

R4 Systems Inc.

3D Viewer

The 3D Visualisation Tool (3D Viewer) in ARES provides a way to extrude a layout and view the board as it would appear in real life. This is extremely useful as a design aid during board layout.

The main features of the 3D Viewer include:

- Realistic representation of board layout in three dimensions.
- 3D footprints pre-supplied for most all of the ARES libraries.
- Preset view buttons provide instant access to top/bottom/front/back etc. of the board.
- Intuitive navigation allows you to quickly traverse the board and zoom around areas of interest.
- Phased rendering allows navigation of board and components while complex silkscreen, tracking etc. is being computed.
- Caching avoids re-computation of the board when the layout is unchanged.
- Code generated resistor banding according to component values.
- Simple script based technology with live preview makes it simple to create your own 3D footprints.
- 'Update Visual Models' command automates the process of applying 3D footprints to existing layouts.
- Model import via the standard *.3ds file format allows the creation of exceptionally detailed footprints in most common 3D tool suites.

Overview

It is likely that the most important change that you might want to make is to customize the appearance of the footprints appearing in the viewer. The good news here is that we have done most of the work by including 3D models across the range of pre-supplied libraries, meaning the only customization required is for footprints that you have created yourself. We have also taken pains to make it easy to create your own 3D model information from within ARES and provided an import facility should you prefer to work inside a foreign 3D graphics package. Finally, a basic silkscreen extrusion algorithm is used where no 3D information is supplied in order that some (hopefully sensible) view is presented regardless.

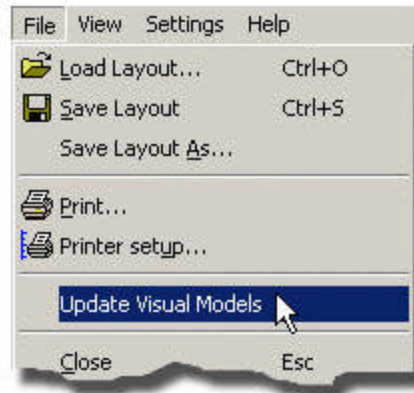
Applying 3D Data to Existing Designs

When you invoke the 3D Viewer command on a layout you have created in an older version of the software or containing footprints that you have created yourself you will see that some (or most!) of the 3D footprints are colored red. This indicates that a 3D model is not present for a given footprint and that a simple extrusion of the silkscreen has been used to provide basic 3D capabilities.

If the board has been created in an older version (pre Version 7) of the software you can automatically apply the 3D model data to the board via the 'Update Visual Models' command on the File menu in the 3D Viewer. This action will do the following:

- 1) Apply 3D data from the library parts to the footprints in the ARES Layout.
- 2) Import the layout back into the 3D Viewer.
- 3) Refresh the 3D Viewer to display the new footprint models.

R4 Systems Inc.

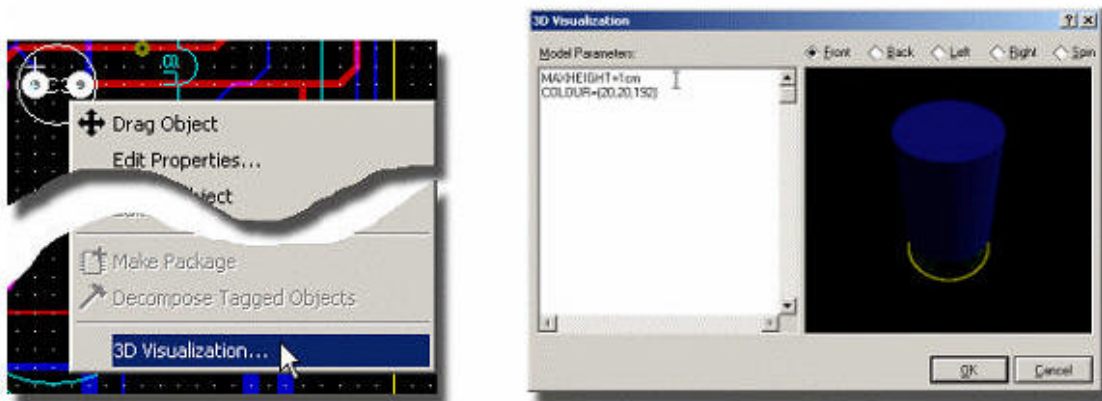


Applying Visual Data from the Libraries to a Legacy Layout

Naturally, this affects only those parts for which 3D models exist in the libraries (all pre-supplied footprints and any custom footprints for which you have created 3D models). Parts which have no 3D data associated with them will still appear in red and you must create a 3D model for them as detailed below.

