OrCAD Tutorial:
Additional Notes

Creating a Library from
A Schematic

February 6th, 2006

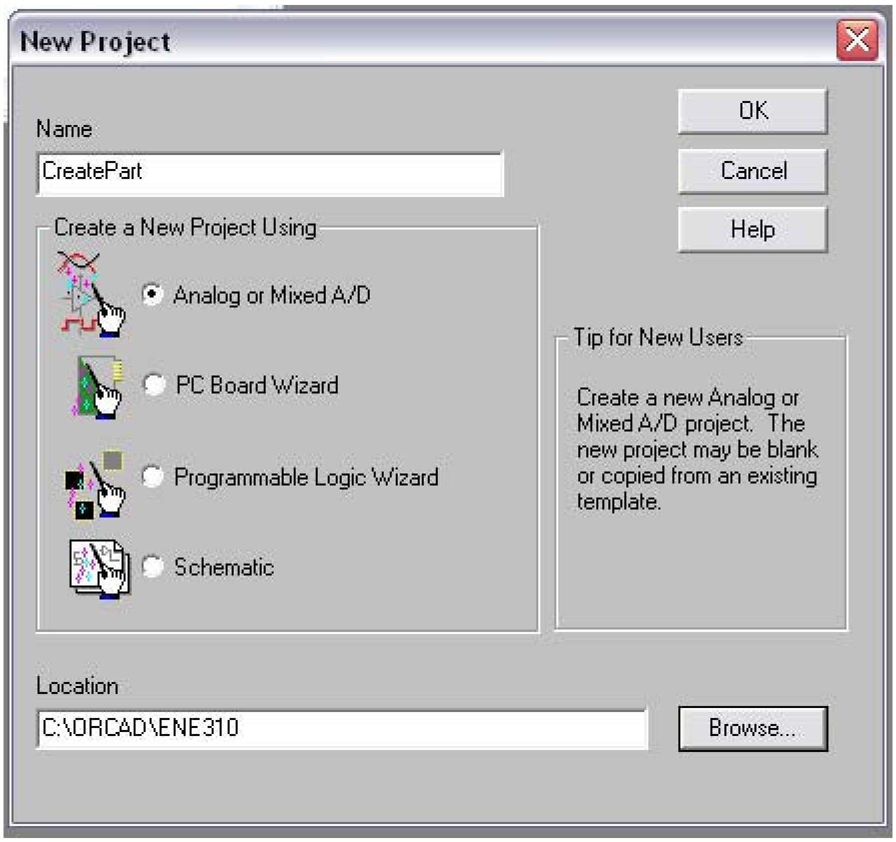
Jannes Venter

OrCAD: Creating a library part of a schematic

This tutorial will briefly guide you towards creating a custom library with a previously created schematic as a component in the new library. The aim is to create a file with a **.olb** extension which contains the schematic as a placeable part in the OrCAD schematic environment, with PSPICE functionality included to describe the circuit.

