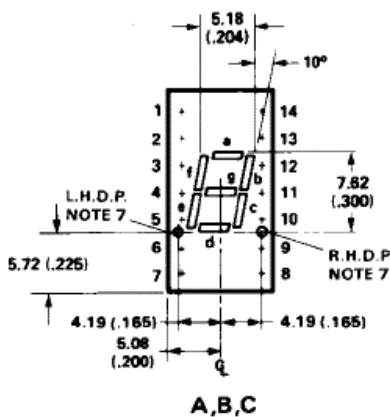
Well, base structure is created with its three main containers Body, Segments and Wires, and we'll be able to populate them. But before work with them, we have to speak about sizes of the 3D model to create.

Note : think to regularly save your work. Do it now.

Seven seg display size

The size of the display and its main parts are those specified in the manufacturer HP-5082-7610 datasheet, as shown here. Sizes are specified in inches (english unit) and in metric unit.

Package Dimensions



*The Side View of package indicates Country of Origin.

For this tutorial, I decided to use the Inch unit, but it was also possible to use the metric unit, as Ares allow specify the scale to use for viewing. So, I'll use the "in" unit in following lines.

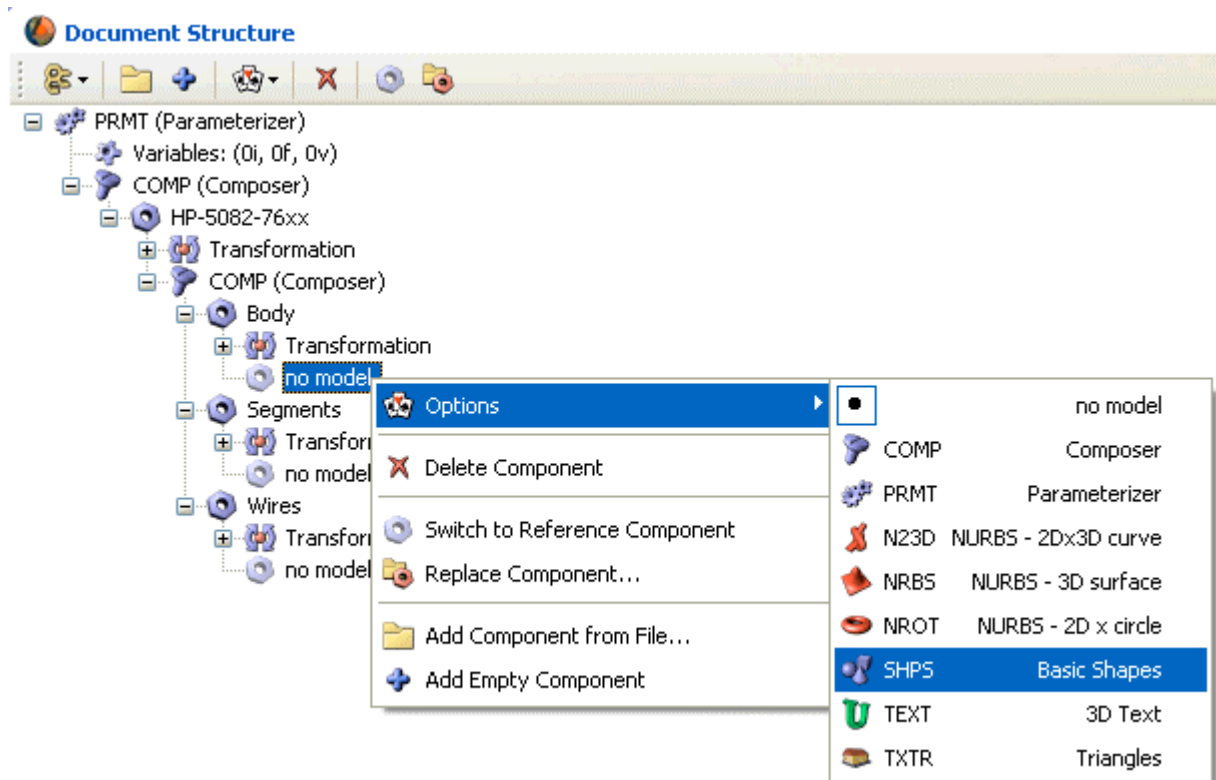
5082-7610 datasheet indicate that main body width is 0.400 in, length is 0.750 in, and height is 0.240 in. We'll start by add a simple BOX component that have this sizes.

Main body part creation (Body container)

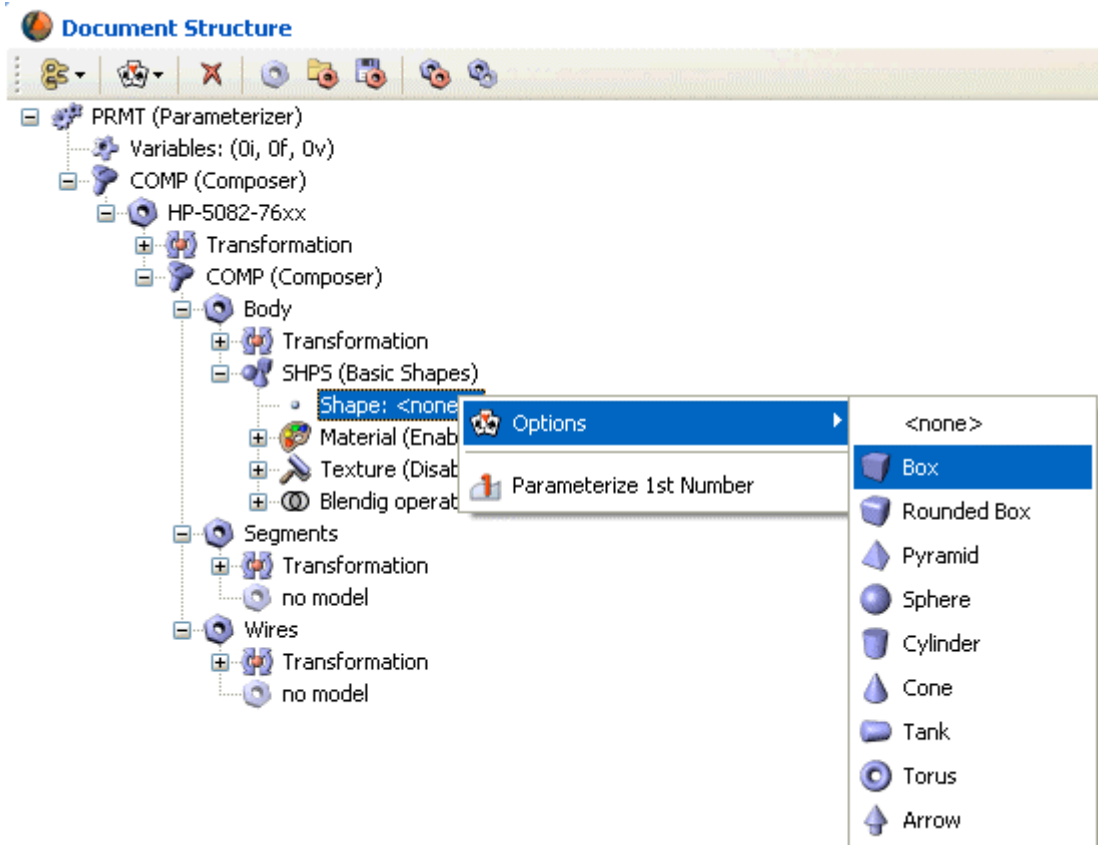
The display body is made with only one element. Later, and if needed, you'll can add other object to allow more realistic aspect. But don't forget that more elements need more time to be drawn.

Main part creation

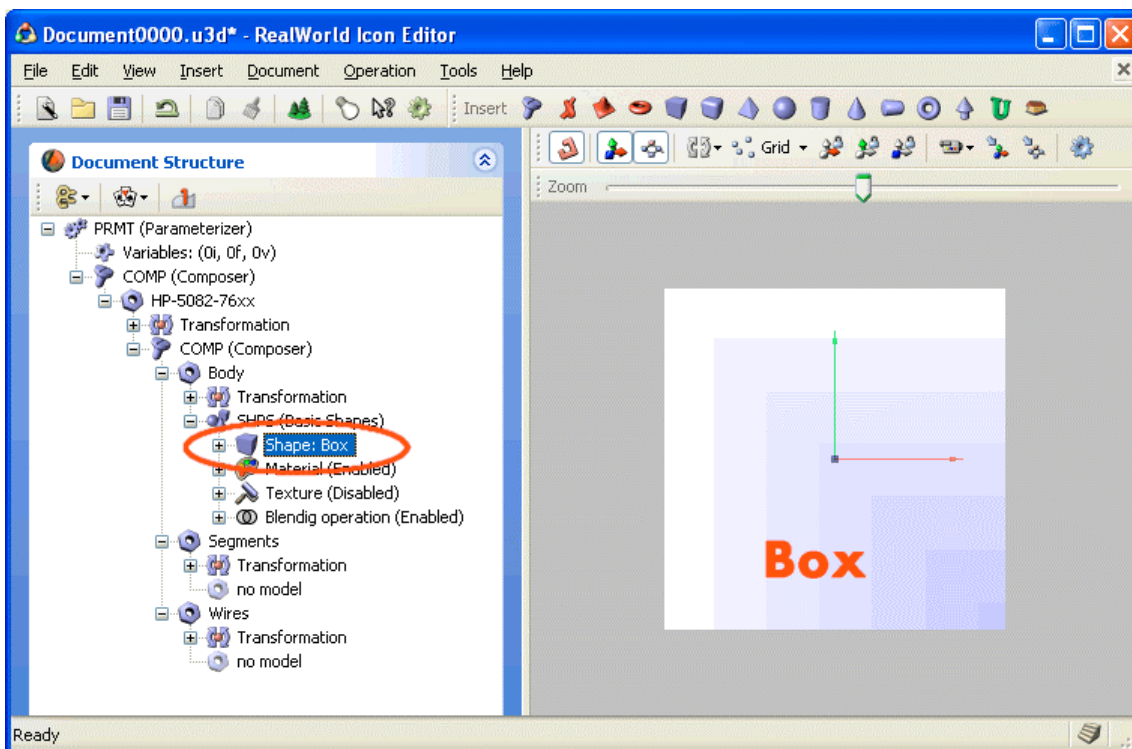
Display body can be viewed as a simple rectangle, and a simple 3D object as BOX is enough. Add a BOX objet in our Body container, by right-clicking on the No model item contained in the Body group. Select Options command, and SHPS (Basic Shapes) command.



SHPS object is not really a component, but an object type. It allow choose a basic shape, as cube, sphere or cylinder. Once this object type selected, you have to specify what sort of 3D figure you want to use. Expand SHPS component by clicking on its + sign on the left, and right-click on Shape: <None>. In contextual menu that appear, click on Options, and on Box.

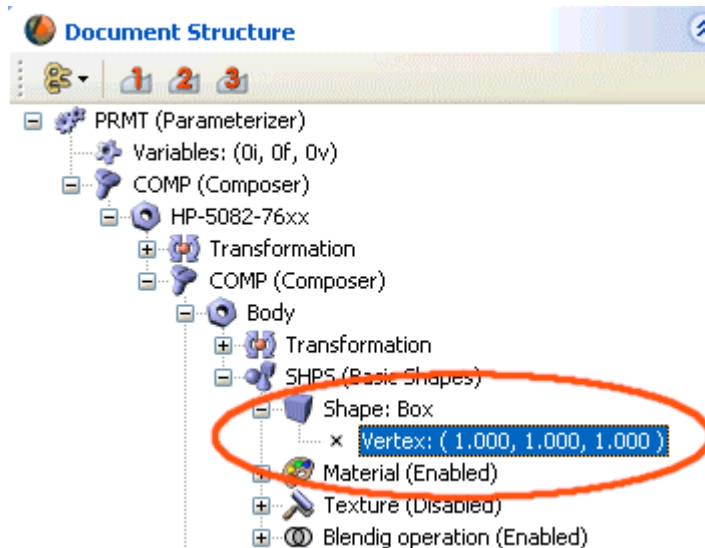


Wow, our first 3D object is inserted in global structure and can be viewed in the RWIE 3D viewer :

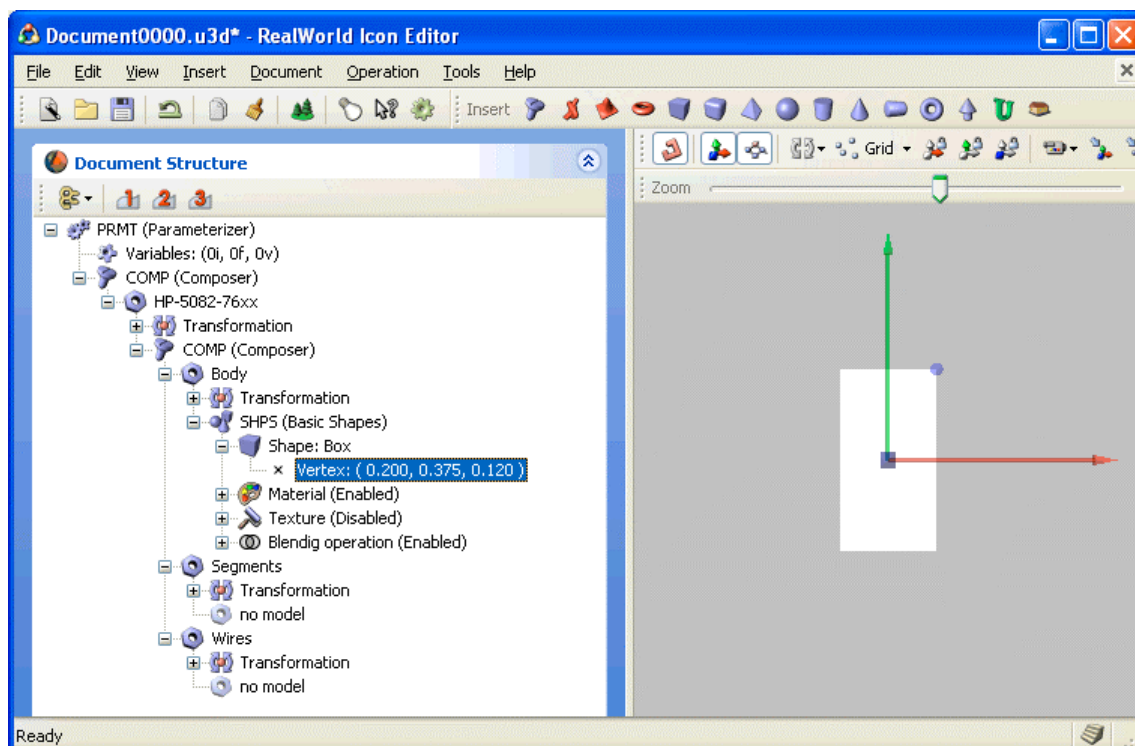


Now, we have to resize the Box object with correct dimensions, knowing size values to specify are relative to center (reference) point that is localised at (0.000, 0.000, 0.000) spatial position.

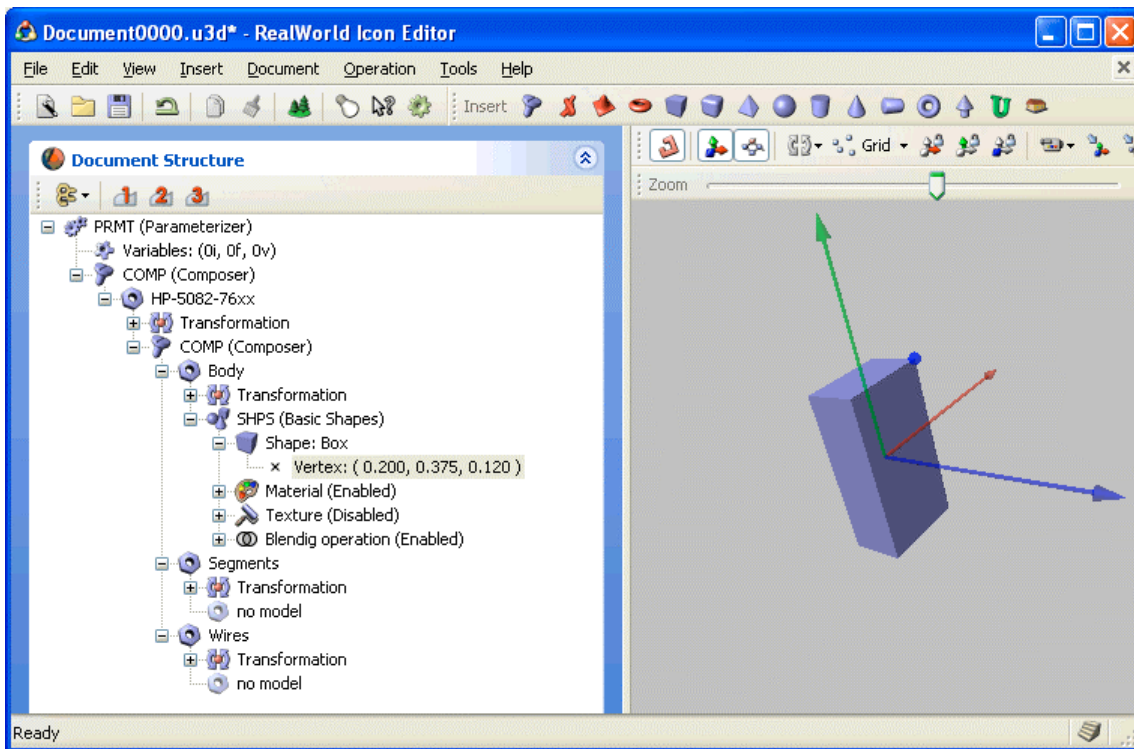
Clic on + sign on the left of Shape: Box item, to expand it. You now can see that this object has coordinate (Vertex) [1.000, 1.000, 1.000], it's a cube for wich each side is 2.000 lenght (+1.000 and -1.000 from reference center 0.000).