Creating 3D Models inside ARES

3D information is stored within the package and we must therefore work inside ARES rather than inside the 3D viewer when creating 3D models for custom footprints. Fortunately, ARES comes equipped with a live 3D preview, allowing you to visualize the model and see the effects of changes you make as you go along. To start work on a 3D model for a footprint simply place the part on the layout and then right click on it, selecting 3D Visualisation from the resulting context menu. You should then be presented with a dialogue form consisting of a text box (for the script entry) and a window on the right showing the current, extrusion only, preview of the part.



Invoking the 3D Visualisation Command in ARES.

Note that there are five buttons along the top of the preview window – these allow you to fix the 3D model at a given position and are often useful when scripting. Also note that the silkscreen graphic is shown together with the drill holes, allowing you to accurately specify height clearance and pin dimensions.

R4 Systems Inc.

The idea now is that we type in parameters according to the type of package in order that our 3D footprint more accurately reflects a real world device. Clearly the type of model depends entirely on the dimensions of a physical part so we've summarized the information below.

Body Type EXTRUDED (default)

This method, used in the default case where no TYPE property has been specified, attempts to form a closed polygon from a graphic on a particular layer, and then extrudes that polygon in the Z dimension. Should there be no closed polygon type extruded behaves in the same way as TYPE=NONE (detailed below), although additional processing is done to determine this case. The LAYER property can be used to override which board layer is used to form the basis of the closed polygon, where the following values are available:

SILK MECH1 MECH2 MECH3 MECH4 KEEPOUT OCCUPANCY

The default for the layer property the silk screen layer.

The following standard properties are also available:

COLOUR	R,G,B value for the colour of the body.
MINHEIGHT	Specifies the height of the bottom of the body from the top of the board.
MAXHEIGHT	Specifies the height of the top of the body from the top of the board.
X	Specifies the X-axis distance of the origin of the body to the origin of the board graphic
Y	Specifies the Y-axis distance of the origin of the body to the origin of the board graphic
ANGLE	The angle of the body relative to the packages angle (e.g. ANGLE=45)

Assignment of properties in the script block is made in the normal way:

```
<property>=<value>
```

with the exception of the colour property which is assigned as:

```
COLOUR=(r,g,b)
```



Hints

- COLOUR=(20,20,20)

Body Type “NONE”

Setting this body type removes both the body and the pins, leaving a simple hole in the board.

TYPE=NONE

R4 Systems Inc.

However if you require a JUMPER for example then you can add the parameter PINTYPE=STRAIGHT which will redraw the pins. For more information, please see the subsequent section on Pin Types.

Body Type “CUBOID”

The following details the provided properties for a cuboid body type:

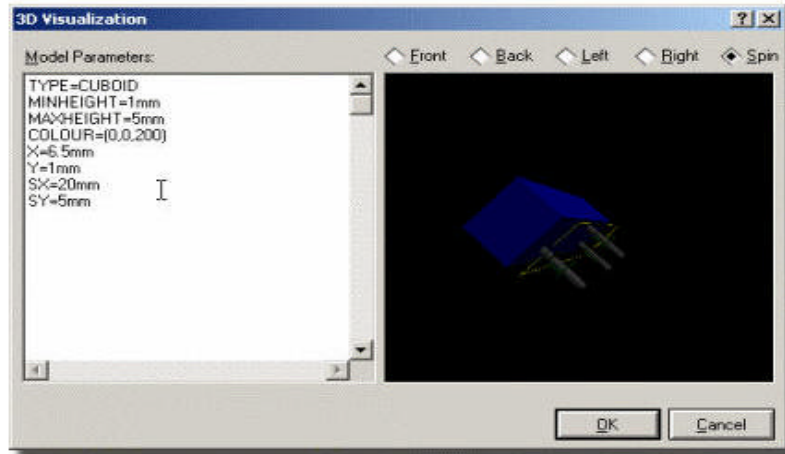
TYPE	Sets the body type (CUBOID in this case).
COLOUR	R,G,B value for the colour of the body.
MINHEIGHT	Specifies the height of the bottom of the body from the top of the board.
MAXHEIGHT	Specifies the height of the top of the body from the top of the board.
X	Specifies the X-axis distance of the origin of the body to the origin of the board graphic
Y	Specifies the Y-axis distance of the origin of the body to the origin of the board graphic
SX	Specifies the size of the width of the cube
SY	Specifies the size of the length of the cube
ANGLE	The angle of the body relative to the packages angle (e.g. ANGLE=45)