Creating a new project

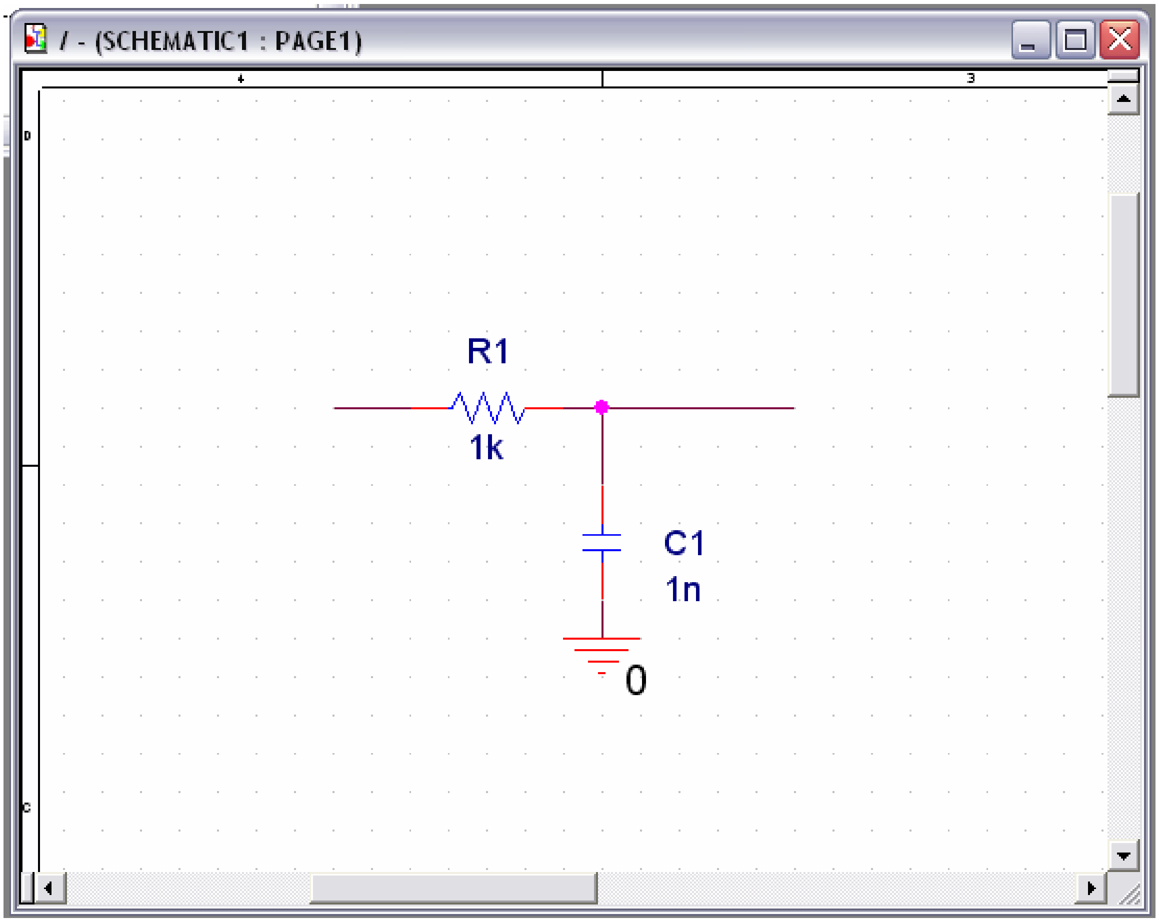
A new project needs to be created to create a infrastructure for the desired schematic to be created. Therefore, as an example, create a new project (*File>>New Project…*) called **CreatePart**.



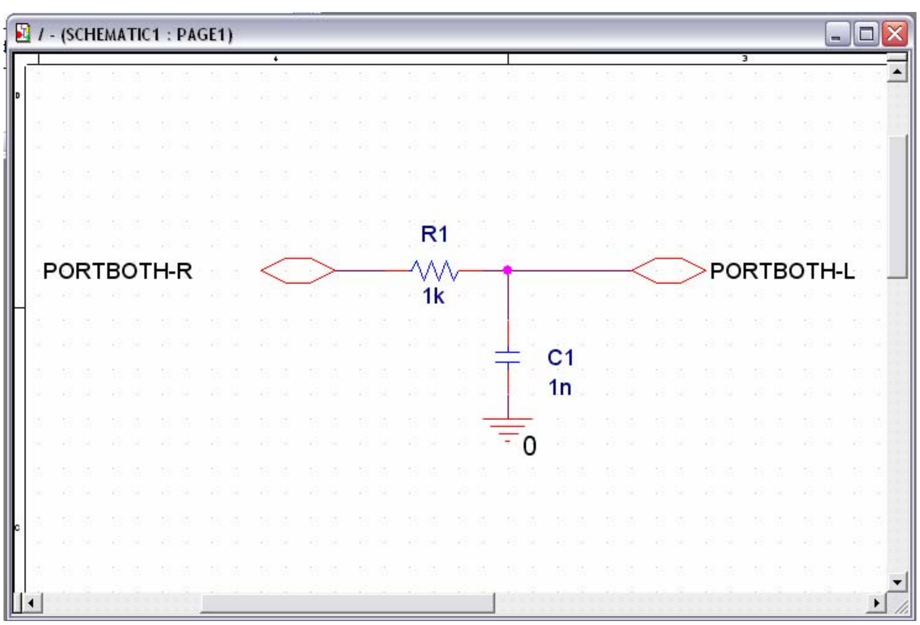
and select the option to base the design on a blank hierarchy.