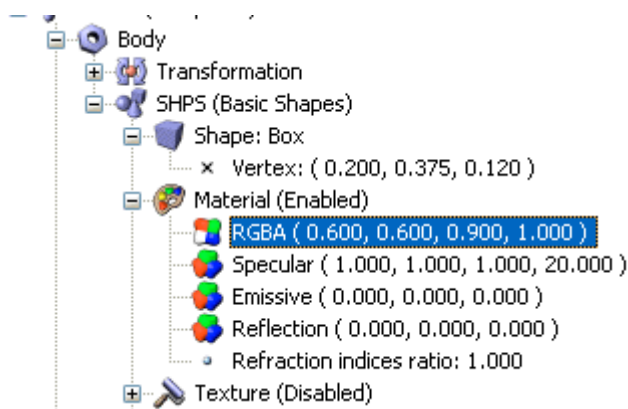
If we specify directly datasheet values to our box object, sure it will be twice regular size. For correct sizes, we have to specify half of the datasheet sizes. So, we specify for the box object, the Vertex [0.200, 0.375, 0.120], and not Vertex [0.400, 0.750, 0.240]. After specifying this sizes, you can see an object similar to one we can see on the following screen capture.



Note : to turn around BOX object in the 3D viewer, you can move your mouse while maintaining the right mouse button pressed.



The actual color is not what we want, we have to slightly modify it. Clic on the + sign of the Material item, to see actual values of the three primary colors : Red, Green and Blue (RGB).



Actual RGBA values are

R = 0.600 (Red = rouge)
G = 0.600 (Green = vert)
B = 0.900 (Blue = bleu)
A = 1.000

Change them to :

R = 0.700
G = 0.700
B = 0.700
A = 1.000

Note : clic on an actual value, wait one second, and clic again on the value (or use your F2 function key).

Note : RVB parameters can be manually changed, but it's easier to make usage of the sliders to quickly change selected values.



Et voilà, main body is finished !

Segments creation (Segments container)

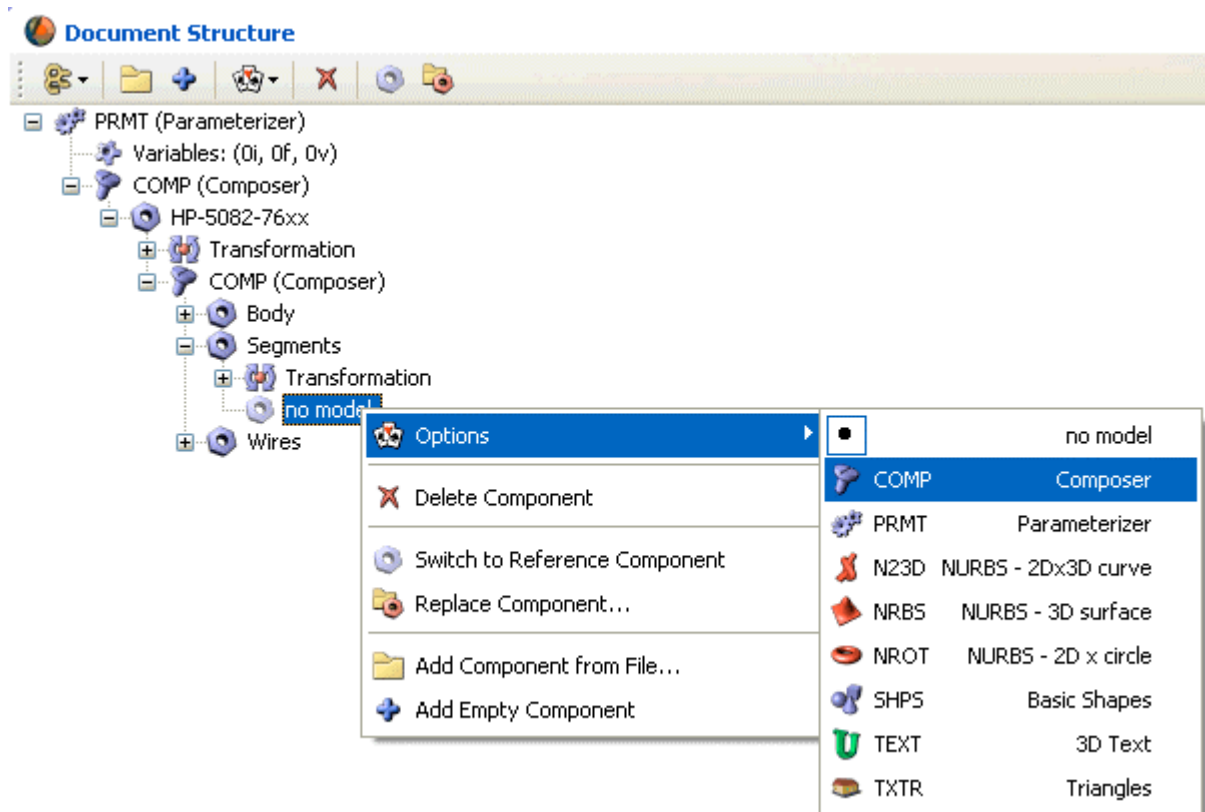
All segments are the same, but three are horizontal (A, D and G segments) and four are vertical (B, C, E and F segments). To have not repeat a lot of times the same operation, we'll create only one horizontal segment and only one vertical segment. After that, we'll duplicate them and change their spatial position.

Horizontal segment creation

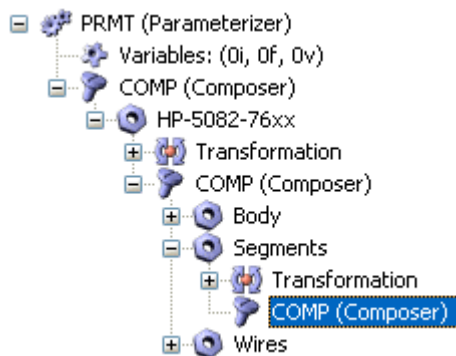
Start with the G segment, which is placed in center of the display.

If the segment was a classical rectangle, we could use a simple Box objet. But segments are inclined (10°) and shape of segment is not a perfect rectangle. So, we have to search a more matching shape. In our case, the NRBS 3D object is the object that seems the more appropriate, as it allow creation of polygonal shapes.

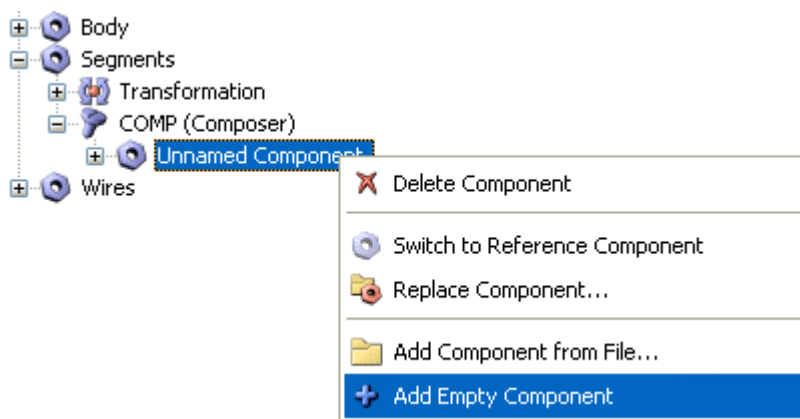
As previously done, we'll create a container, this time to handle all segments. This is not an obligation, but allow manipulate them in one bloc. In the Segments group, right-click on No model, and select Options and COMP commands, to add a new COMP container.



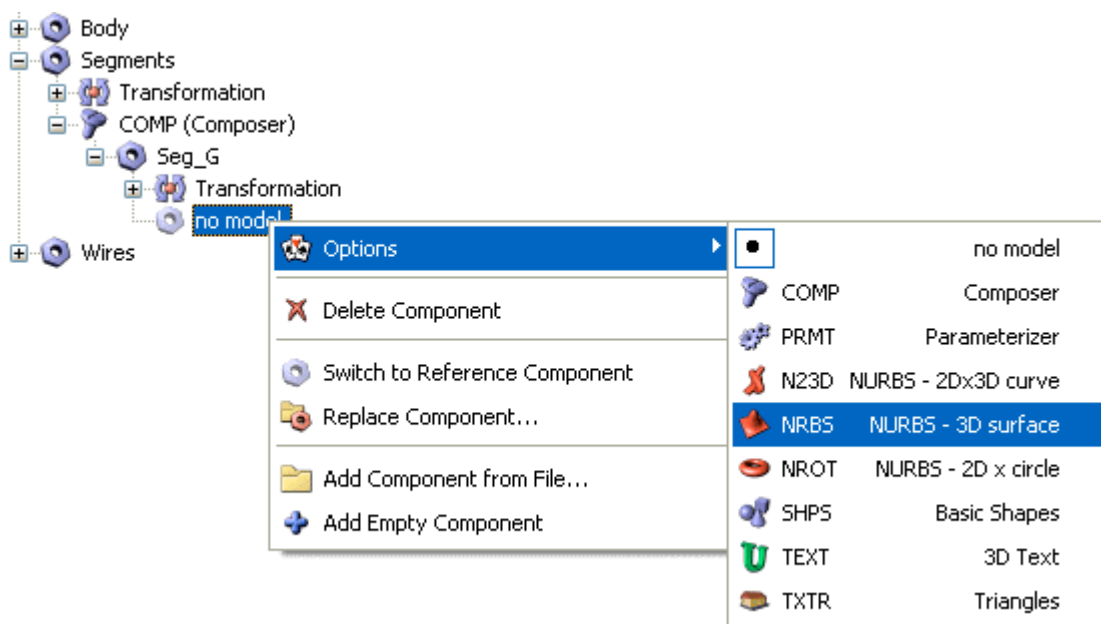
You should have this :



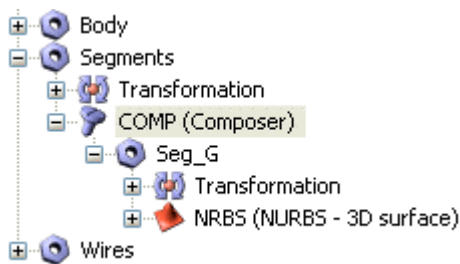
Now, add an empty component in this new container, by right-click on Unnamed Component item, and by using Add Empty Component command, that should be familiar now.



Rename Unnamed Component item to Seg_G, and expand it to see what it contain (if you don't know how to do these two actions, re-read previous text). And now, make the empty component as a rectangular surface (NRBS), with Options and NRBS commands, that can be accessed after a right-click on the No model item.



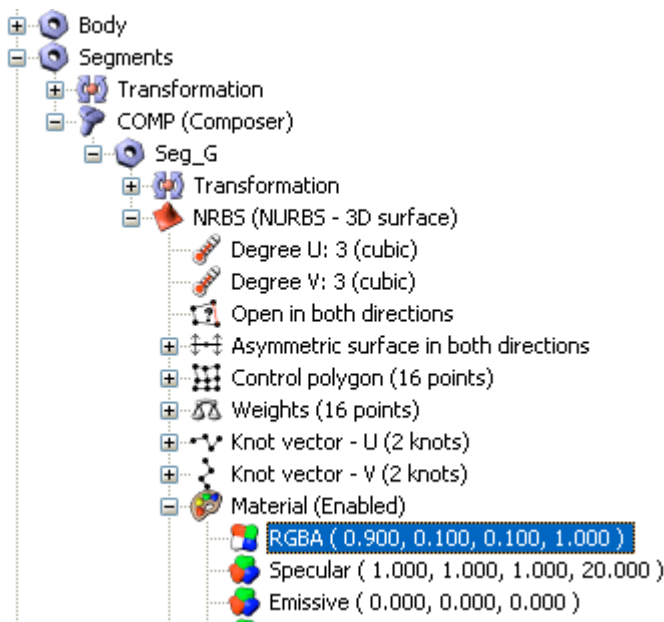
Seg_G element now contain a NRBS 3D surface object.



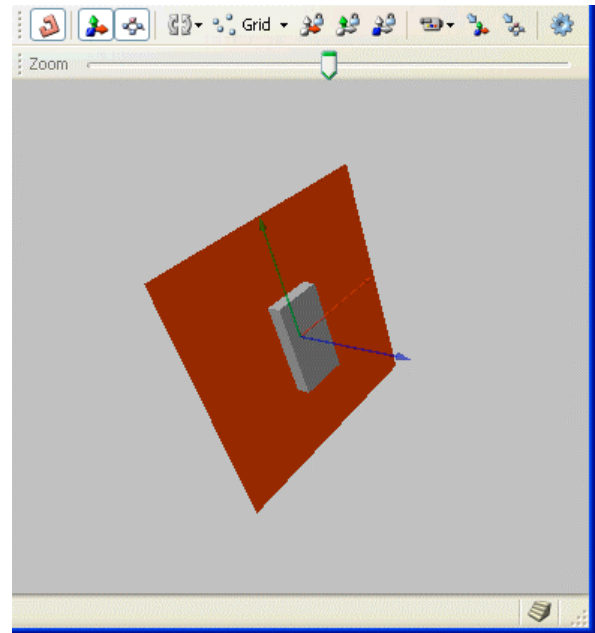
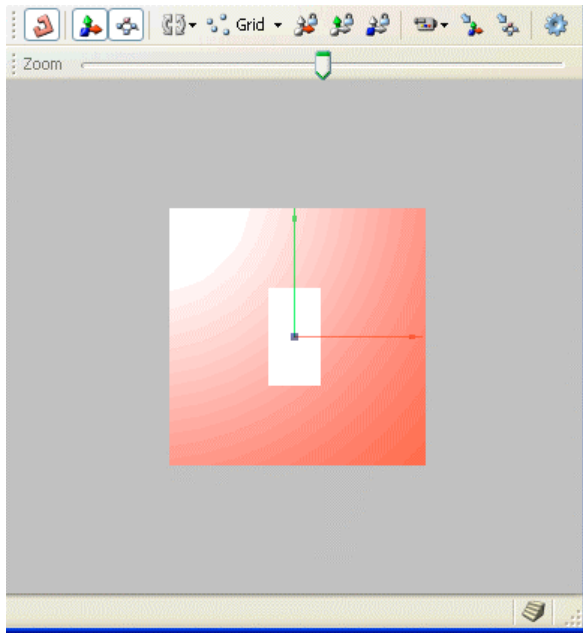
This object contains a few parameters that allow you to modify its visual appearance, you can click on its + sign to see them.

Well, we now have two visible objects in the RWIE 3D viewer. That we see is not really what we want to see, and this is more true if the two objects are viewed from the front. To have a better view, and because some operation order is not important, we'll change the second object's color, and make it red, before resizing it. This will allow the two objects to be better distinguished.

Right-click on the + sign of the Material item, so we can see and modify the RGBA parameters. As previously seen, the first parameter is relative to red color quantity, and the two following are relative to Green and Blue color quantity. We want a red-colored segment, that's why we specify (0.900, 0.100, 0.100, 1.000) values to RGBA parameters (let the A parameter to 1.000).



With these new RGBA parameter values, the second created object is red, as you can see on the following screen capture. Note that the Seg_G object cuts the Body object on its center. This is more visible if you turn around objects in the 3D viewer (move mouse while keeping the right mouse button pressed).

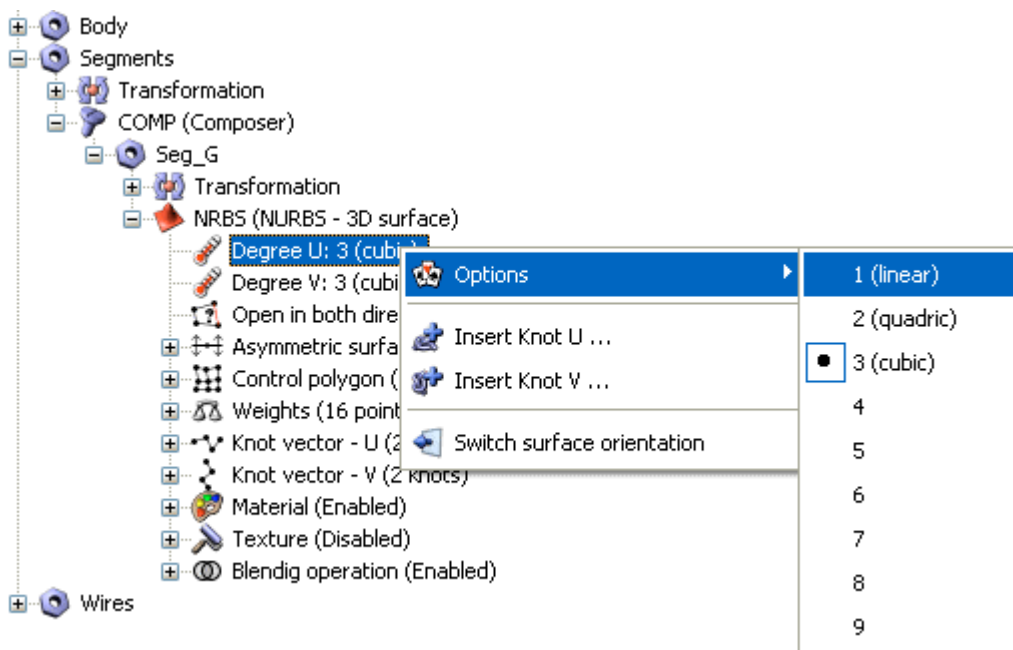


All this is beautiful, but segment is too large and we have to reduce its size.