Examples of this type include the PRE-HMT, PRE-MT0.75, PRE-MT1.25 and the PRE-HMIN footprints in our libraries. If you select and place one of these parts and then invoke the 3D Visualisation command from the context menu (as above) you should see that the following properties have been added:

```
TYPE=CUBOID  
MINHEIGHT=1mm  
MAXHEIGHT=5mm  
COLOUR=(20,20,20)  
X=6.5mm  
Y=1mm  
SX=20mm  
SY=5mm
```

This duly produces a 3D cuboid as shown in the preview window. Adjusting values will live update the preview window, allowing you to quickly customize to your requirements.

R4 Systems Inc.



The PRE-HMT part with 3D Visualisation Data Applied.

Body Type AXIALCYLINDER

The following details the provided properties for a axial-cylinder body type:

TYPE	Sets the body type (AXIALCYLINDER in this case).
COLOUR	R,G,B value for the colour of the body.
MAXHEIGHT	Specifies the height of the top of the body from the top of the board.
X	Specifies the X-axis distance of the origin of the body to the origin of the board graphic
Y	Specifies the Y-axis distance of the origin of the body to the origin of the board graphic
LENGTH	Sets the length of the cylinder
ANGLE	The angle of the body relative to the packages angle (e.g. ANGLE=45)
RADIUS	Controls the radius of the cylinder about the cylinders axis.
BAND	Sets the position and colour of the resistance bands

The final two parameters are worthy of further discussion as they allow provide considerable depth of control and also enable automatic colour coding of resistor bands for example.

The RADIUS Property

The RADIUS property is of the form:

$$RADIUS=P1,P2,P3,P4$$

Where:

P1	Controls the vertical radius of the cylinder.
P2	Controls the horizontal radius of the cylinder.
P3	Controls the depth of the radius towards the cylinders axis.
P4	Controls the depth of the radius along the cylinders axis

Essentially, the first two parameters allow for the production of elliptical bodies and the second two parameters extend the functionality to allow for the creation of torpedo shapes.

Note that it is not necessary to supply all four parameters if the desired shape does not require them. The default for the first radius parameter is a small nominal value such that the cylinder is

R4 Systems Inc.

visible. The default for the second radius parameter is to take the value of the first radius parameter. The default for the third radius parameter is zero. The default for the fourth radius parameter is to take the same value as the third

The BAND Property:

The BAND property is of the form:

$$BAND_x = P1, P2, (r, g, b)$$

Where x can be a number from 1 to 9 (up to 9 bands permissible) and:

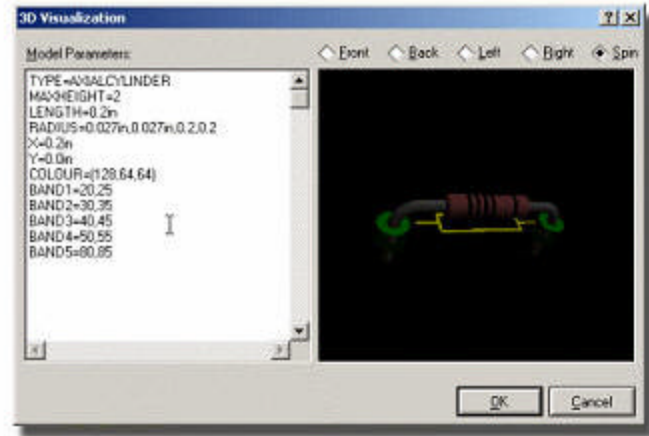
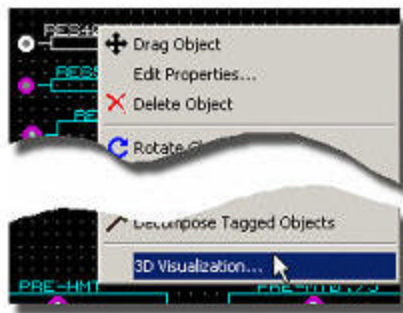
P1	The starting point of the band along the axis of the cylinder (as a percentage of the cylinder size).
P2	The end point of the band along the axis of the cylinder (as a percentage of the cylinder size)
RGB	The colour value of the band. If omitted, the default colour will be calculated from the components value field, allowing for automatic colour coding of resistors etc.