Creating the schematic – RC network

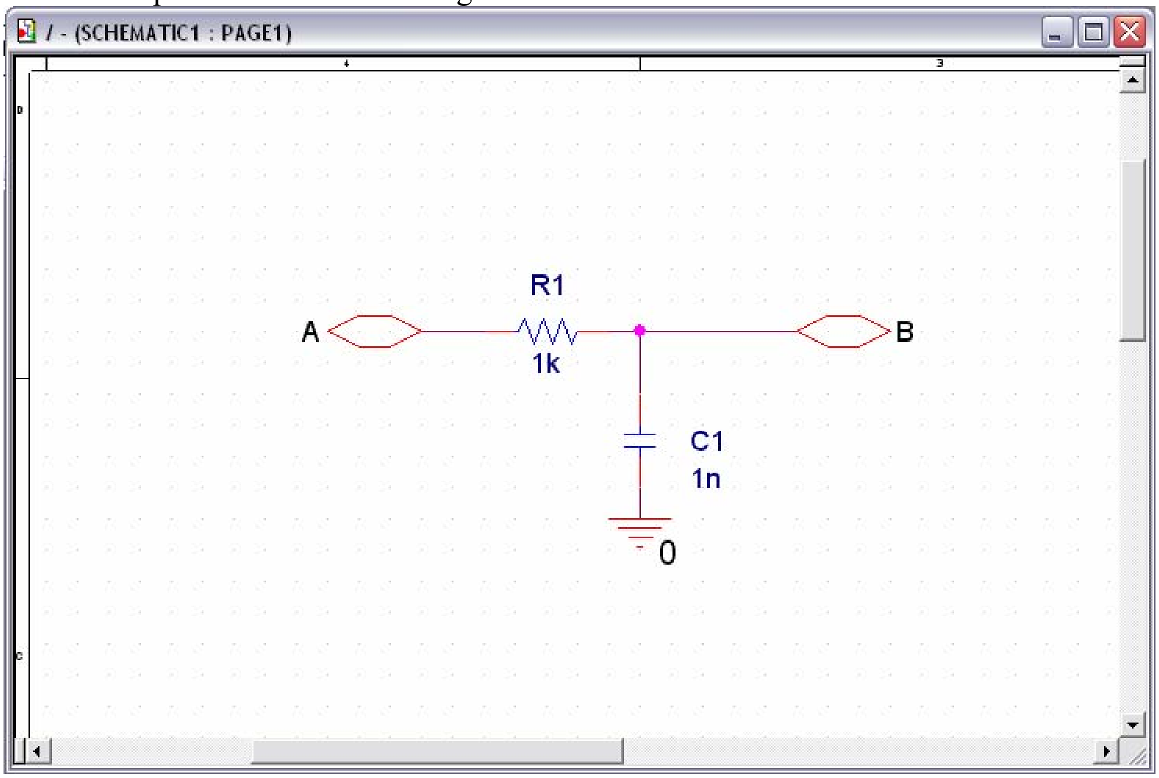
For the purpose of the tutorial, an RC-network will be used to illustrate the necessary methods. In the schematic editor which is now open (if the prior steps were followed), create an RC-network with R = 1 kΩ and C = 1 nF. Use the **analog.olb** library to obtain the necessary components.



To add connectivity to the circuit when using it as a single “black box” component, you need to define *ports*. To do this, select the *Place port* tool from the drawing toolbar and add hierarchical ports to the desired points on the schematic.