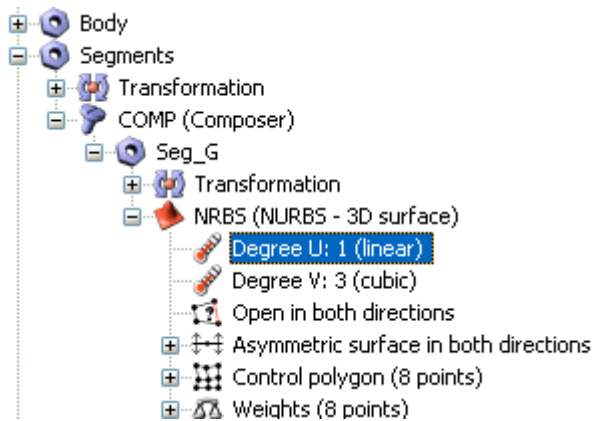
If you want take a cofee before continue, you can.

Before resizing our segment object, a little information about NRBS object is necessary. NRBS object contain some control points, and the number of control points can be adjusted for vertical orientation and for horizontal orientation. By default, the NRBS object has 16 control points, and we need only 4 of them for making our segment. So, we'll reduce the number of control points, for easily modify our object, and to limit its weight. For this, we have to reduce the number of control points in the two orientations "Degree U" and "Degree V".

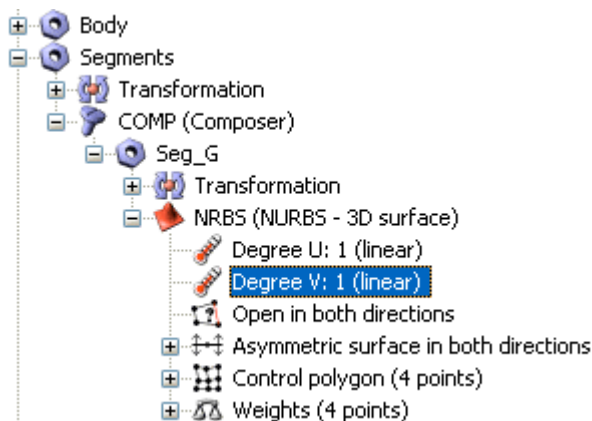
Right-click on Degree U item, and via contextual menu, select Options command, and 1 (linear) command, to change the selected parameter from 3 (cubic) to 1 (linear).



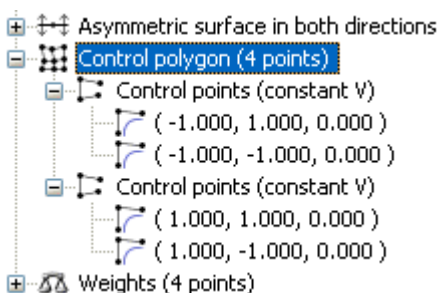
After this modified, you should have this :



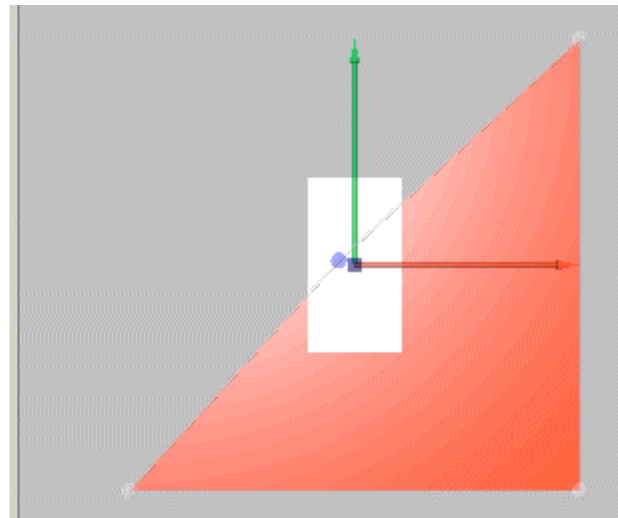
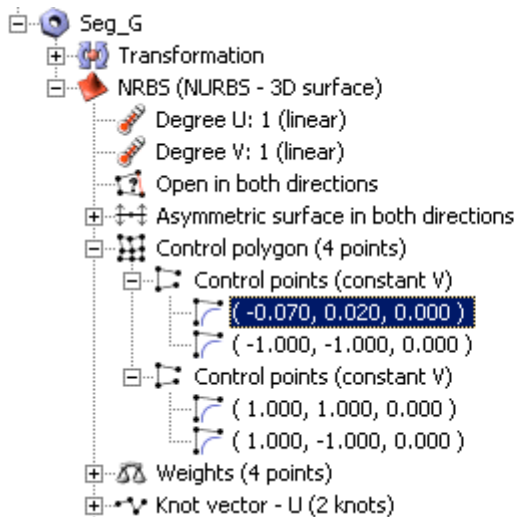
Note that number of control points (Control polygon item) is now 8 and not 16. Do the same with the Degree V parameter, and change its value from 3 (cubic) to 1 (linear). Now, Control polygon is 4 and not 8.



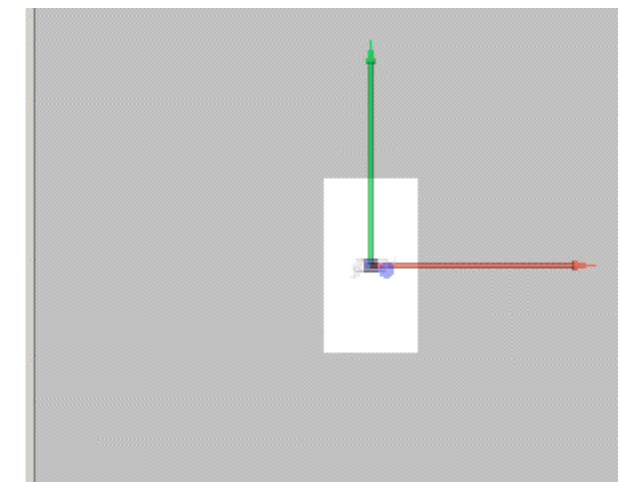
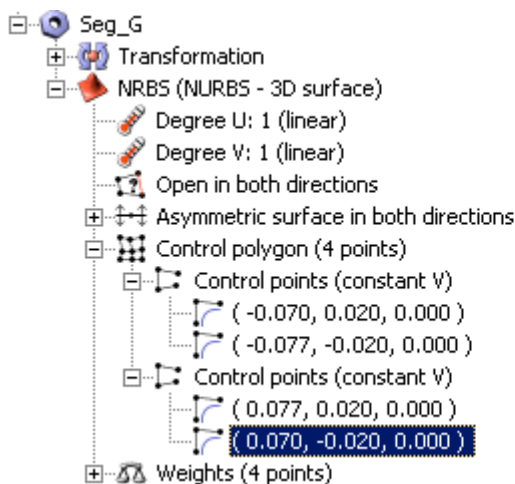
Fully expand the Control polygon item by clicking on its + sign and also by clicking on the + sign of the two Control points items. We now can see the spatial position of the four control points.



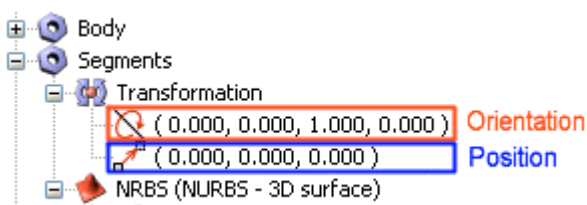
That we see here looks like that we saw before with Vertex or RGBA parameters : a group of values attached to some parameters, where parameters are here the polygon control points. Probably these values can allow to change the control points position, and by this define shape of the polygon... Hum. Just a supposition, but I want to verify this hypothesis. Try modifying values of the first control point, just to see what happen.



Effectively, something has changed in the polygon shape, which is no more a simple rectangle. The modified point is now near the center point of the body part, and is masked by it, even we can see the blue circle that allow position localisation. Think a moment what we see is not annoying, and change the values of the three other control points, as indicated in the following scree capture (on the left of the RWIE windows).

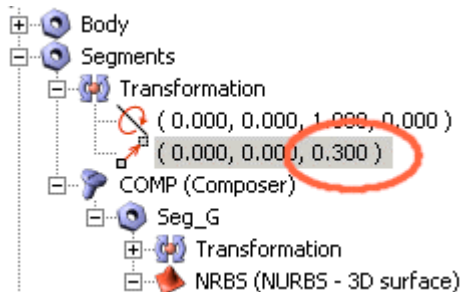


Our Seg_G objet seems to have totally disappeared of the 3D view. This is normal, however, because the four control points are now in a spatial zone included in the body part, that mask all. As you can think, make the segment visible is certainly possible. And you're right. We don't love work for nothing. As polygon is contained in the Segments container, and as this Segments container has positionning functions, we'll make profit of them. Clic on the + sign on the left of Transformation item, directly under Segments item, to see the orientation and position parameters.

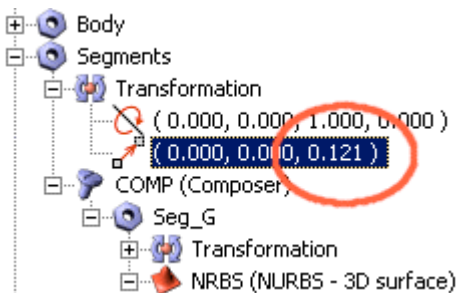
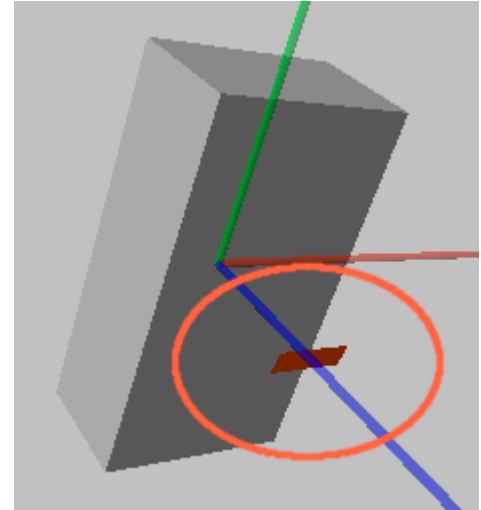


Clic once on the Position item (in blue on previous screen capture), and modify sliders position to see what happen. You probably can see that resized segemnt can move in the 3D space environnement, following X, Y and Z axis values specified by sliders position.

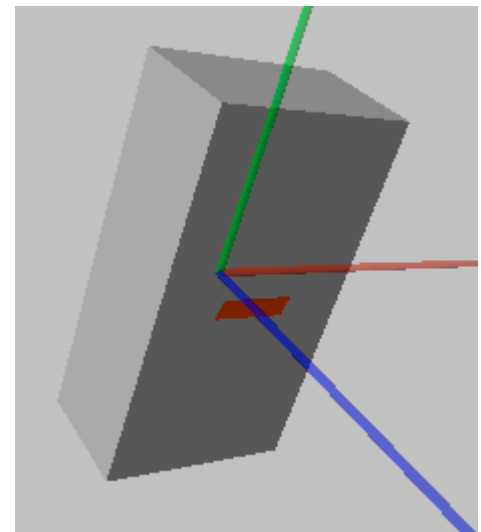
Change the third value of the Position value, to make it equal to 0.300 in a first step, and to make it equal to 0.121 in a second step.



With 0.300 value, segment is more advanced (Z axis), and we see it better. But it is too much advanced and don't touch the display body (Body rectangle).



With 0.121 value(that is Body height, plus one part of unit - 0.001), segment seems to be pasted to the body rectangle, that is what we want.

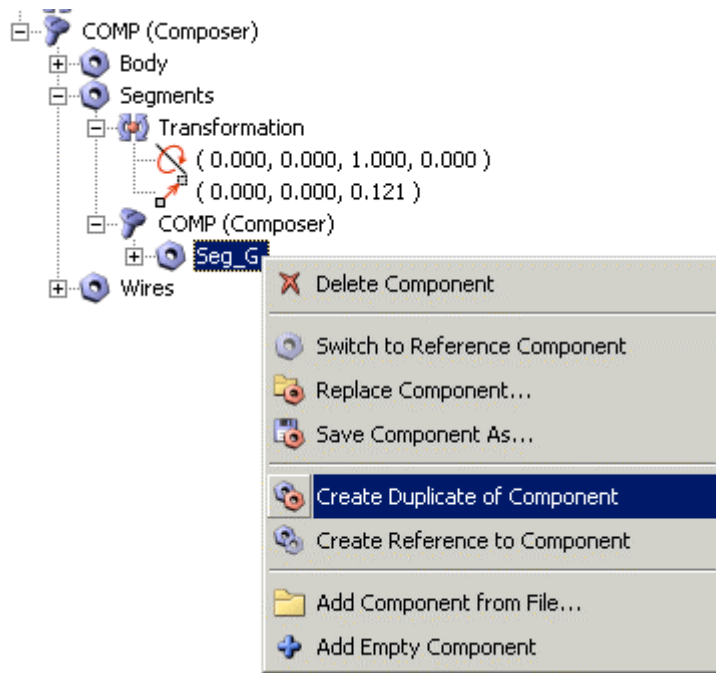


We now have a view that can remember you something, isn't it ? Possible number displaying is really limited with only one segment, so we now have to dupplicate it. Another segment upper actual one, and another copy under it.

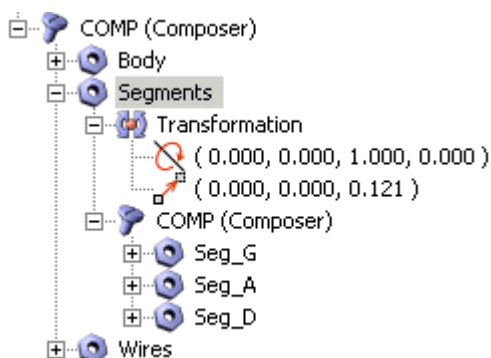
Horizontal segment duplication

Existing segment is enough for us, no need to change it at the moment. And as working for nothing is not for us, it's normal to want duplicate it. After all, text editors have all a usefull "Copy / Paste" function !

Right-clic on Seg_G item, and select Create Duplicate of Component command.



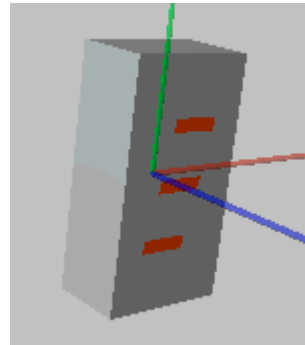
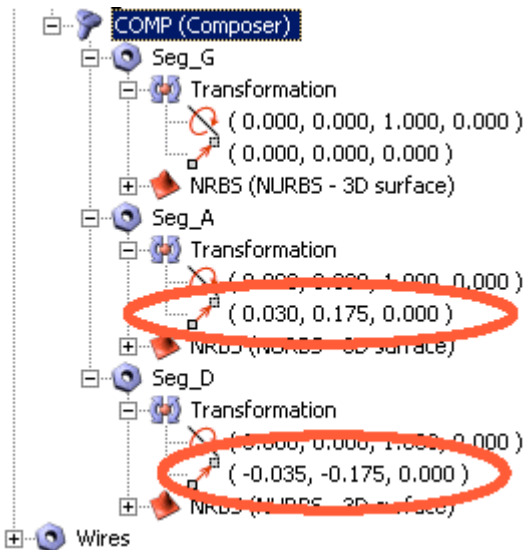
A new objet, identical copy of Seg_G object, is created and named Copy of Seg_G. Rename Copy of Seg_G to Seg_A. Again, place mouse cursor under Seg_G, and right-clc. In contextual menu that appear, select the Create Duplicate of Component command. A second copy of Seg_G is created and named Copy of Seg_G. This time, rename the copy to Seg_D.



Horizontal segments positionning

Now, put Seg_A upper than Seg_G, and put Seg_D lower than Seg_G. To respect component inclination, we'll shift the upper seg_A on the right, and shift lower Seg_D on the left.

Values for the new placement of Seg_A et Seg_D are specified in the following screenshot, you can directly use them.



I don't know if you think as me, but it seems our component have now a look that allow distinguish it from a resistor. And I don't want to stop now.