The vast majority of resistors in our libraries are created in this way. If, for example, you pick a RES40 from the libraries and invoke the 3D Visualisation command from the right click context menu you should see that the following properties have been added:

```
TYPE=AXIALCYLINDER
MAXHEIGHT=2
LENGTH=0.2in
RADIUS=0.027in,0.027in,0.2mm,0.2mm
X=0.2in
Y=0.0in
COLOUR=(128,64,64)
BAND1=20,25
BAND2=30,35
BAND3=40,45
BAND4=50,55
BAND5=80,85
```

In particular, note that no colour parameter has been specified for the bands as we want the 3D viewer to calculate the colour codes for us depending on the value of the resistor. What we are saying is that BAND1 starts 20% along the cylinder and stops 25% of the way along, BAND2 starts 30% of the way along and stops at 35% of the way along, etc. etc.

R4 Systems Inc.



The RES40 package with 3D visualization data applied

The calculation of the colour band values only takes place when the 3D Viewer is invoked on a board. Feel free to try this, first editing the RES40 and setting a value and then invoking the 3D Viewer from the Output Menu.

Pin Types

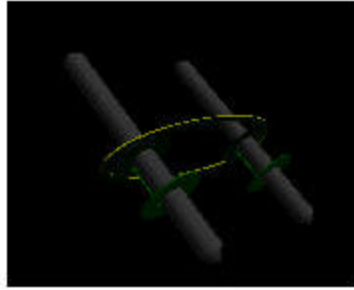
Having customized the body type it is sometimes also necessary to change the pin type for the 3D model.

There are seven styles of pins available for use as follows:

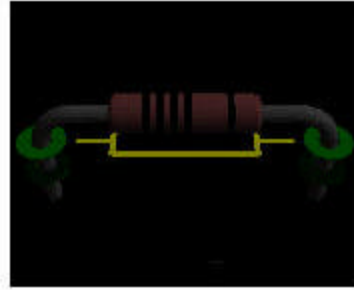
1)NONE 2)STRAIGHT 3)BENTWIRE 3)SMTJ 4)SMTZ 5)SMTB 6)CUBOID

R4 Systems Inc.

Straight



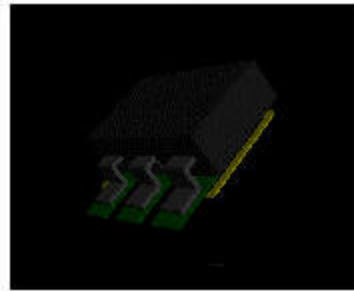
Bentwire



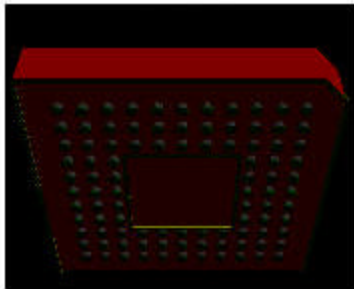
SMTJ



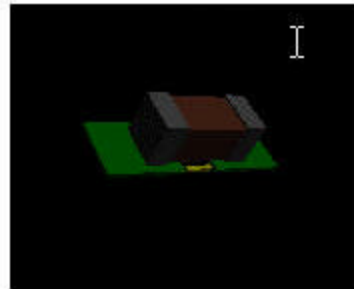
SMTZ



SMTB



CUBOID



The default pin type in a given situation is dependant on the pad style in ARES and also on whether the pad is below the footprints 3D body.
For through hole pads, the default pin style is BENTWIRE unless the pin is under the package body or there is no package to attach to, in which case the default becomes STRAIGHT. For surface mount pads, the pin style is SMTZ unless the pad is below the package body or there is no package to attach to, in which case the default becomes SMTJ.