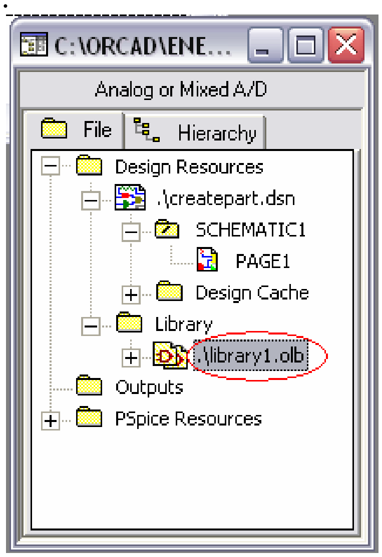
You need now to rename the ports to some specific name which will be later used to "hook up" the schematic to a symbol when creating a part from the schematic. Therefore, rename the port names as in the diagram below.



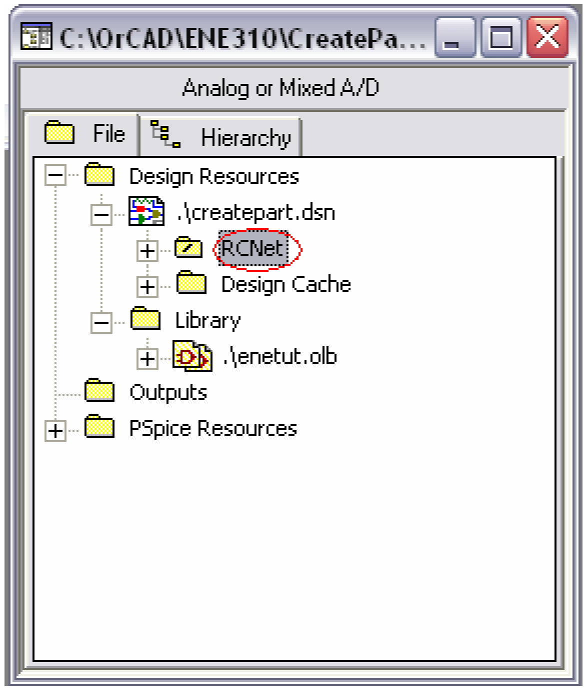
Now save the schematic and close the schematic window. Next, make sure you are looking at the project manager window.

Creating and naming a library

Go to *File>>New>>Library* and select the option. The following change will occur in your project manager window:

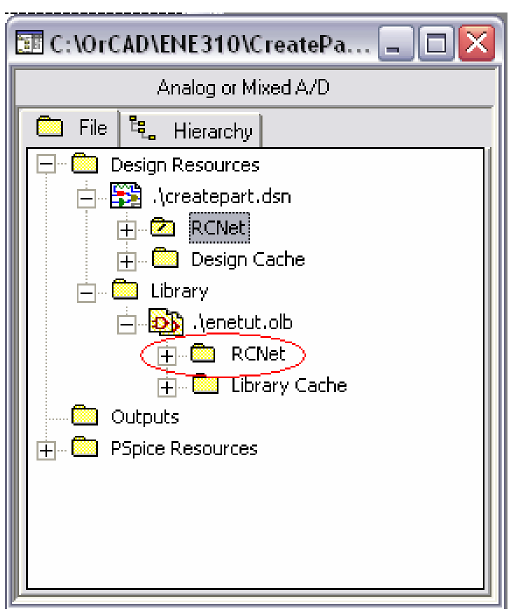


Rename the library as **ENETut.olb** by right-clicking on it and selecting the *save as…* option. Rename the schematic as well:



Including the schematic into the library and creating a part

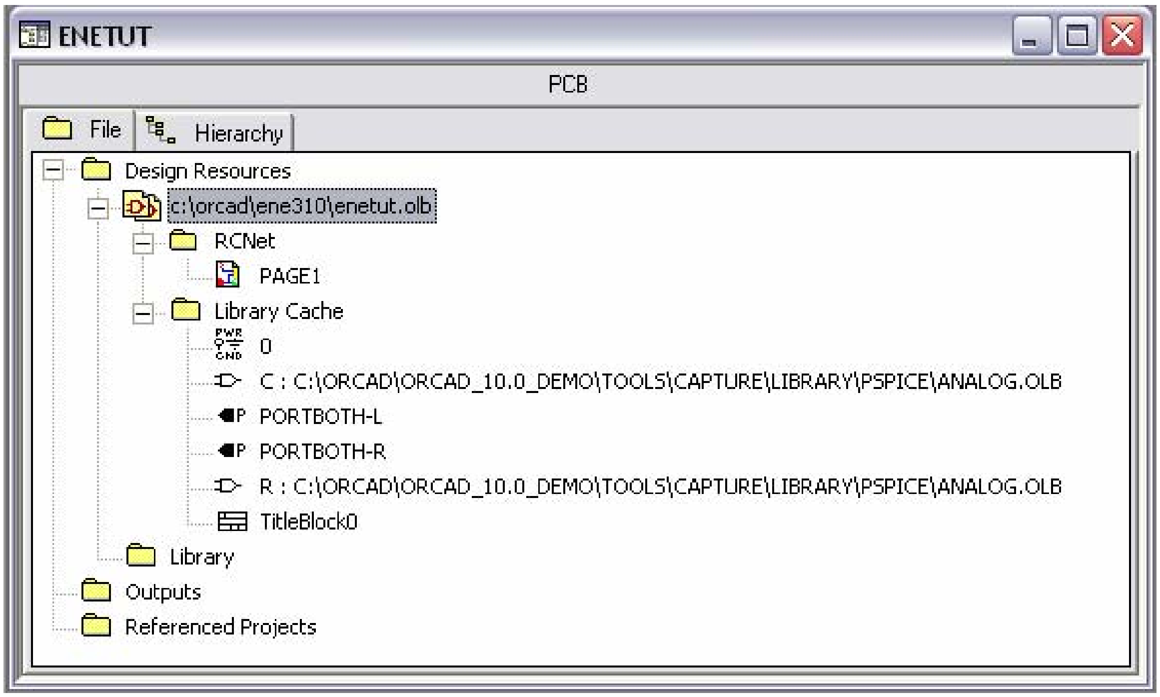
The next step is to add the created schematic to the library so that the information is available in the library itself. To do this, select the **RCNet** schematic folder and drag it across to the library **enetut.olb**. A notice will appear which states that a copy will be made of the schematic folder in both the library and the project root folder. After completing the operation, the project manager should look like this:



Now save the library (right click option). Notice the path where the **.olb** file is stored. Close the project completely and select *Open>>Library…* and select your **enetut.olb**