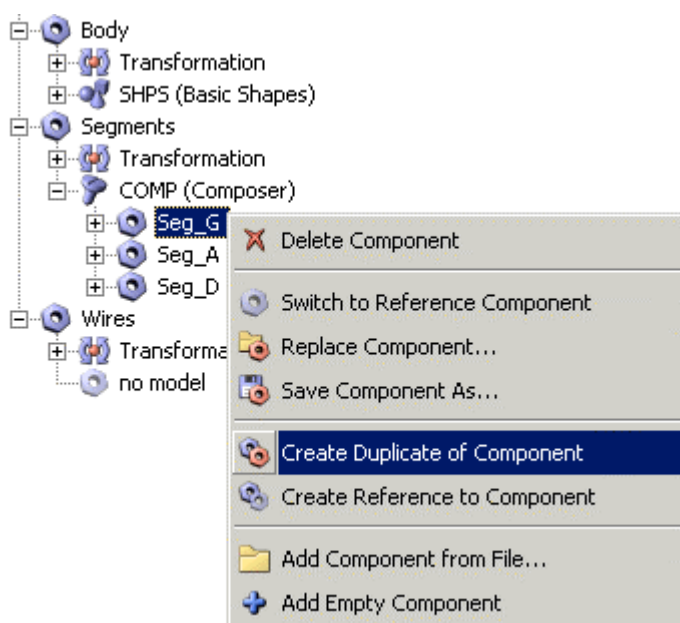
But for a coffee.

Now, we'll create a vertical segment, doing the same previous actions. And we'll do three copies of this vertical segment, that we'll be positionned in the good space position.

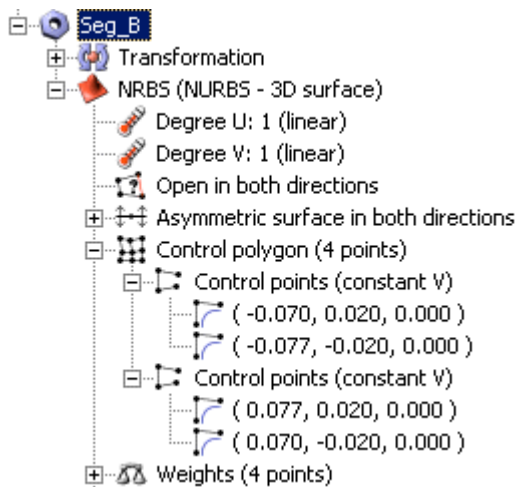
Vertical segment creation

As operation to do now are identical the operations already made, they will be less detailed. If you don't understand how to do some action, don't hesistae to re-read previous test.

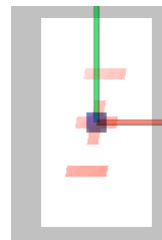
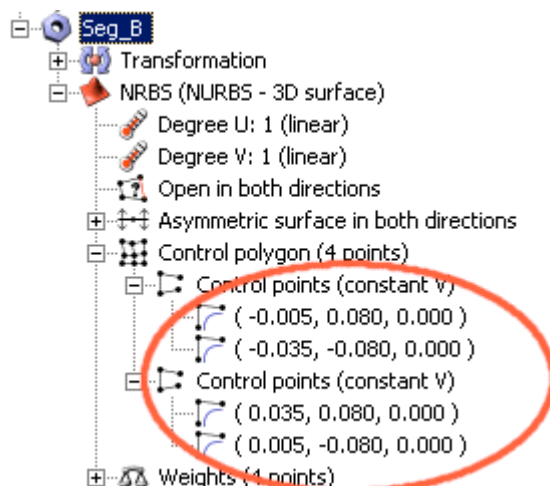
Make a copy of the Seg_G segment by right-click on its name, and by selecting Create Duplicate of Component command.



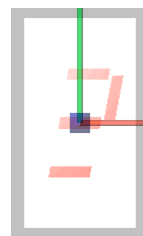
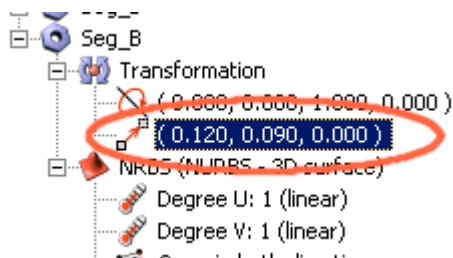
Rename the new Copy of Seg_A object, to Seg_B, and expand Seg_B branch to make visible its control points parameters (clik on all + sign as needed). You should have the following view.



Seg_B control points have the same parameters values as Seg_G, as it's a copy. We have to modify them to give the Seg_B a vertical aspect. For this, please use the values visibles on the following screenshots.



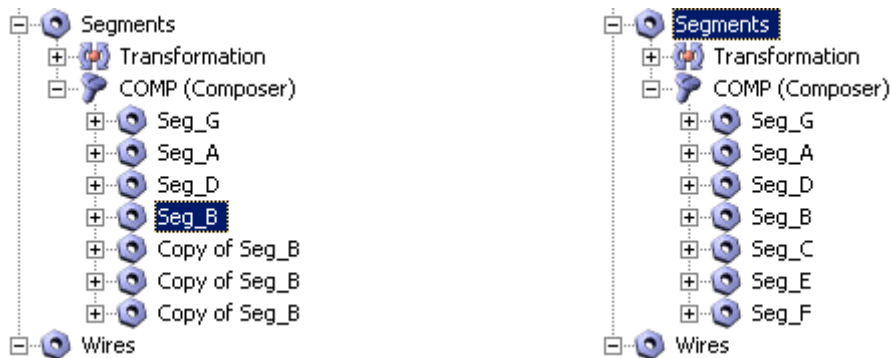
Note that again, the new component is centered and positionned with [0.000, 0.000, 0.000] coordinates. Shift this element on the right, and upper its actual value, by using following position values.



Here's a vertical segment correctly placed. Do the same for the three other copies.

Vertical segment duplication

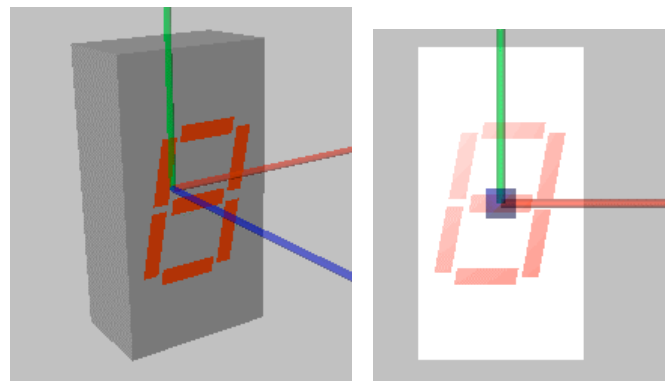
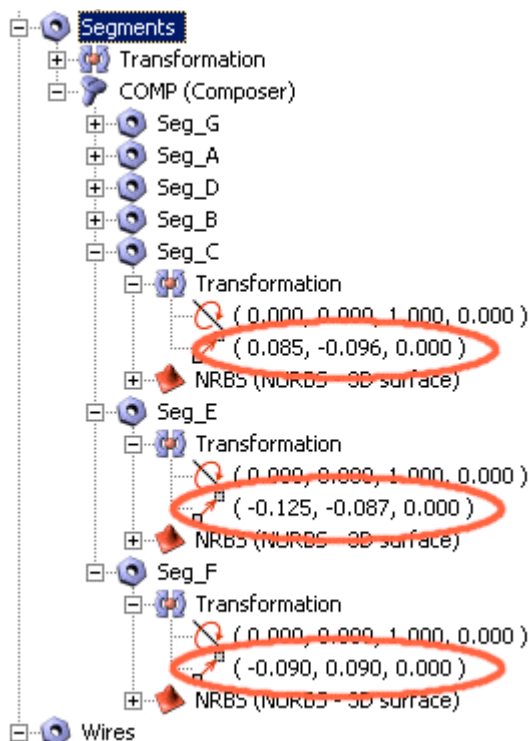
As you begin to know basic mechanisms of RWIE, you will not be surprised if I ask you to place mouse cursor on Seg_B item, to use three times the Create Duplicate of Component command, and to rename the three copies for give them the natural names Seg_C, Seg_E et Seg_F.



At this step, you can see the three new segments superposed one on the other in the center of the body part, making a "+" sign with the horizontal Seg_G segment. Do what we have to do for correct positionning of theses three segments.

Vertical segments positionning

As sometimes I'm also tired, I give you correct position values to allow you quickly place the segments.



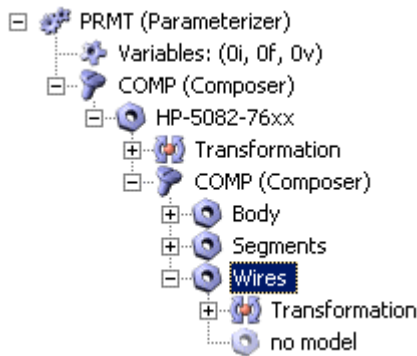
What you see if correct values have been entered.

As already said, you can move around 3D object by moving your mouse while keeping the right mouse button pressed.

All is OK on your screen ? So, felicitations ! You can take another coffee !

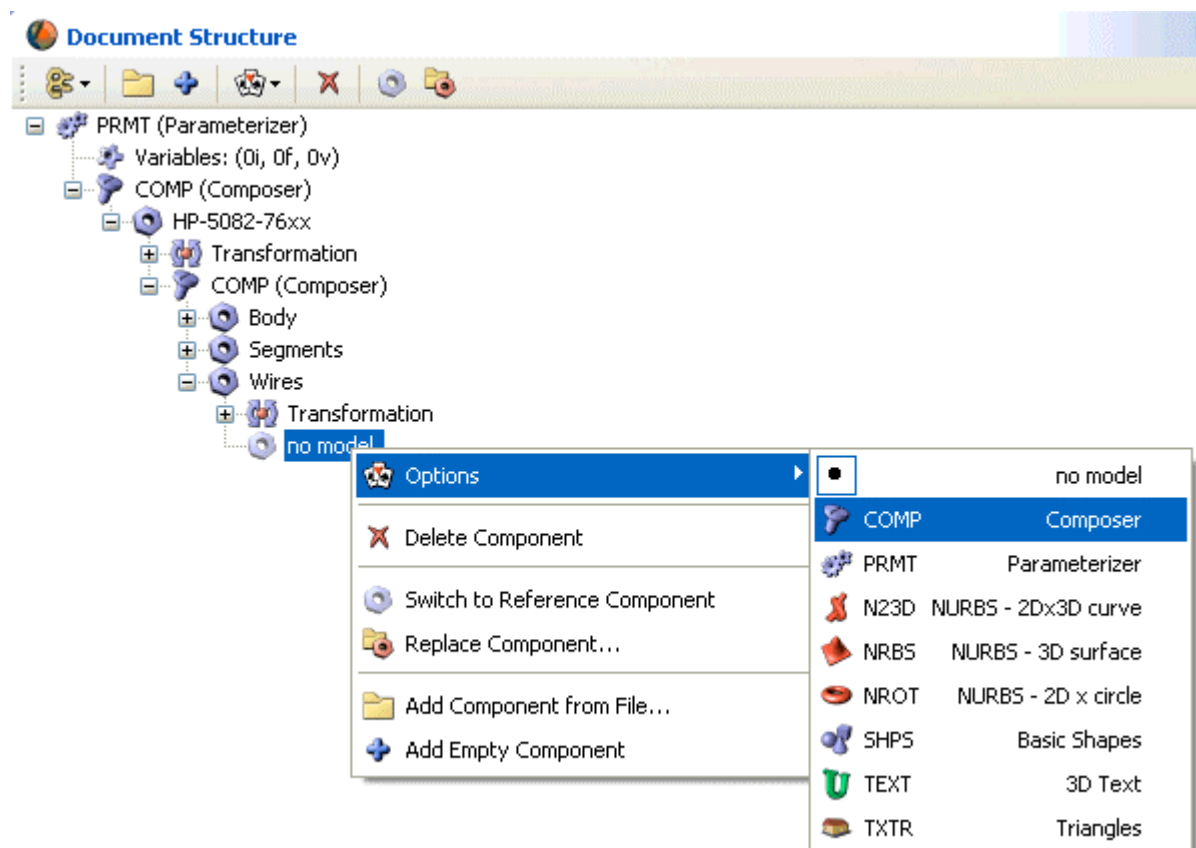
Electrical wires creation (Wires container)

A big part of the already made work will serve us to create the electrical wires, that appear to be identical in shape and color : we'll create one wires, that we'll duplicate in some quantity. And each one will be correctly positionned to be aligned. For this, we'll use simple mathematical formula. To see clearly the actual structure we work with and continue working in good conditions, we'll collapse all branches of the structure, but the Wires item, by clicking on all "-" sign as needed.

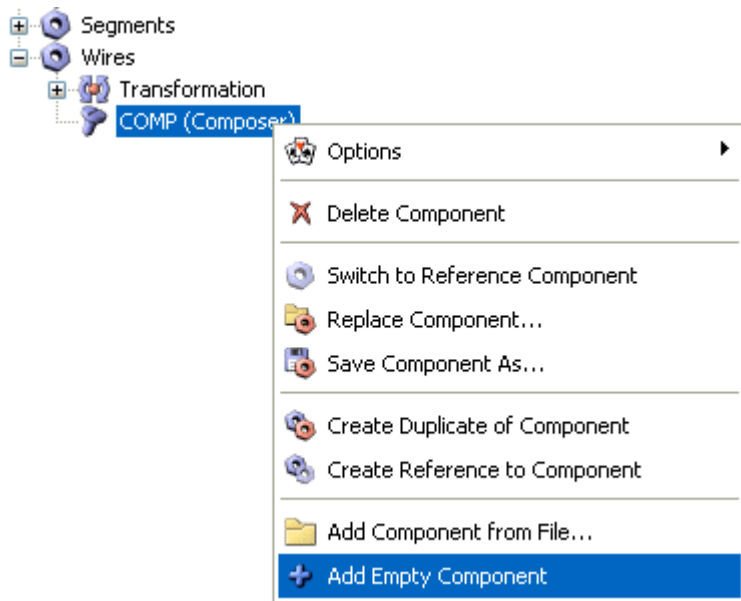


One wire creation

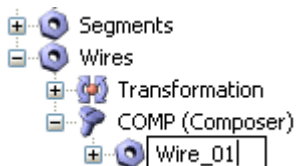
For the wires, we'll discover another 3D object type. Some solution exists for wires creation, and we can draw them as plate or cylinder. But as we all love some changes from time to time, we'll use rectangular wires, and for this, make usage of the TANK 3D simple object. As we did for segments, we'll create a container for all wires, to allow manipulate them in bloc. Right-click on No model, and select Options, and COMP.



In this COMP container, add an empty component, by right-click on COMP and by select the Add Empty Component command.

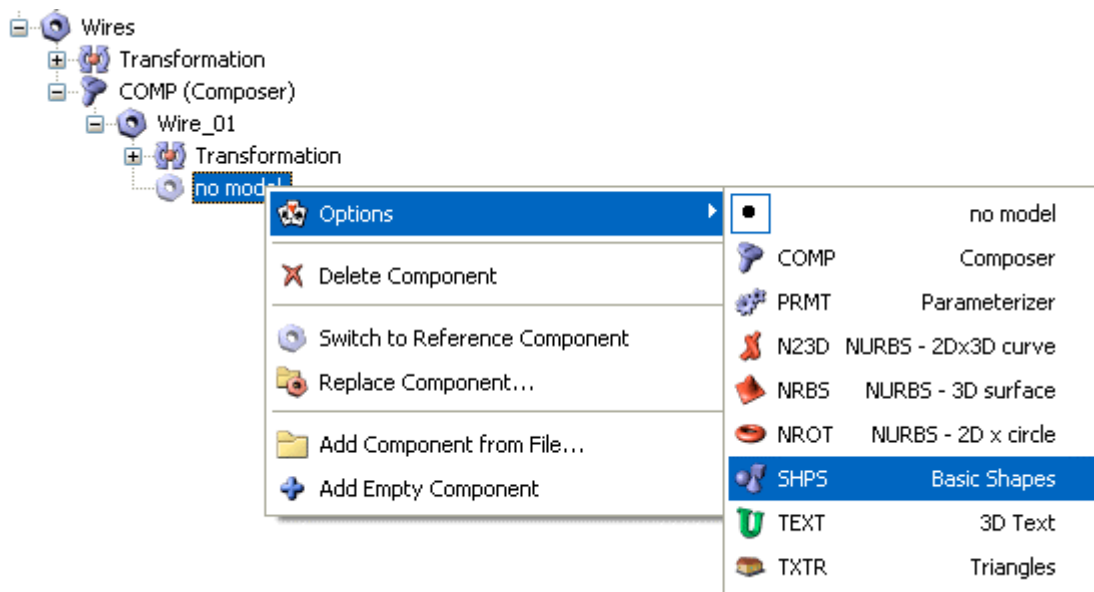


Clic on the + sign of COMP item to see the new empty created component, that is called Unnamed Component. Rename this new component to Wire_01, by clicking **once** on it to select it, and by **directly** entering the new name on PC keyboard.