These defaults tend to work well but if need be you can assign a specific pin type in the normal way. i.e. PINTYPE=xxxxxx. Additional properties that can be applied to pins are detailed below:

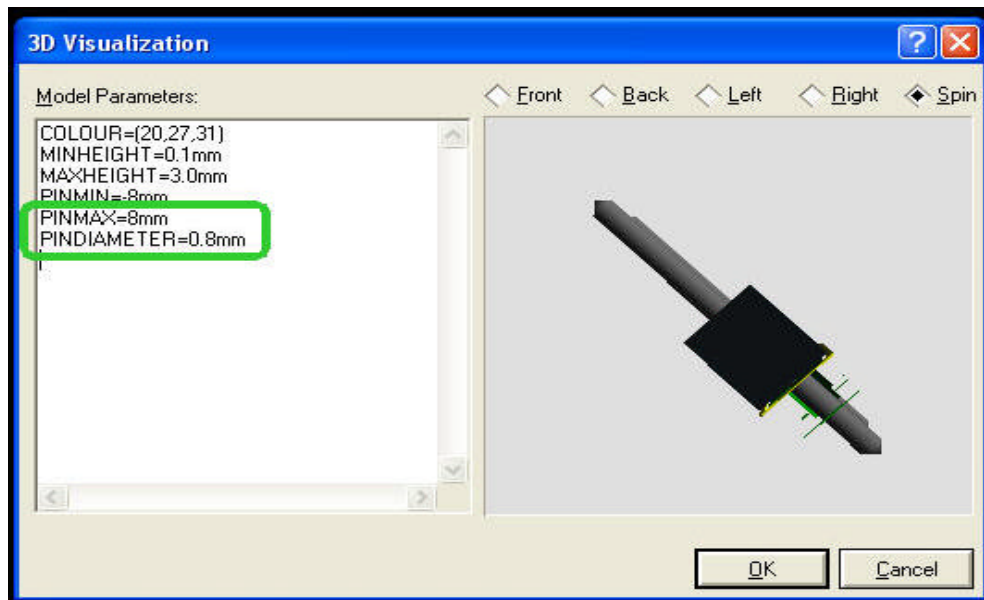
PINCOLOUR	R,G,B value for the colour of the pin.
PINMAX	This indicates the highest extent of the pin from the surface of the board and is used by all pin types.
PINMIN	This indicates the lower extent of the pin from the surface of the board. For through pins this will likely be a negative value, for surface mount it is likely to be zero.

R4 Systems Inc.

PINDIAMETER	For STRAIGHT and BENTWIRE pins, this controls the diameter of the pin. For SMTB it controls the diameter of the hemisphere.
PINLENGTH	For SMTJ and SMTZ this controls the length of the base of the pin.
PINTHICKNESS	For SMTJ and SMTZ this controls the thickness of the pin.
PINWIDTH	For SMTJ and SMTZ this controls how width of the pin.
PINMOVE	This moves the pins closer or further away from the boundary of the package body by the fixed amount specified.

you should see that the following properties have been added:

COLOUR=(20,27,31)
MINHEIGHT=0.1mm
MAXHEIGHT=3.0mm
PINMIN=-8mm
PINMAX=8mm
PINDIAMETER=0.8mm

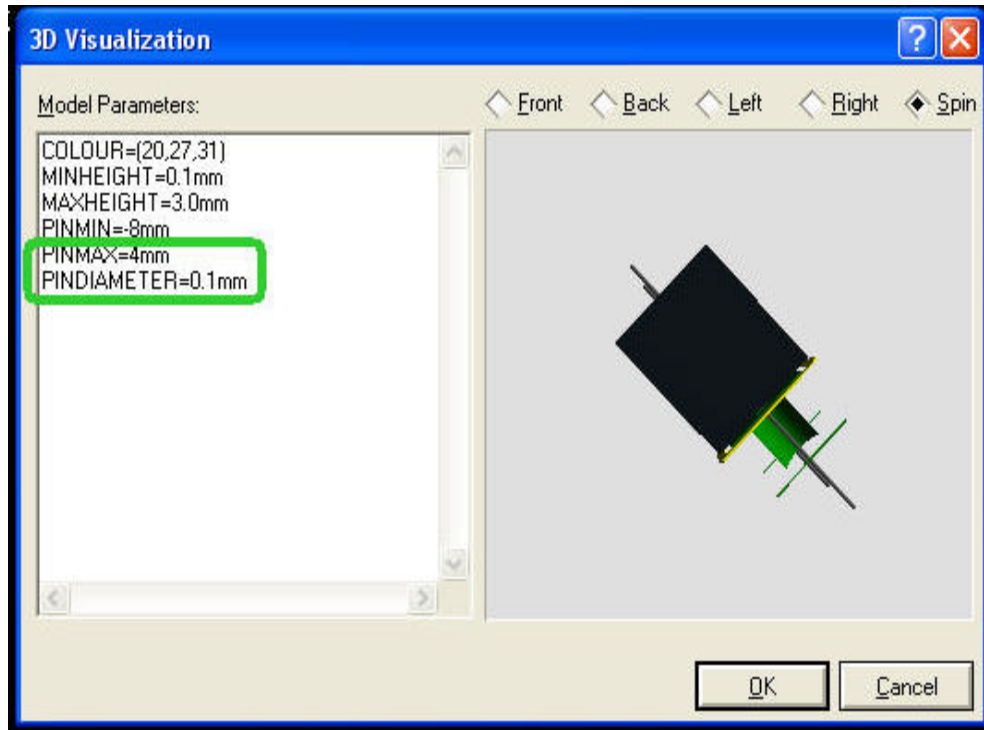


By changing

PINMAX=4mm
PINDIAMETER=0.1mm

See how it effects on pin size and diameter

R4 Systems Inc.



Importing 3D Models from another Package

Models can be imported into the 3D Viewer via the following simple procedure:

To use a 3D model created in another package:

- 1) Export the model from your package as a single object in the 3DS file format.
- 2) Place the 3DS file in the library directory of your Proteus installation.
- 3) Place the footprint in ARES and select the 3D Visualisation option from the resulting context menu.
- 4) The following properties must then be added:

```
TYPE=MODEL  
FILE=<filename>.3ds  
NAME=<objectname>  
SCALE=<scale><units>
```

The 3D Viewer operates in terms of numbers and not in terms of physical dimensions. You must therefore specify what size a single unit is. This is typically 1mm, 1in, 1cm etc.

Finally, it is possible that the model may require an X or Y offset to position it correctly. You can therefore adjust the position via the following properties:

```
X= <x-offset><units>  
Y= <y-offset><units>
```

R4 Systems Inc.

For example, a DIP40 that was created in a 3D package might have the following script entered in the 3D Visualisation dialogue form:

```
TYPE=MODEL  
NAME=DIP40_600  
FILE=DIP600.3DS  
SCALE=1in
```

Guidelines for Model Creation in another Package

The key to any good 3D model is sufficient detail that it looks correct from the distance the user is likely to use, while at the same time keeping the number of polygons that make up the 3D model low enough that the display generation and update will be reasonably fast.

A classic example of where the above two criteria conflict is in a cylinder: the number of facets that make up the approximation of a cylinder has to be such that the stepping is not obvious, whilst at the same time too many facets will quickly produce a high polygon count that could quickly swamp the rendering pipeline's processing resources, this problem can be further compounded when the 3D model in question is merely a small radial capacitor that no one would even notice missing or is perhaps obscured by other components.

Hints

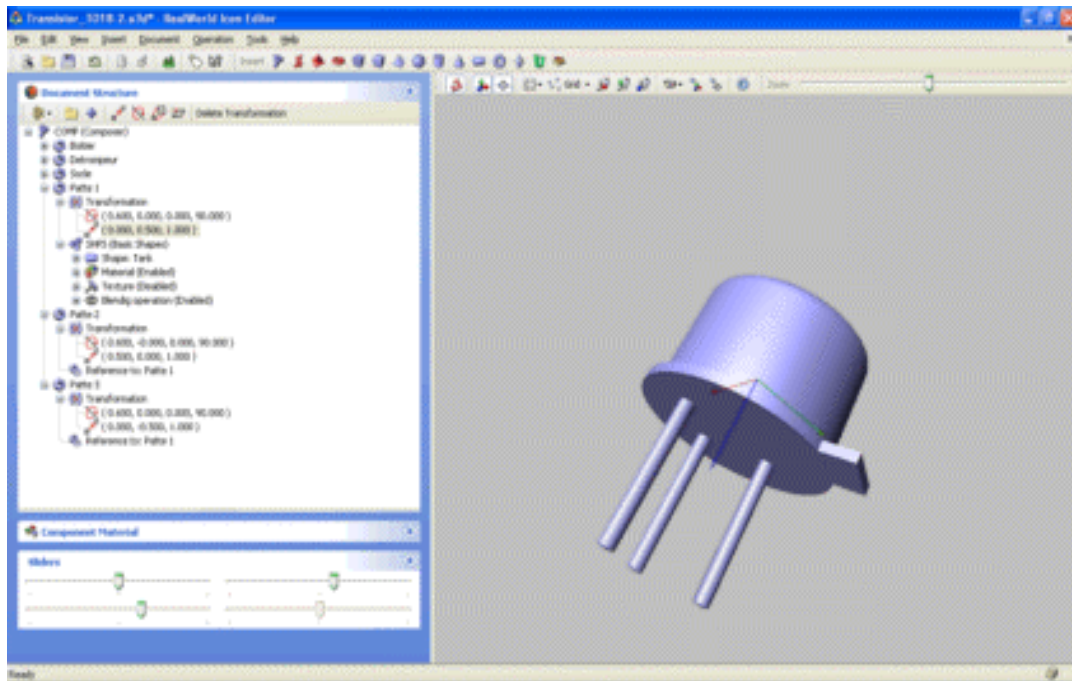
- Give the model sufficient detail that the object is representative of the component and that the user would be able to identify the approximate type of component, though not necessarily the specific component.
- Use as few polygons as you can get away with: if your model has twice as many polygons as is required your model will be twice as slow to draw, this is not normally a problem for one component but if you have PCB with 50 of them on you may get problems.
- One of the areas where excessive polygons occur is on smoothed non planar shapes: curves. The trick here is, where possible, to use smoothing, this causes a face to look curved without actually having to be curved.
- Keep an eye on the number of vertices and faces in a model, if they are hitting the thousands and it is not a particularly complex part, you are likely making the model excessively detailed or wasting polygons.
- Do not be afraid of hand editing the results that are produced by the 3D editing package's own algorithms, often they do not produce optimal results.
- The import routine assumes a model is a single object in the *.3DS file, it is quite possible that you will have to merge multiple objects to create a final object. You will also have to name that object appropriately.
- The file extension (.3ds) must be included in the FILE property