Creating a part

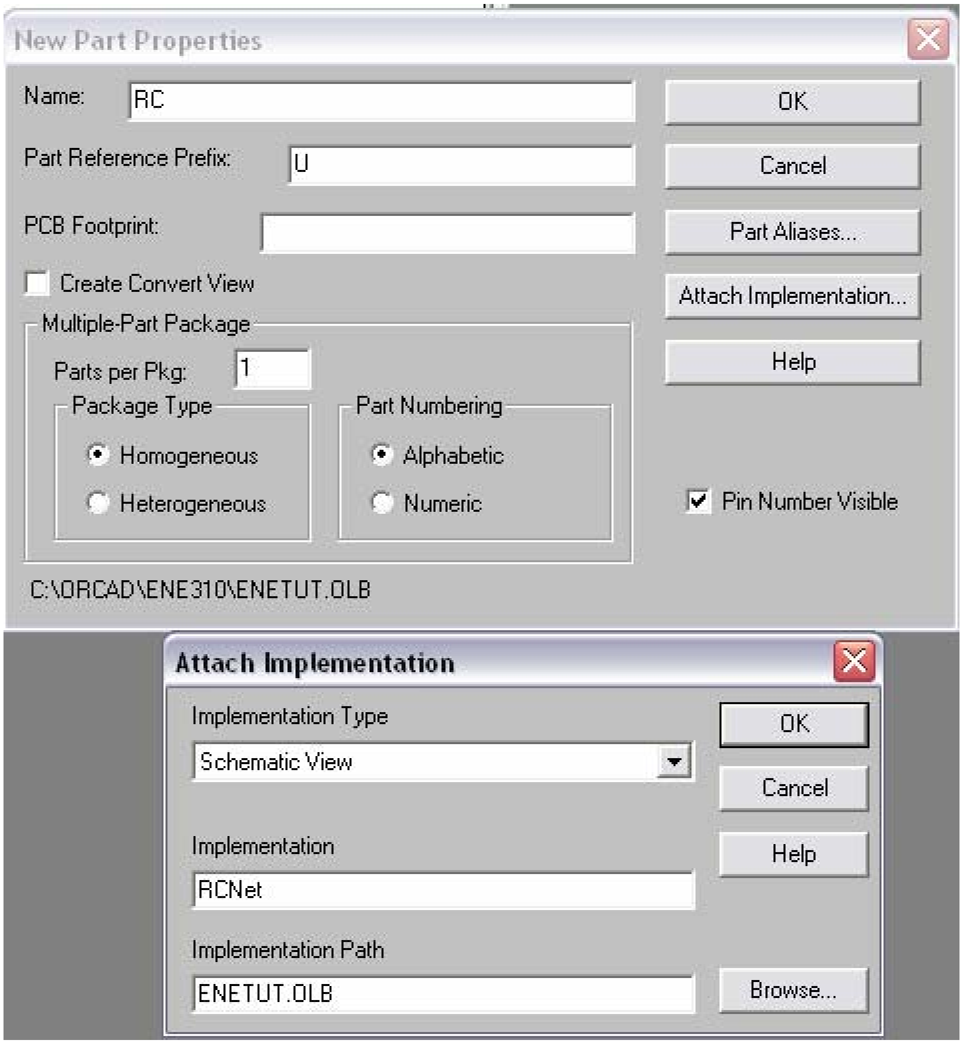
If all went well, your new project manager for the library file should look like this:



Notice that all the design information (eg. ports, the capacitor and resistor etc.) are referenced in the library cache. Also notice the presence of your schematic folder **RCNet**.

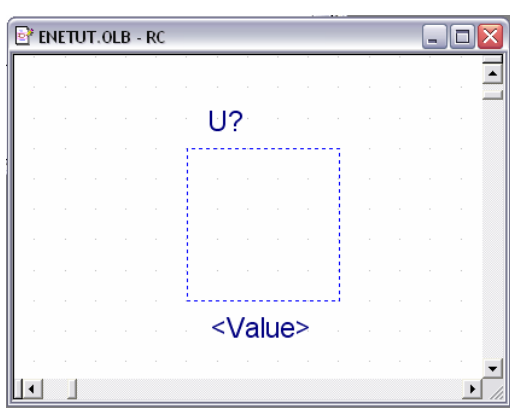
Now right-click on the **c:\orcad\ene310\enetut.olb** entity and select *New Part.*
In the *Name* space, fill in the name for the new part, eg. RC. The next step is to assign an implementation for the part, which is basically the method to describe the electrical behaviour of the part. Click on *Attach Implementation*, select **Schematic View** in *Implementation Type*, since we are interested in hooking up a schematic to a part. In the *Implementation* box, type in the name of the schematic folder which is to be used, namely **RCNet** in this case. In the *Implementation Path,* one need to define only the library name itself, since the schematic is included in the library (remember the dragging…).

The settings are shown in the snapshot.