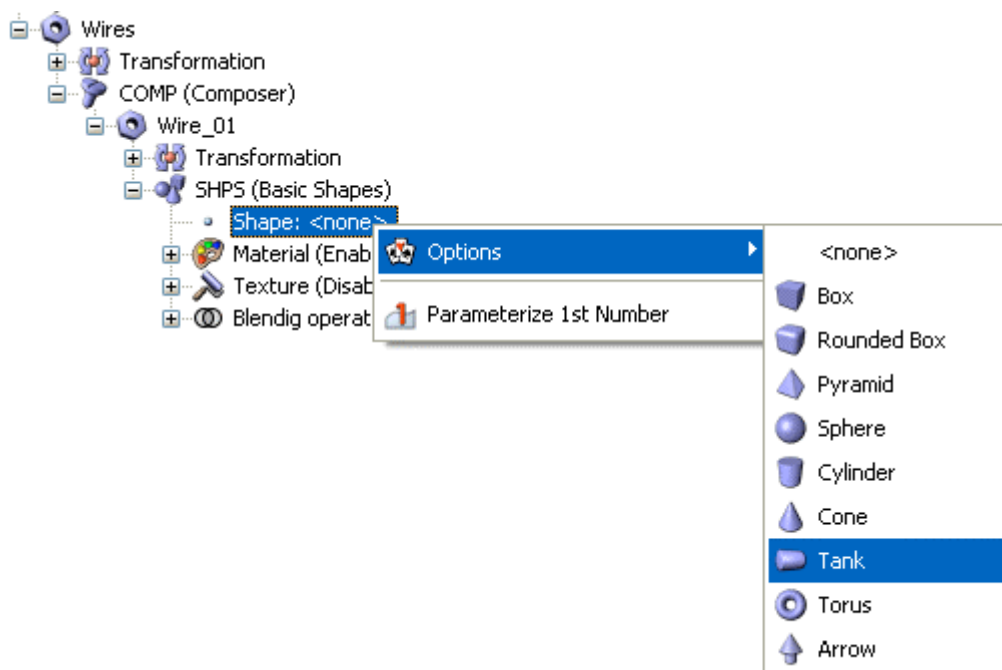


Ah, finally no need to clic twice on the actual name to enter in edit mode... Remember this tip.

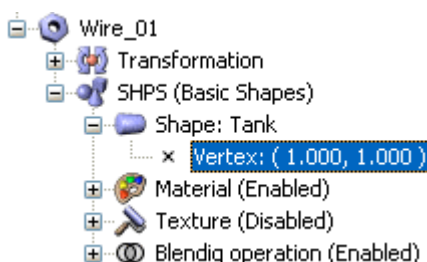
Remain specifying the component type that we want to use for the wire element. Expand the Wire_01 branch (you should know how to do, now), right-click on No model item, and select Options and SHPS (Basic Shapes) commands.



Expand the new created SHPS branch, right-click on Shape: <none>, and select Options and Tank commands.

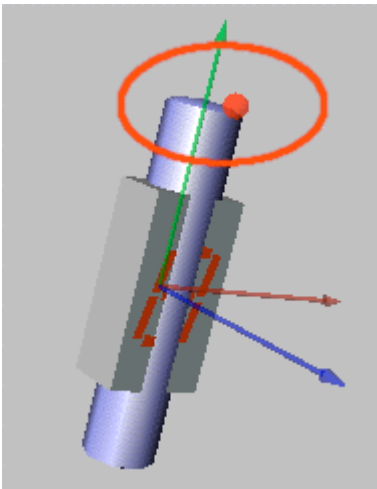


Oups ! A big cylinder bloc come over existing objects ! Do I did error ? No, you don't. The size of the new object has by default the Vertex [1.000, 1.000], that is bigger than values of others components (all less than 1.000).



If we want the display component wires can enter in the PCB holes, sure we have to reduce them a little. Play with sliders ? Why not. But more interesting is perhaps to learn how change size directly from 3D viewer.

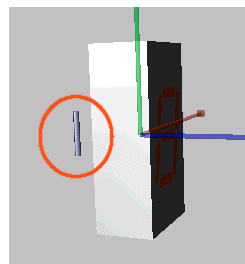
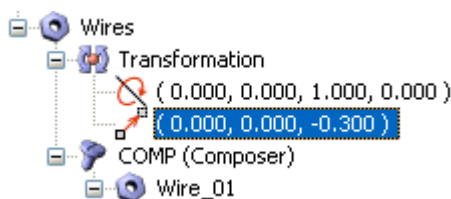
Make sure Vertex items is selected (as previous screenshot show it), clic on it if not. In the 3D viewer, clic on the blue circle placed on the selected 3D object (the circle get red), and let left mouse button pressed while you move the mouse.



By clicking on the blue circle placed on the selected 3D object (it get red), it is possible to resize component by simply moving the mouse. This method allow to get approched values, quickly and intuitively.

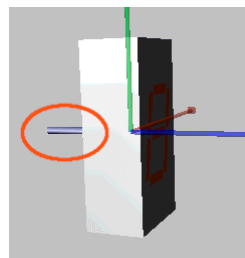
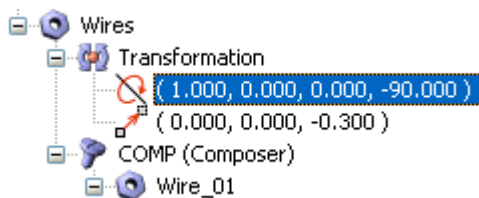
At the same time you move the mouse, look on the left side of the RWIE windows, to see and learn existing relations between mouse move in the 3D view and the numerical parameters changes.

Wire size we can see in the previous screenshot is always too big, Vertex values must be again reduced, for sure. Do a try with Vertex $[0.015, 0.100]$ values. Hum... annoying. Size is now perhaps correct, but wire is not visible. You say ? In the center ? But you're totally right ! The wire is now smaller than the body and is masked. This thing already arrived in the past, and I'm happy your memory is so good ! Well, can I suggest to move wire in the 3D space ? With the Wires bloc position parameters, perhaps ? Try this with following values.



Good ! Wire is now visible, near body. Good thing.

But all is not so pleasant, and inserting so positionned wire in a PBC will probably be problematic. Re-orientation is needed, no ? As I don't like read the user manuals and as I like make some random adjustments, I try to change Wires group orientation with the following values.



Wow ! Won on the first test ! What hazard ! Thing seems to be easy, finally. But I am OK with you : before to know, we don't know.

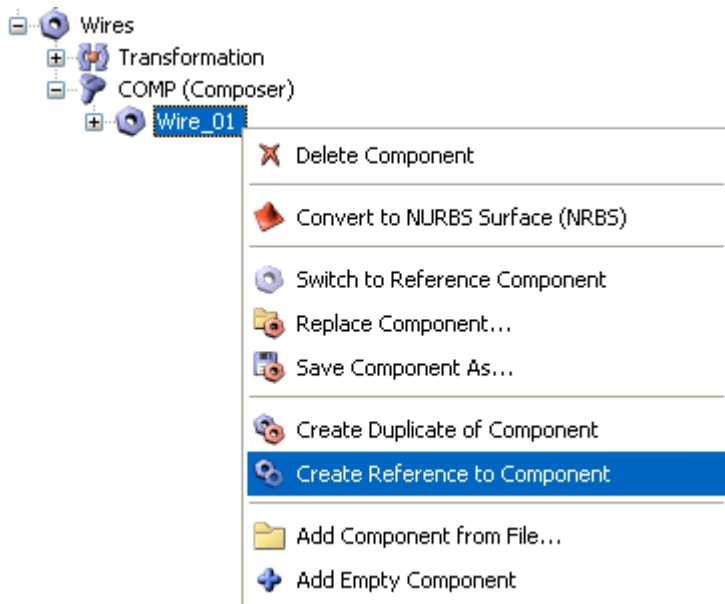
We finally did a goog wire, not exact reflect the reality, but enough realistic to let it in this state. Remember the more important is to keep simpler parts as possible, to limit memory usage and allow fast drawing.

Remain to create some other wires to give the display a more attractive and "functionnal" object.

Wire Duplication

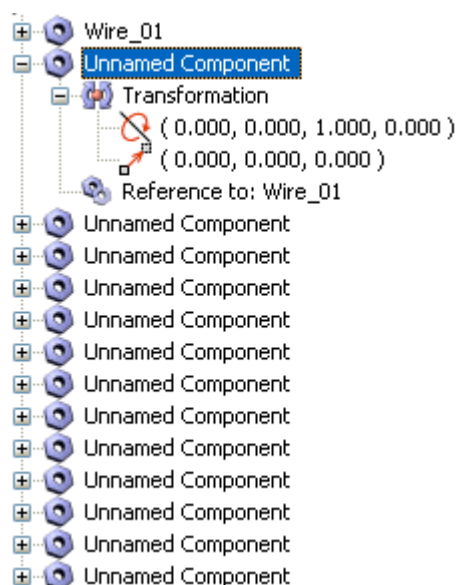
Now, we'll duplicate the only made wire, to get 14 identical wires. Display we search to modelise don't own 14 wires, but it's more easy do make 14 wires and after that delete ones that are not needed. See the display with 14 wires as an "generic" display.

Right-click on the Wire_01 item, and select Create Reference to Component command. Don't select Create Duplicate of Component command.



This time, we'll create copies that are totally identical, and only their position will differ from one to the other. This could be done for horizontal and vertical segments if number displayed and color of the leds wer fixed.

Repeat this operation 12 times, for obtaining 13 components referenced to Wire_01. If we expand the branch of one of the new created referenced components, we'll see that they are less parameters that can be modified, than the original reference Wire_01. In fact, only orientation and position parameters can be edited, all others as size and color, are linked to Wire_01 object. If we modify size or color of the Wire_01 object, all attached copies will heritate of the new values at the same moment. No need to try explain in details interest of this capability...

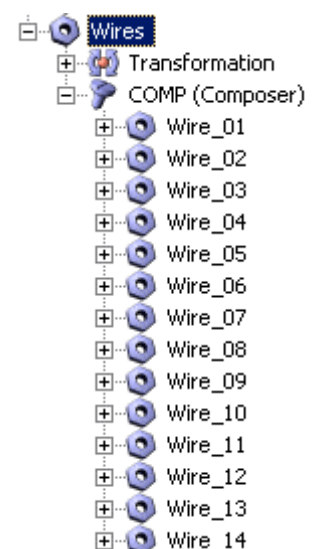


We have here a reference objet (Wire_01) at which are attached (referenced) 13 objets of same type. The 13 referenced objects are actually named Unnamed Component.

<<<----

Of course, you anticipated the ask, and you had renamed these 13 objects, from Wire_02 to Wire_14.

----->>>



At this step, we have 14 identical wires, but we see only one because they all share the same spatial position. We'll now change the position of each wire to get to rank of 7 wires each.

Wires positioning

It's totally possible to manually define the position of each wire, one by one. For exemple, by defining the position of the first wire, and by adding the same amount of spacing between each wire. This is the method, not at all productive, I used at the begining, when I discovered RWIE. As you can imagine, this is not this methode we'll use here, because another method, more "professionnal" and less consuming time, exist.

Use of mathematical formulas

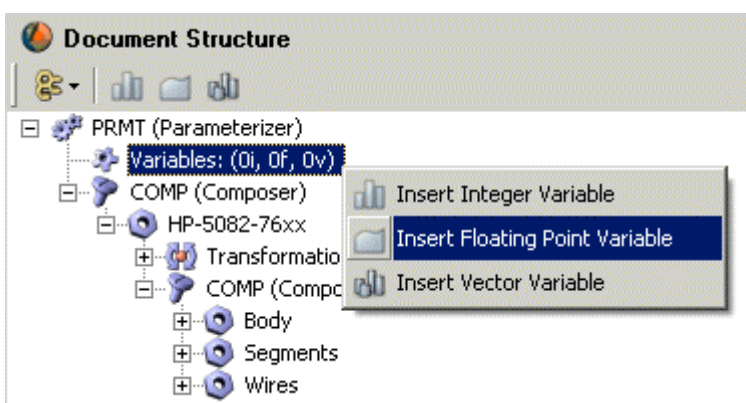
I'm sorry as you can be sorry, and "mathematical formulas" should be eradicated from all tutorials. But I have to admit that sometimes, formulas can help to simplify some tasks. Yours, too.

Before continue, you must know that parameters of lot of elements used in RWIE, as position, size, orientation or color, can take a value you manually specify or take a value stored at "another place", as in a well named variable.

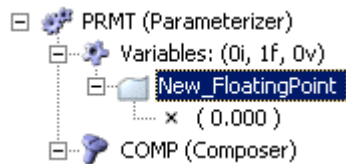
Imagine a practical case that can arrive more often than you can imagine : you have asome wires (pins) with a specified lenght, said 0.200, and you want modify the lenght of each wire. For 8 wires, you had make 8 times the same operation, as you have to modify the lenght of each wire, one after the other. Now, imagine that the parameter used to specify the wire lenght, is not a numerical value as 0.200, but is an alphanumeric string, as a name, for exemple WireLenght. In an adequat place, that can be a variables container for exemple, we decide to attach a numerical value to this name, say that WireLenght = 0.100. If now, we don't specify 0.100 as a value for the lenght of each wire, but we specify that wire lenght equal the name WireLenght, each wire component will not read the value 0.100, but the value WireLenght. Of course, this name doesn't mean something in terme of numerical value. But each component that use this name will search if a variable called WireLenght exist, in a variables container for exemple. If such variable name exist, component will use the value affected to it. Latter, if you decide to change the value of the variable WireLenght, (for exemple change initial value of 0.100 by the new value 0.150), all wires (8 in our exemple) will automatically take the new value. In other terms, only one modification can affect any components as desired.

After this little thorical text, we can return to our software manipulations.

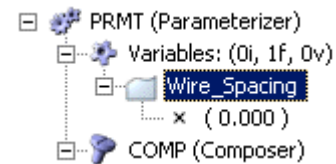
On the top of the total 3D objects structure, right-click on Variables item (under PRMT), and select Insert Floating Point Variable command.



A variable called New_FloatingPoint is created, this variable take the default value 0.000.

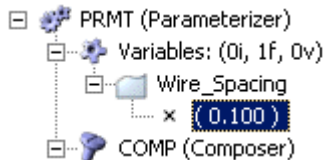


New variable called New_FloatingPoint

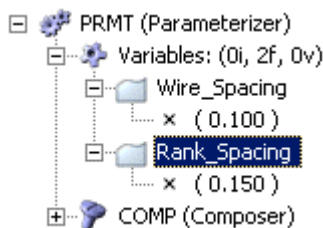


New variable renamed Wire_Spacing

As we use the english size unit, and as the wire inter-spacing is 0.100 (2.540 mm), we specify the value 0.100 to the Wire_Spacing variable.

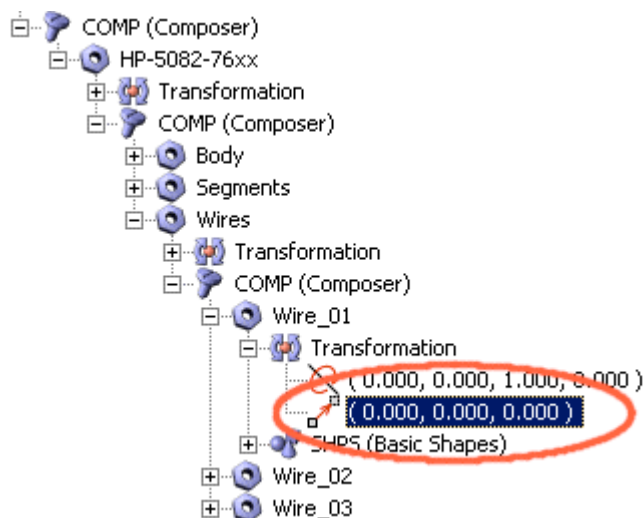


We want two ranks of 7 wires each, and the two ranks will be spaced by 0.300 in (value specified in the datasheet). Rank spacing from the reference center will be the same for the two ranks, but will be positive for one and negative for the other. So, we have to specify a rank spacing of 0.150 (positive value) for the first rank, and -0.150 (negative value) for the other. Create a new variable in the variables container, this variable will handle the space between the rank and the reference center, and will be called Rank_Spacing.

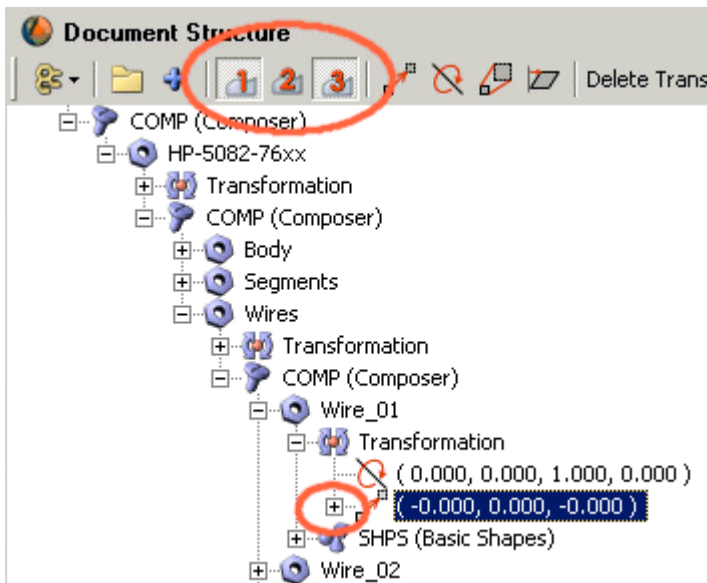


No need to make more, our Wire_Spacing and Rank_Spacing variables will be very usefull.