R4 Systems Inc.

Example:

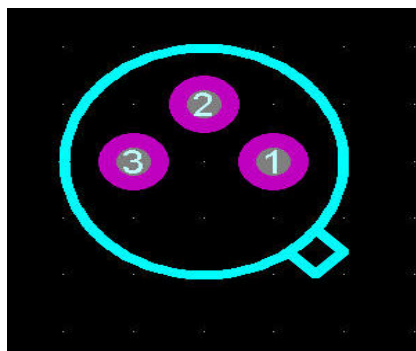
In the following Example, We want to add new external 3D View to ARES footprint, We will try to put an example components created for use in Proteus software : Isis for schematic drawing, Ares for PCB design. There are different CAD softwares for drawing 3D Components, Select one of them. Our sample is a Transistor TO18 package.



After drawing your 3D component in the 3D Software, 3DS files have to be placed in the "Library" directory of Proteus, for example :

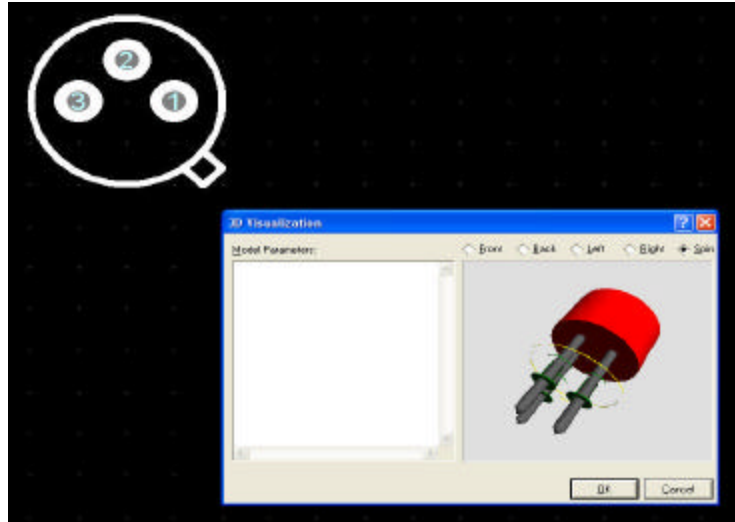
<C:\Program Files\Labcenter Electronics\Proteus 7 Professional\LIBRARY>

Run ARES and make a footprint like following picture:



By default, Ares use its internal 3D engine to do basic objects, or sometime use external files.

R4 Systems Inc.