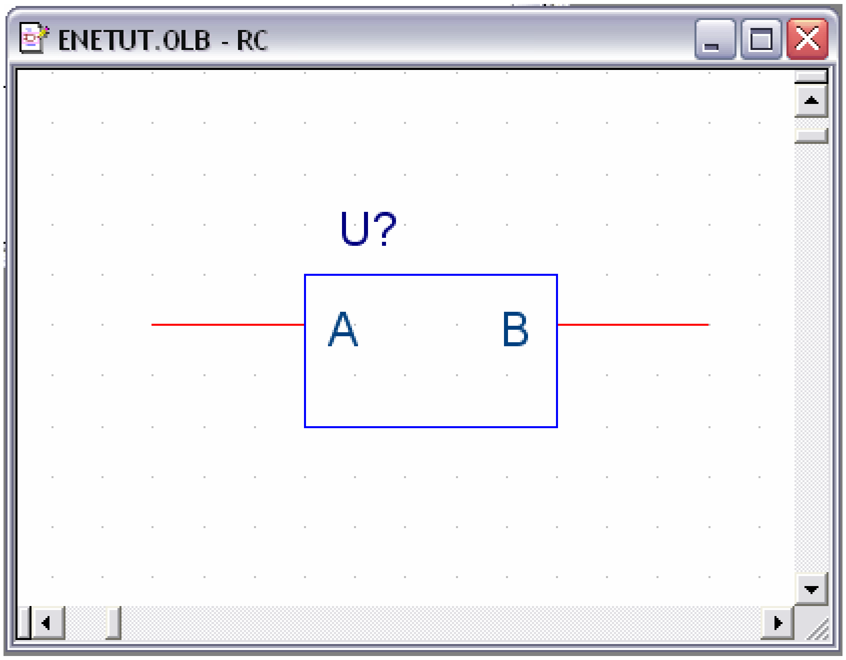


Editing the part symbol

Now, in creating the part, we need to define a symbol which will be used in subsequent schematic compilations. The following screen is automatically presented upon completion of the previous steps.

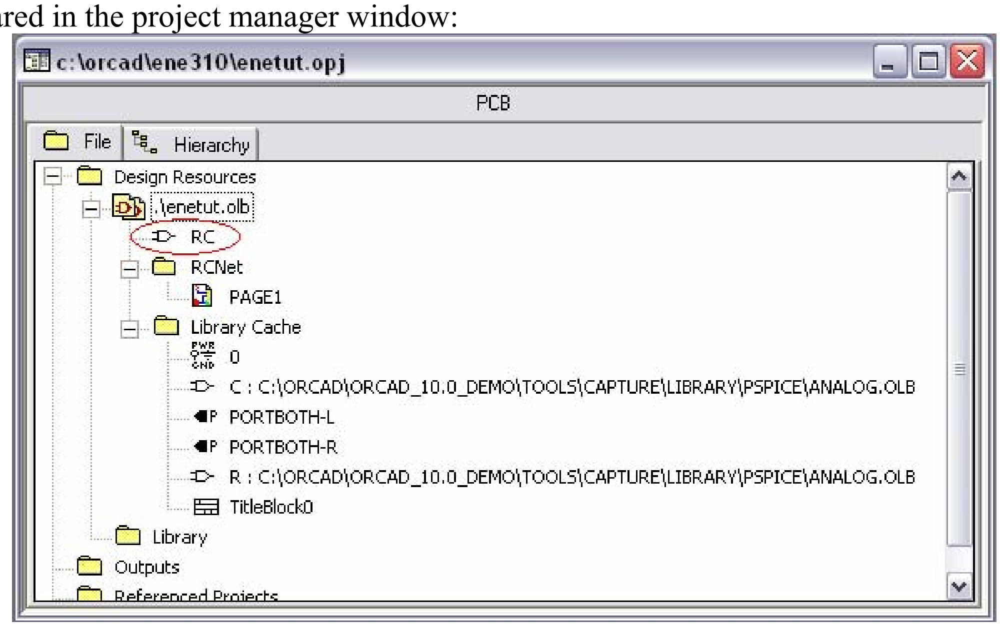


The created schematic only has two ports, named **A** and **B**. Therefore, draw a symbol to be used and define the ports by placing pins from the toolbar and naming them TO CORRESPOND TO THE SCHEMATIC PORTS.



Notice that the **<Value>** text has been deleted, since we are not using it. Next, select from the menus, *Options>>Part Properties…* Add a new property called **Primitive** and set the value as **NO**. This tells the simulator to look "underneath" the component, where it will find the schematic implementation which describes its PSPICE behaviour.