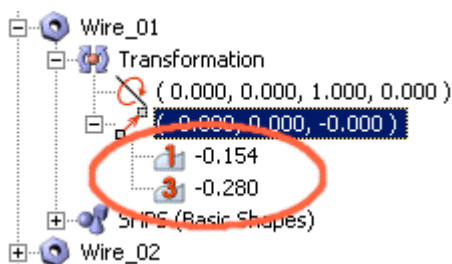
Now, make usage of our variables at the place they will be really usefull : in the wire positioning parameters. Clic on all + sign that allow to expand the Wire_01 branch, to allow acces to positionning parameters values, stored in Transformation group.



Clic on position parameters to highlight them., and clic on the "1" button (Parameterize 1st Number) et "3" button (Parameterize 3rd Number), buttons are in the structure toolbar, see following screenshot.

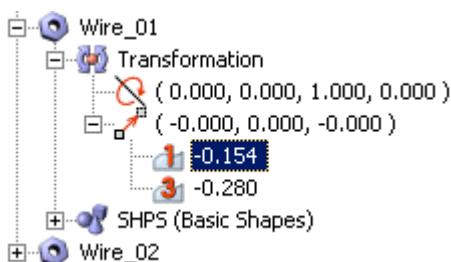


A + sign has been added at the left of the position values, meaning at least one item was added. Clic on the + sign to see what was added.



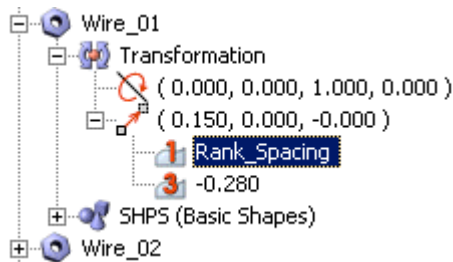
We can see that two parameters has been added, one called "1" and is attached to the first value of the position item, and another called "3" and is attached to the third parameter of the Position item. The three parameters of the position item, from left to right, represent X, Z and Y axis. The parameter "1" is so link to the X axis, and the parameter "3" is linked to the Y axis. X axis is the one we'll use to define the spacing between the two wires ranks, from the reference center. We'll decide the parameter "1" will handle the spacing value within ranks, that itself is already contained in the Rank_Spacing variable.

Clic once on the parameter "1" to select it.

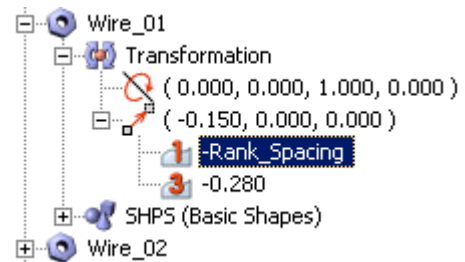


And in two separated step :

- Step 1 : write the Rank_Spacing variable name, with an underscore caractere between "Rank" and "Spacing".
- Step 2 : add a minus caractere just on the left of Rank_Spacing name.



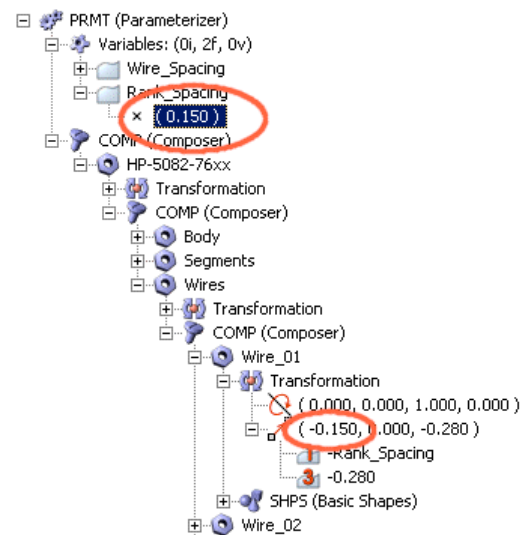
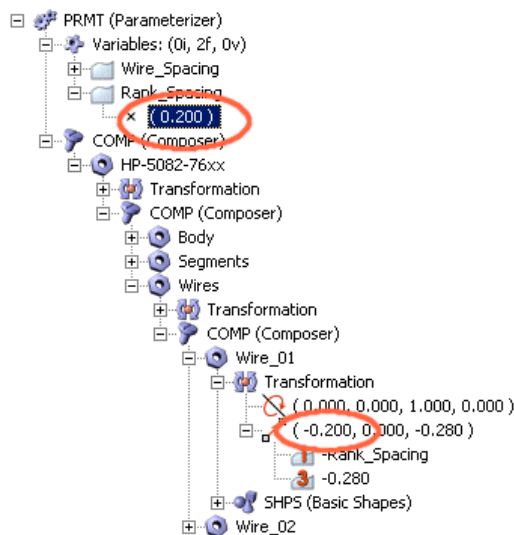
Step 1



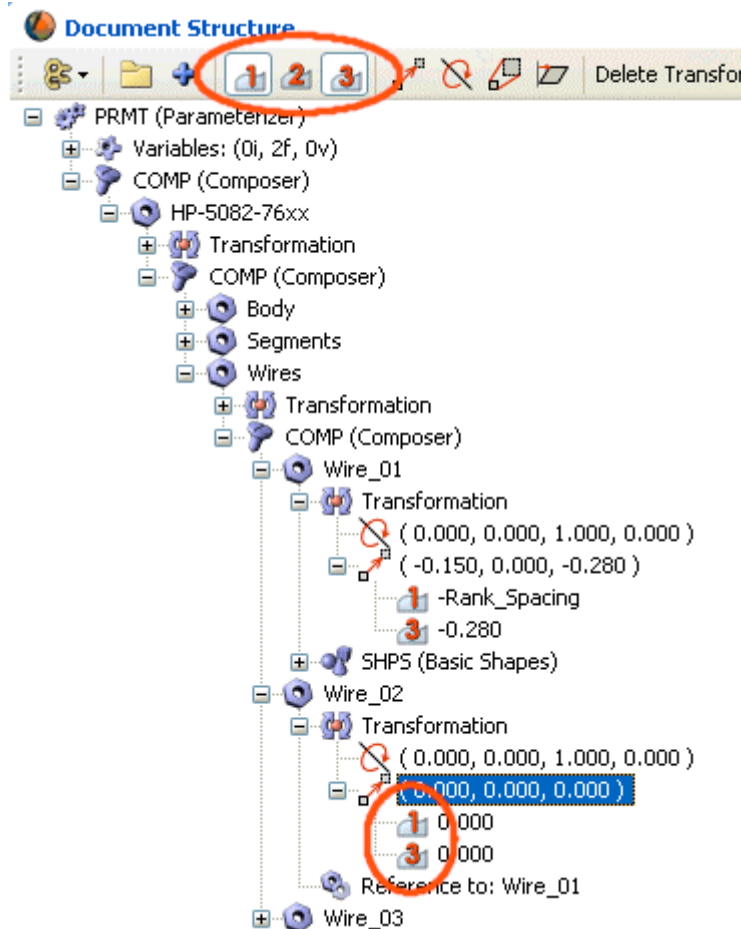
Step 2

At each step, note the change of the first value in the position items. You can also note that value of X axis has been modified, from 0.150 in step 1, to -0.150 in step 2. This value is the exact value that has been attributed to the Rank_Spacing variable.

Note totally convinced ? Try to modify the value of the Rank_Spacing variable, and see what happen on the position item of Wire_01.



Now tha all is clear in all minds, we'll repeat the operation to link all X axis value of each wire, to the Rank_Spacing parameter. Expand all branches of the Wire_02 object, clic on its position item to select it, and activate the two "1" and "3" "parameterizer" buttons of the toolbar.



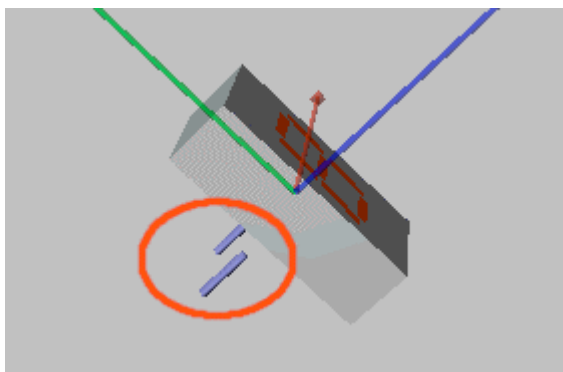
Do the same for the 12 other wires (as you'll can see, it's not so long, don't be afraid). When done, set the name "-Rank_Spacing" to parameter 1 of the first wires rank that contain Wire_01 to Wire_07 (already done for the first one), and set the name "Rank_Spacing" to parameter 1 of the second wires rank that contain Wire_08 to Wire_14.

Tip to go more quickly :

- clic on the name "-Rank_Spacing" affected to parameter "1" of Wire_01, to select it,
- hit CTRL+C on your PC keyboard (shortcut for Copy operation),
- clic on the 0.000 value of parameter "1" to modify in "-Rank_Spacing", to select it,
- hit CTRL+V on your PC keyboard (shortcut for Paste operation).

Note that for Wire_01 to Wire_07, the '-' sign is added on the left of the name Rank_Spacing, that is not the case for the other wires.

Once this work finished, you can see two wires in the 3D viewer.



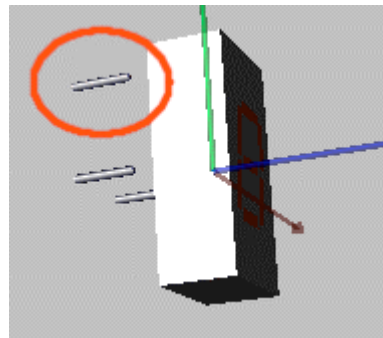
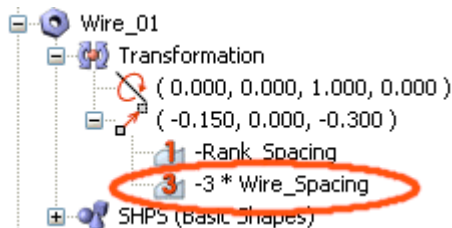
We can see only two wires, because on the two sides (ref to center 0.000), 7 wires share the same place. The only movement of the wires we made, was effective on X axis only.

To view all the wires, we have to change their Y axis position value. In our case, spacing them by an intervalle of 0.100 in. Do you remember ? It's the value we gave to the Wire_Spacing variable. Go on !

Up the position of Wire_01, that should be, as Wire_14, on the top of the rank. Clic once on the parameter "3" of Wire_01, and write following formula :

$-3 * \text{Wire_Spacing}$

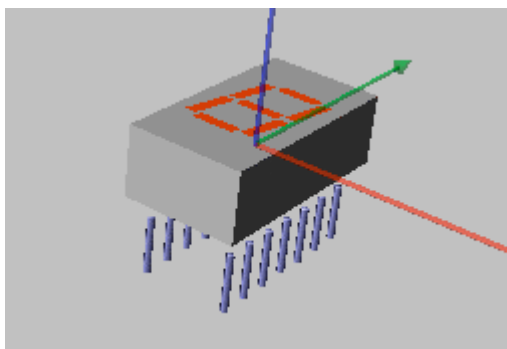
Result : we can see three separated wires in the 3D view.



Don't stop now, and put at the good place, the following (simple) formulas.

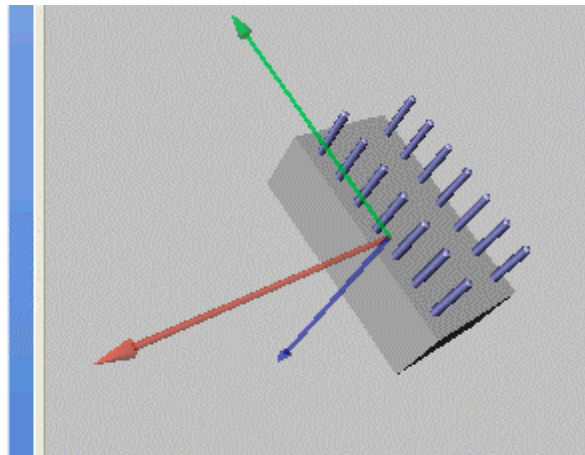
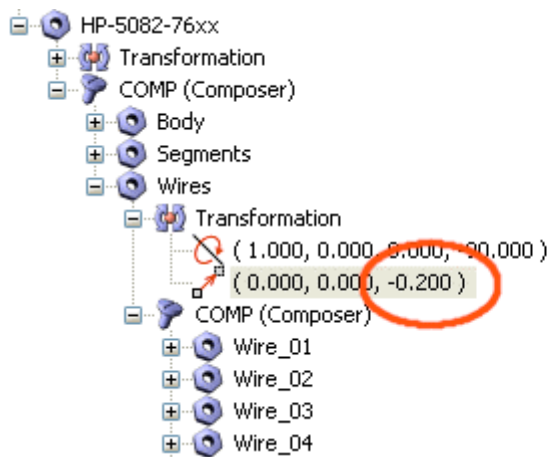
Wire	Formula for the parameter "1"	Wire	Formula for the parameter "1"
Wire_01	$-3 * \text{Wire_Spacing}$	Wire_08	$-3 * \text{Wire_Spacing}$
Wire_02	$-2 * \text{Wire_Spacing}$	Wire_09	$-2 * \text{Wire_Spacing}$
Wire_03	$-1 * \text{Wire_Spacing}$	Wire_10	$-1 * \text{Wire_Spacing}$
Wire_04	$0 * \text{Wire_Spacing}$	Wire_11	$0 * \text{Wire_Spacing}$
Wire_05	$1 * \text{Wire_Spacing}$	Wire_12	$1 * \text{Wire_Spacing}$
Wire_06	$2 * \text{Wire_Spacing}$	Wire_13	$2 * \text{Wire_Spacing}$
Wire_07	$3 * \text{Wire_Spacing}$	Wire_14	$3 * \text{Wire_Spacing}$

Result : we can see all separated wires in the 3D view.



We now have our 14 wires visible, but not really well positionned on the body, as there is too much empty space between them and main body.

To make more closer the wires and the body, we can change individually the position of each wire. Good solution ? Not sure. Try modify entire bloc of wires position, do you want ?



Now better.

So, we can stop here, if you want, and our final object can be exported as this.

But what about color and number to display ?

Note : following steps are not required to do 3DS export of the component, and you can direct jump to 3DS export section if you're not interested by number and color changing.

Segment color selection

Easy ! You just have to change the value of color parameters in the Materiel / RGBA item. Yes, you're right, this can be done as this. But do you want this for each led segment ? At least 7 changes for a unique parameter change.

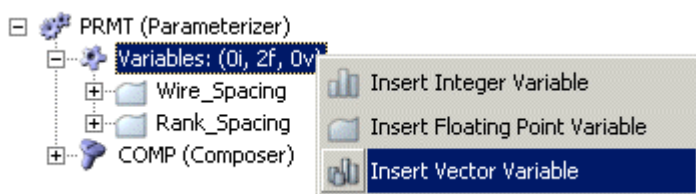
Imagine you want change the red color by the green color : you have to reduce red value and increase green value. Total : 14 changed values for 7 segments. No need to take some minutes and thinking about a simpler solution ?

A suggestion : what happens if we create a new variable called Red, and if we use in each RGBA line ? By modifying the Red variable, all segments will be canged at the same time, no ? Well, create now three new variables, called Red, Green and Blue.

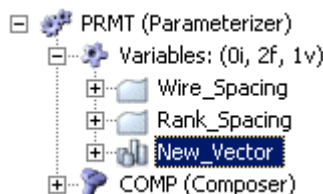
But...

But there is another more elegant solution. Yes, RWIE software developper already though about this "option", that is not unique at our school case. And he implemented a special variable of type "Color" that need only one line of parameters.

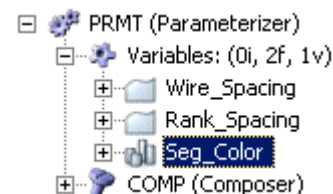
Right-click on the variables item, and select Insert Vector Variable command.



A new variable called New_Vector is added after the two existing Wire_Spacing and Rank_Spacing variables.