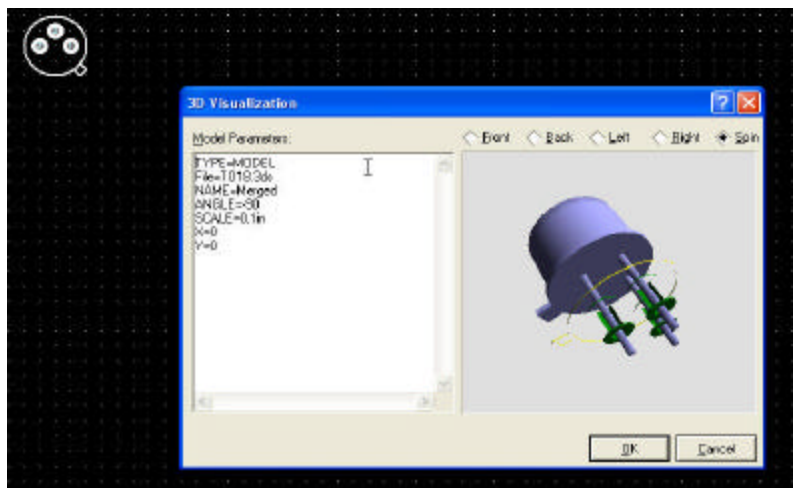


So if you don't specify any parameters here , ARES will show the the abovementioned picture by default.

To use an external 3D component, you must specify the component type which is contained an external file. This can be done in the 3D visualization window of Ares by attributing MODEL value to the TYPE parameter. For exemple, to use external TO18.3ds for a 2N2907 transistor, you have to place the component on the PCB, right click on the 2D footprint, and in the context menu, choose the 3D Visualization command. In the 3D Visualization window that come, write the following text.

```
TYPE=MODEL  
File=TO18.3ds  
NAME=Merged  
ANGLE=-90  
SCALE=0.25  
X=0  
Y=0
```

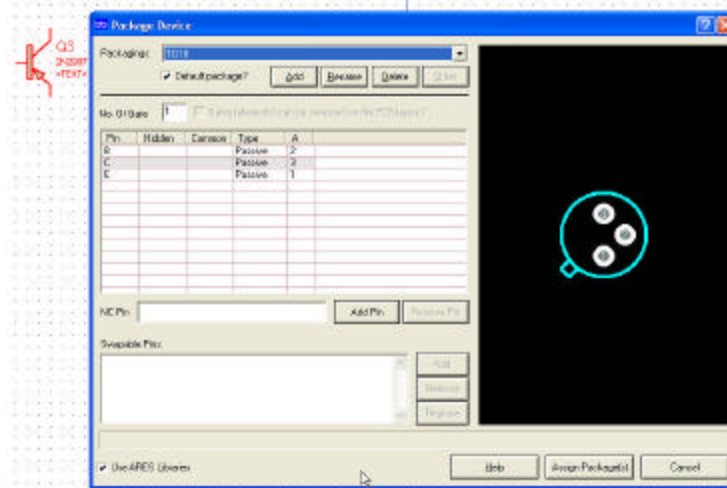
3D component will change interactively, and you can see the result on the view menu. See the following picture.



R4 Systems Inc.

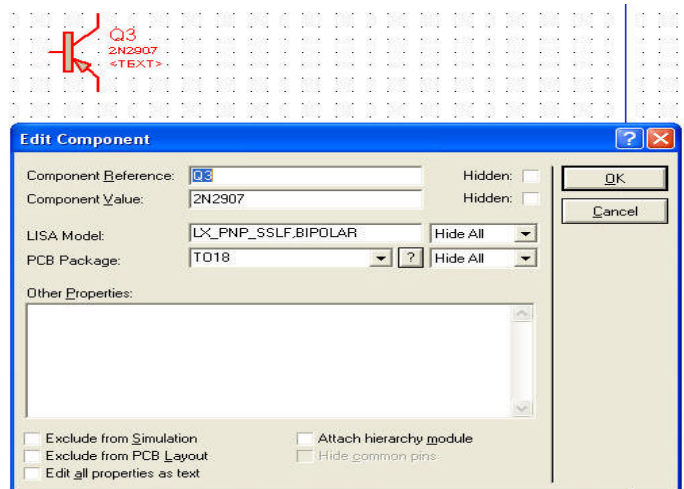
Click OK and make a package as it was discussed in Making Package section.

Now, open ISIS, and pick a Transistor, like 2N2907, and add new PCB Package, one way could be right click on component and select Packaging Tool from the context menu, see following picture.



Assign package to the component and save it into the library

Now, new package has been added to the component. See next picture.



Hints

- We used RealWorld Icon Editor (RWIE) as a 3D Designer, The official website is www.rw-designer.com.
- For Export 3DS from RWIE Software, ask them for 3DS export function.
- 3DS files have to be placed in the "Library" directory of Proteus.