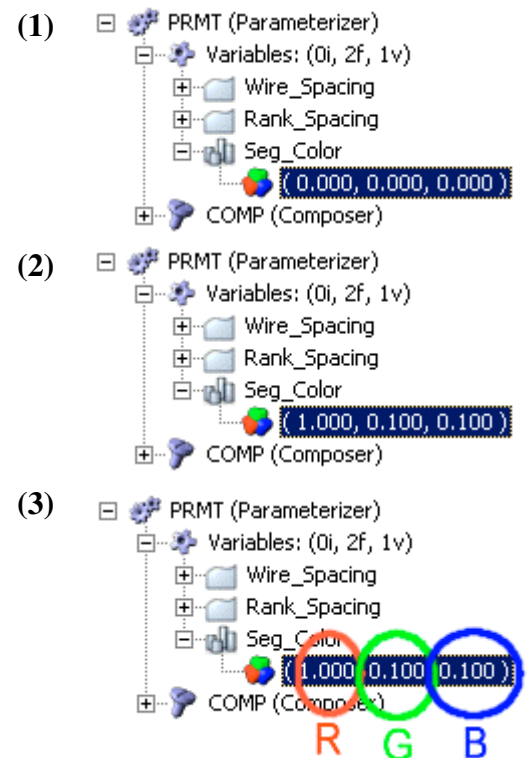
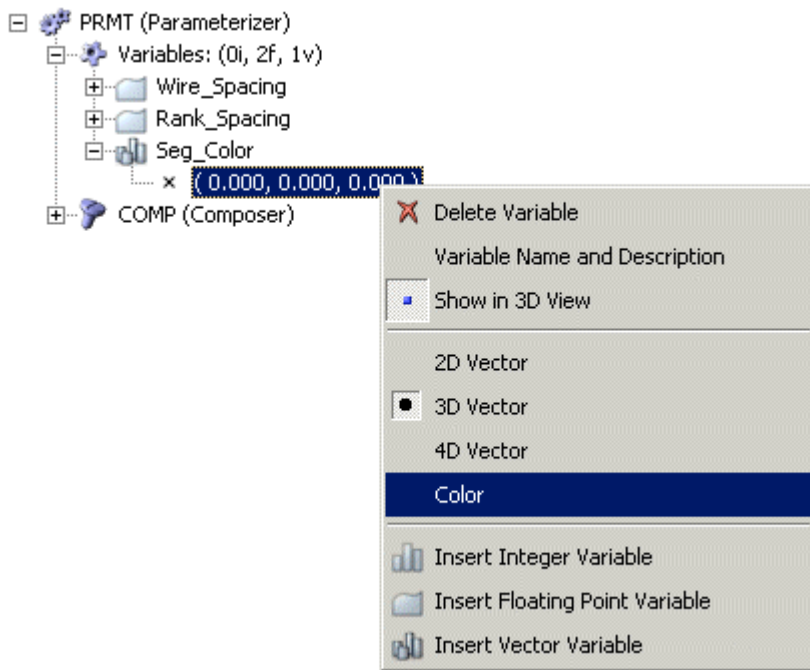
New variable "New_Vector"



"New_Vector " Variable renamed to "Seg_Color"

Expand Seg_Color by clicking on its + sign. Right-click on the line that contain the three 0.000 values, and select Color option (actually fixed to 3D Vector). Effect of this we'll be that software know that the three values attributed to Seg_Color will be Red, Green and Blue values.

Try changing Seg_Color values with [1.000, 0.100, 0.100].

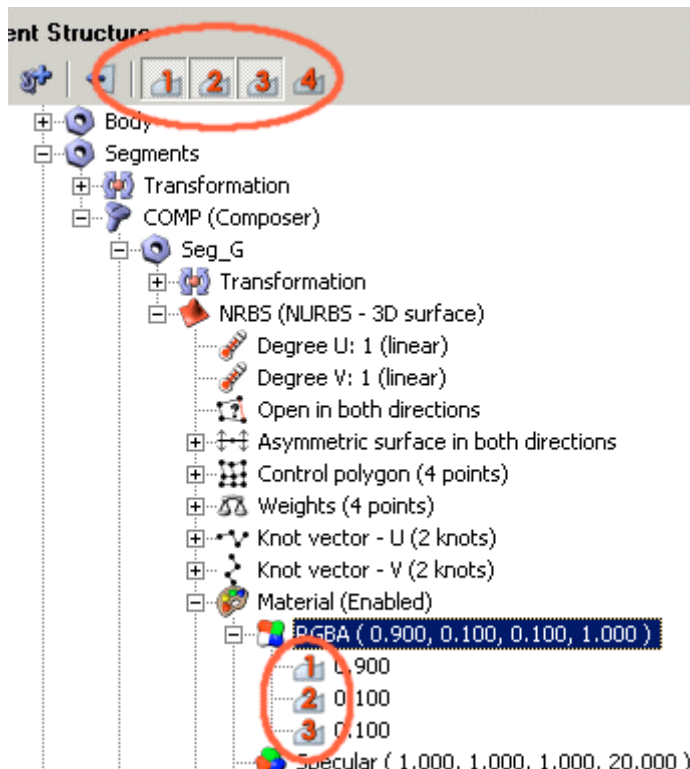


(1) : Seg_Color was specified as a Color type variable.

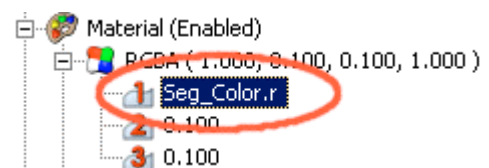
(2) : Seg_Color has a red dominant color.

(3) : Red value is stored in the first parameter, Green value in the second, and Blue value in the third.

Expand Seg_G to see RGBA line. Clic on the RGBA item, and activate "1", "2" et "3" buttons of the toolbar. When done, clic on item "1" to select it, and write following text : Seg_Color.r

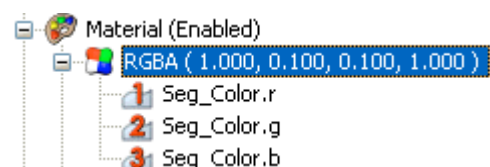


1 - Cliquer sur la ligne "1" pour la sélectionner, puis taper le texte Seg_Color.r



2 - Cliquer sur la ligne "2" pour la sélectionner, puis taper le texte Seg_Color.g

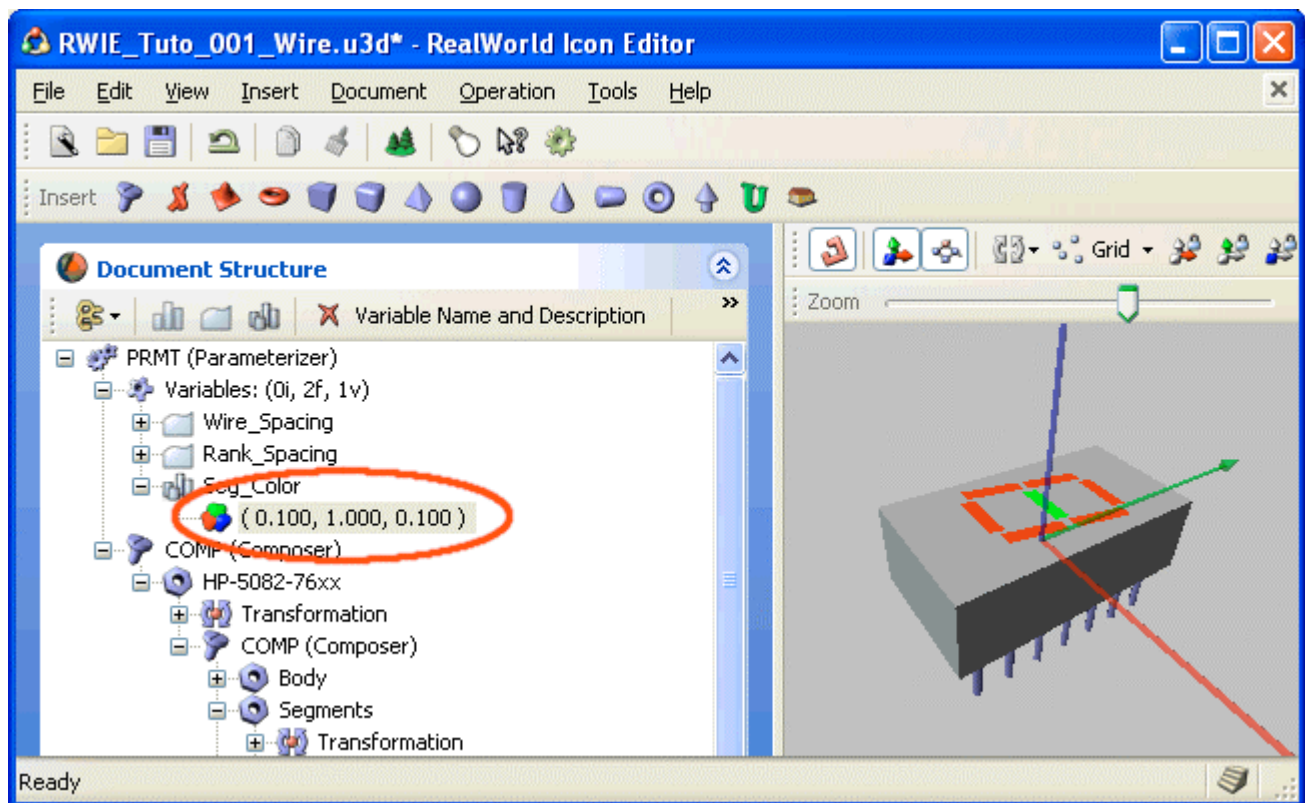
3 - Cliquer sur la ligne "3" pour la sélectionner, puis taper le texte Seg_Color.b



Explanation :

Seg_Color variable, of Color type, own three parameters that are Red, Green and Blue. If we use the value `Seg_Color.r` (Seg_Color and "R" character separated by a point) in a formula, this mean we refer to the parameter Red of the Seg_Color variable (R for Red). If we want use the blue value of the Seg_Color variable, we have to use the value `Seg_Color.b` (b for Blue). Same for Green parameter, that can be accessed trough the value `Seg_Color.g` (g for Green).

Test : modify Seg_Color variable to get a Green dominant color. Set Red value to 0.100, and set value of green to 1.000. Don't touch to the blue value. You should see this :



Centered Seg_G segment color has changed and is now green, and all other segments kept their red color. And this, without modifying the parameters of color of the Seg_G RGBA item. It's easy to understand that if we modify all parameters "1", "2" et "3" of each RGBA items of the other segments, all segment will light in green, too.

Do you want now change all segments for allowing same action for each, when Seg_Color values are changed ? You can do it.

But if I can make a suggestion, wait a little, because in the next section, we'll discute about the number to display, and the parameters lines will be more exploited.

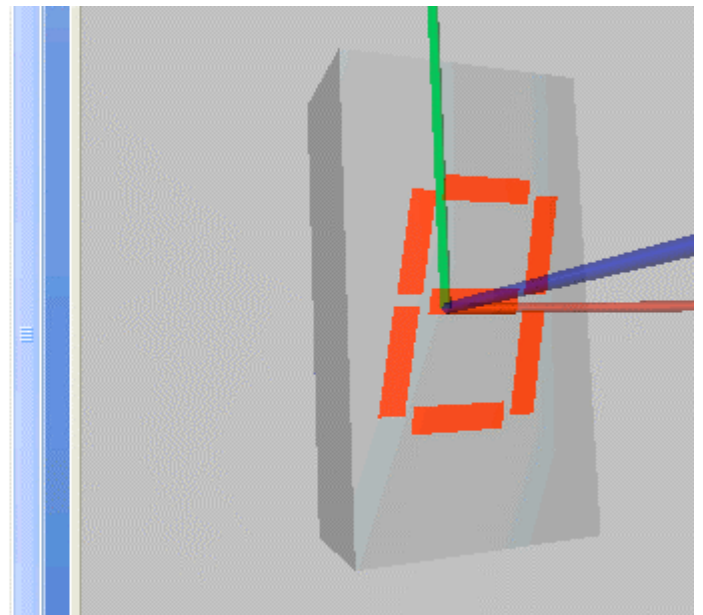
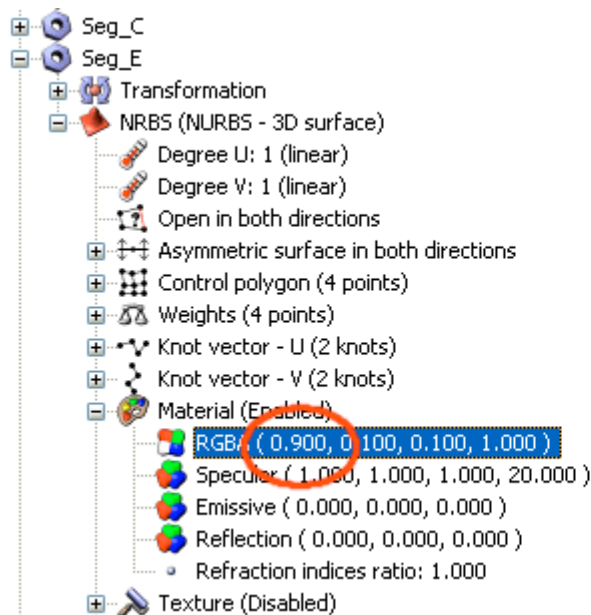
Choix du chiffre à afficher

Ask : how to "light on" or "light off" segments ?

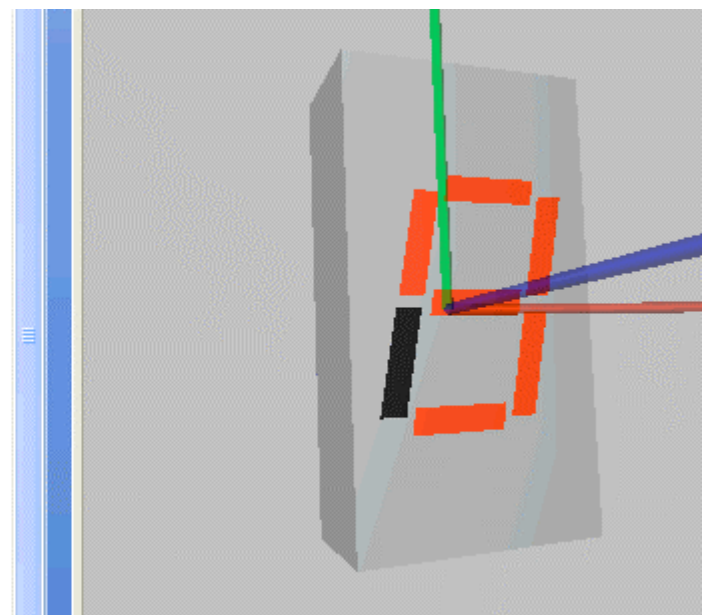
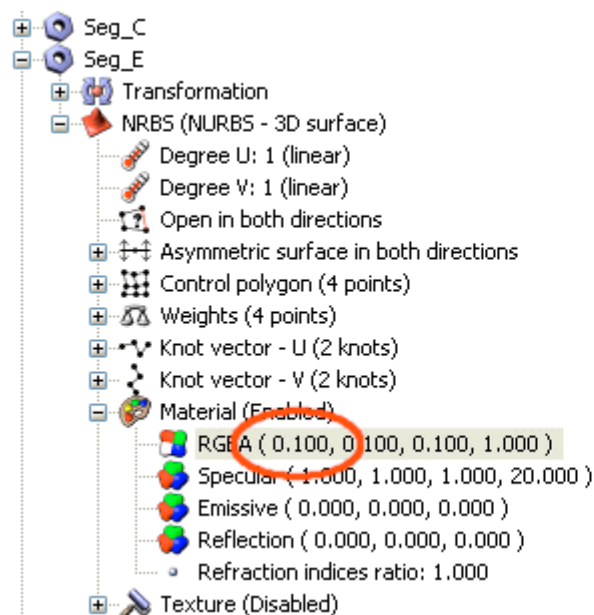
Answer : by reducing color values to a minimum for Off segments.

Example : switch off Seg_E, by reducing red value.

Here, all segments are On, red color value of each segment is high :



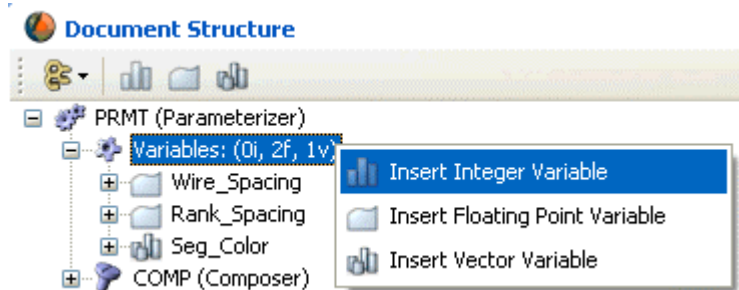
Here, all segments are On but Seg_E, because its red value is low :



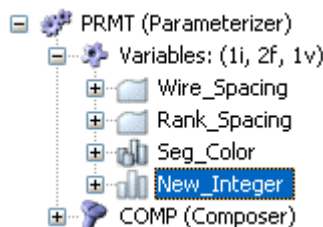
What ? It's all ? Change one parameter is enough to change displayed number ? In this case, yes. But try now to display "1" number. You must change four parameters, not only one, as segments A, D, F et G have also to be switch off. Not really easy if changes occur often.

As we did it for the wires, and as we did it for Seg_G color, we'll now define a numeric variable that we'll use in mathematical formulas. This will these formulas that will allow change of desired segments.

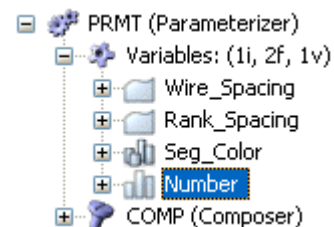
Right-click on Variables items, and select Insert Integer Variable command.



A new variable called New_Integer is added to the existing variables.

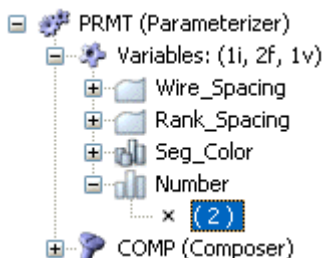


New "New_Integer" variable added



"New_Integer" variable renamed to "Number"

Expand Number branch, and set the Number value to 2.



Now our Number variable is created , use it in the following formulas :

For the parameter "1" of the Seg_G RGBA line, write this :

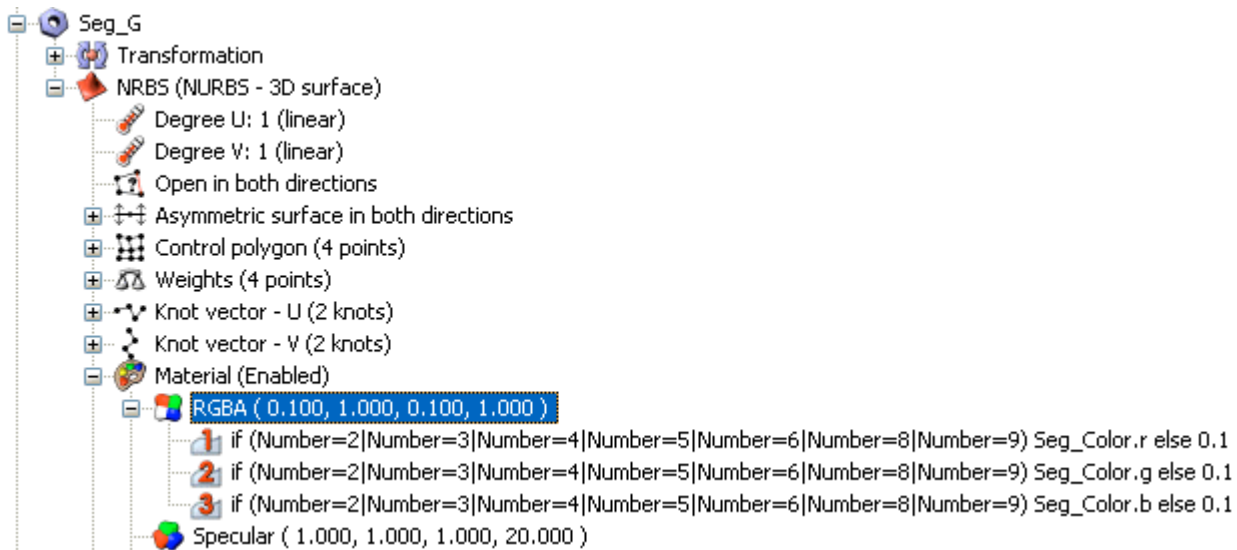
```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.r else 0.1
```

For the parameter "2" of the Seg_G RGBA line, write this :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.g else 0.1
```

For the parameter "3" of the Seg_G RGBA line, write this :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.b else 0.1
```

These formulas are read and executed (interpreted) when the component read the value of its Red, Green and Blue parts values.

Explanations

Formulas can contain keyword, as `if .. (then) .. else`, that allow to use a value that depend of an external condition, based or not on the use of one or more variables. Here, we look if the parenthesis contain yes or not, a true evaluated condition. If condition is true (Number variable equal to 4, for example), so the parameter "1" take the value of `Seg_Color.r`. If condition is false (Number variable equal to 7, for example), so the parameter "1" take the value 0.1 (or 0.100, it's the same thing). In fact, vertical bars, placed between equal evaluation (for example between `Number=4` and `Number=5`) stand for "or" function. Parenthesis contain is true if Number variable equal to 2, or if equal to 3, or if equal to 4, etc. Evaluation return false if Number value is not contained in specified values.

Note that we can also use "more than" or "less than" character, that allow in some case to simplify formula. In actual case, the two following formulas Formula V1 and Formula V2 are equivalent and return the same result if Number has a value comprised between 0 and 9 (other values don't have any sens).

Formula V1 :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.r else 0.1
```

Formula V2 :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number>7) Seg_Color.r else 0.1
```

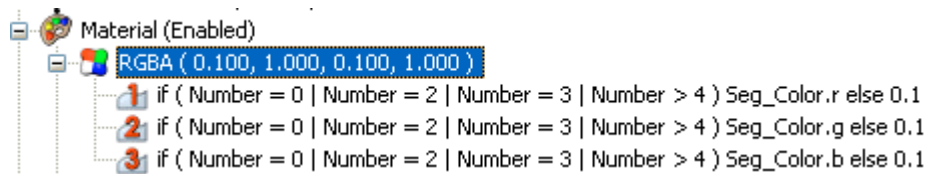
Apply now this type of formula at each segment RGBA parameters, taking care of what segment has to be lightened for a specified Number value. For example, `Seg_A` is On when we want display the numbers 0, 2, 3, 5, 6, 7, 8 and 9. `Seg_E` is On when we want display the numbers 0, 2, 6 et 8.

Segment Formule requise

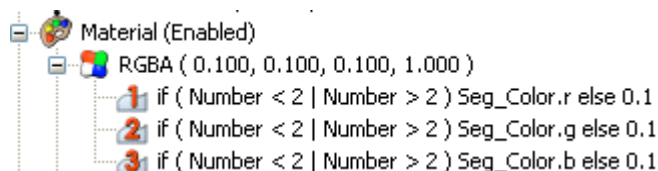
Seg_A	<code>if (Number = 0 Number = 2 Number = 3 Number > 4) Seg_Color.r else 0.1</code>
Seg_B	<code>if (Number < 5 Number > 6) Seg_Color.r else 0.1</code>
Seg_C	<code>if (Number < 2 Number > 2) Seg_Color.r else 0.1</code>
Seg_D	<code>if (Number = 0 Number = 2 Number = 3 Number = 5 Number = 6 Number = 8 Number = 9) Seg_Color.r else 0.1</code>
Seg_E	<code>if (Number = 0 Number = 2 Number = 6 Number = 8) Seg_Color.r else 0.1</code>
Seg_F	<code>if (Number = 0 Number = 4 Number = 5 Number = 6 Number = 8 Number = 9) Seg_Color.r else 0.1</code>
Seg_G	<code>if (Number=2 Number=3 Number=4 Number=5 Number=6 Number=8 Number=9) Seg_Color.r else 0.1</code>

Previous formulas are for parameters "1" only, that are relative to Red color. They are identical for the three parameters "1", "2" et "3", except that we use `Seg_Color.g` for the parameter "2" and `Seg_Color.b` for the parameter "3". See previous or/and next screenshots.

For Seg_A :

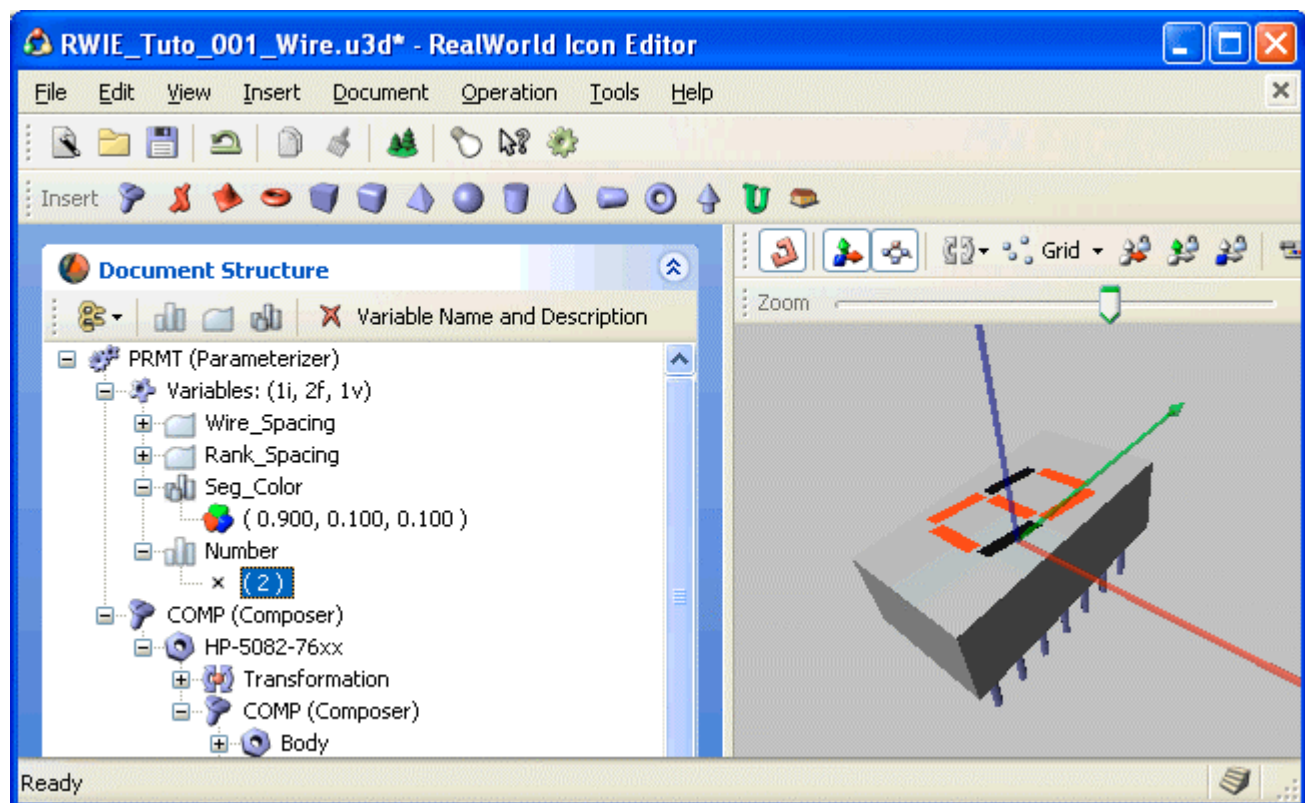


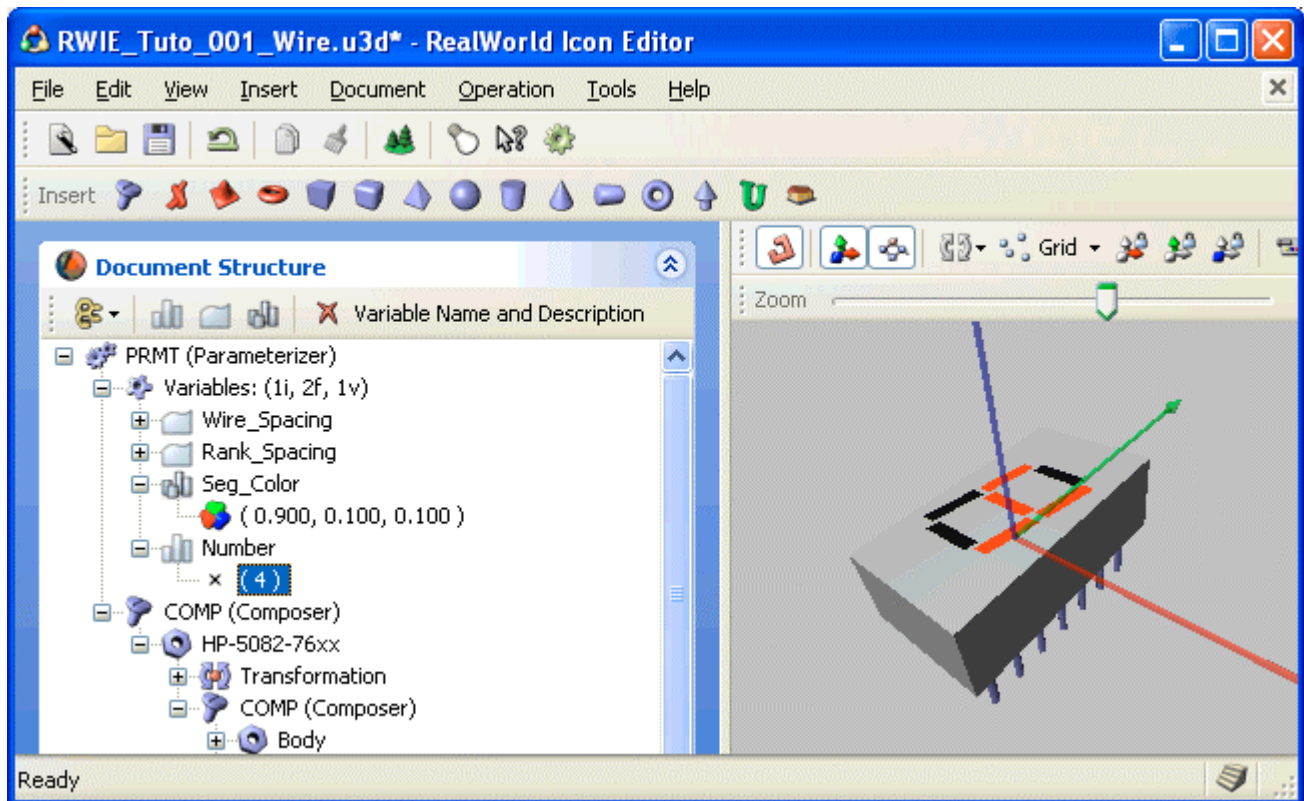
For Seg_C :



Important ! Formulas are case sensitive, min and maj must be repeated. If you named a Number variable (with maj N) in the variables container, and if after you reference this variable with an name without maj (number, with a min N), formula will not be executed because it contain an invalid / inexistant element.

Now, the value of the Number variable is propagated to the displayed number itself.

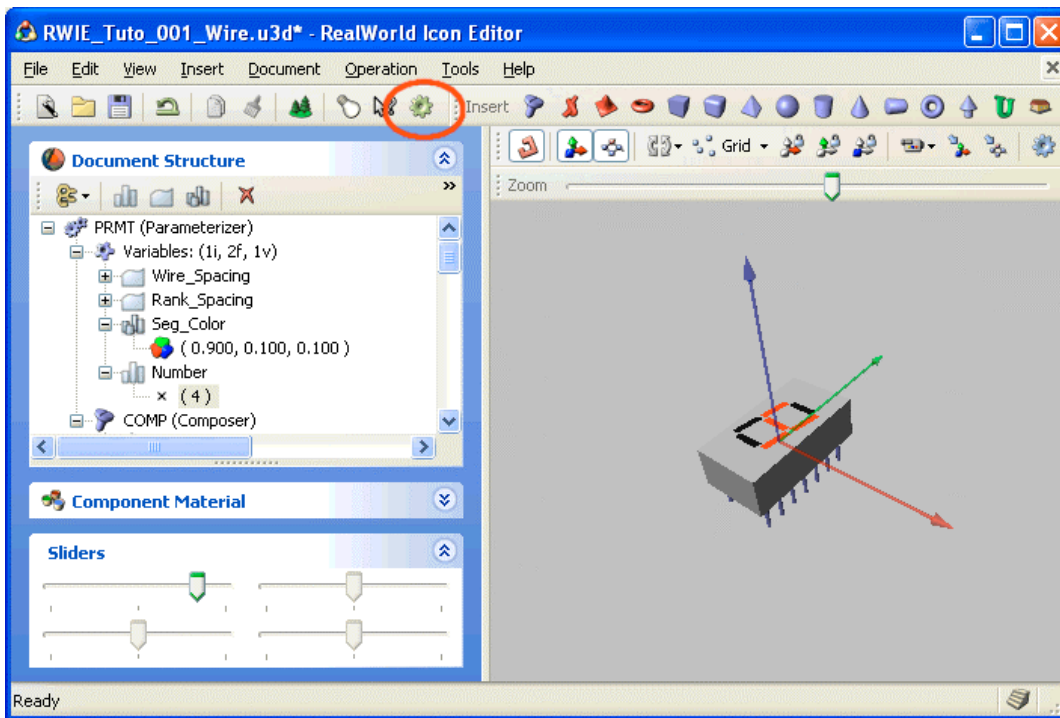




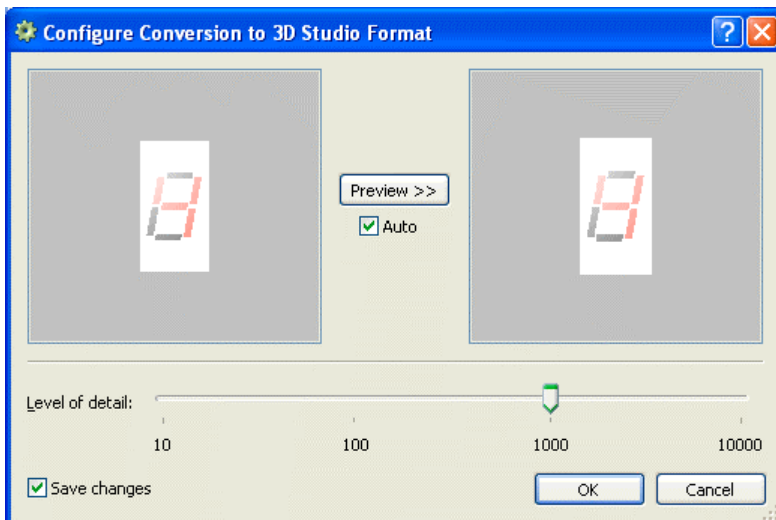
3DS export

3DS export require modified copy of the RealWorld Icon Editor software. Version actually downloadable on the rw-designer site, that is V2006.2, don't allow 3DS export. Before make 3DS export, you have to follow the procedure described on the following page : http://www.sonelec-musique.com/logiciels/rwie/rwie_3ds_export.htm

Object is enough pleasant for you and you want make your so waited 3DS file for use it in the Proteus / Ares 3D viewer ? Simply clic on the "Save as 3D Studio file" button, on the file toolbar.



A window appear, that allow you to specify the level of triangulation details for the exported file.



In main cases, a value comprise in the 100 .. 1000 range is enough to get good results. For components that have rounded parts, some tests are necessary to get the best compromise between file size and visual appearance.