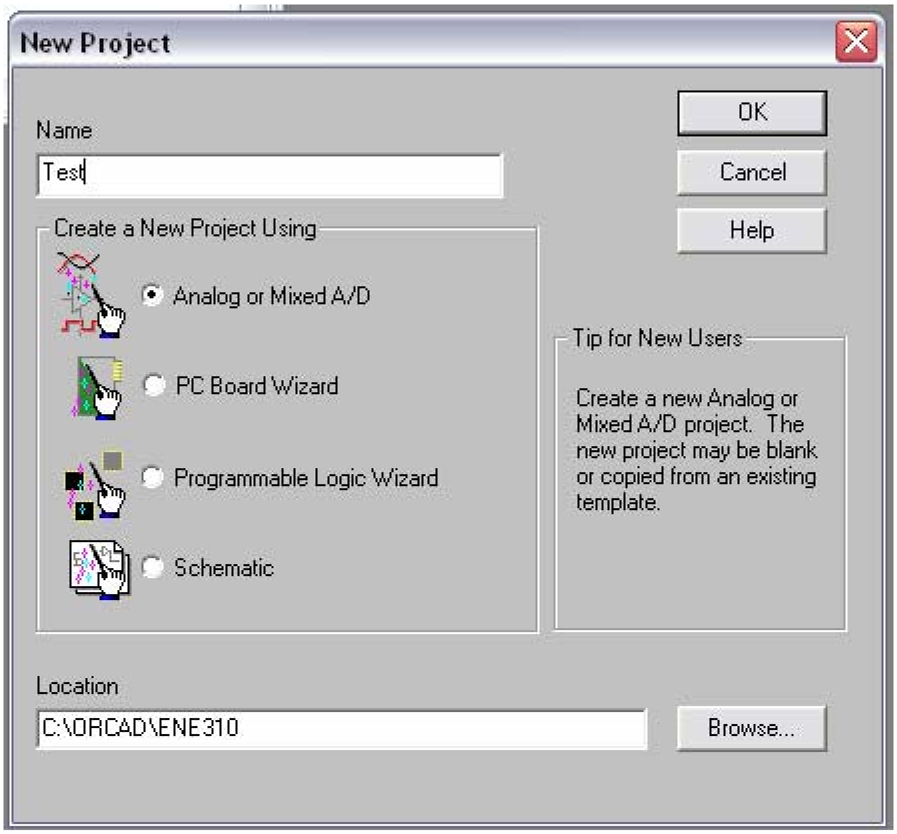
After this operation, select *File>>Save* to save the complete library. A new part has now appeared in the project manager window:



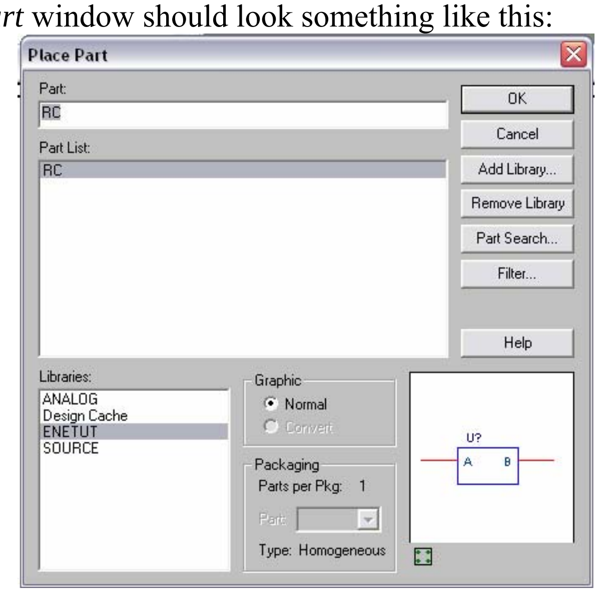
This indicates the presence of a useable component. The next step would be to test the inclusion of the library in a totally independent project.

Verifying the library functionality

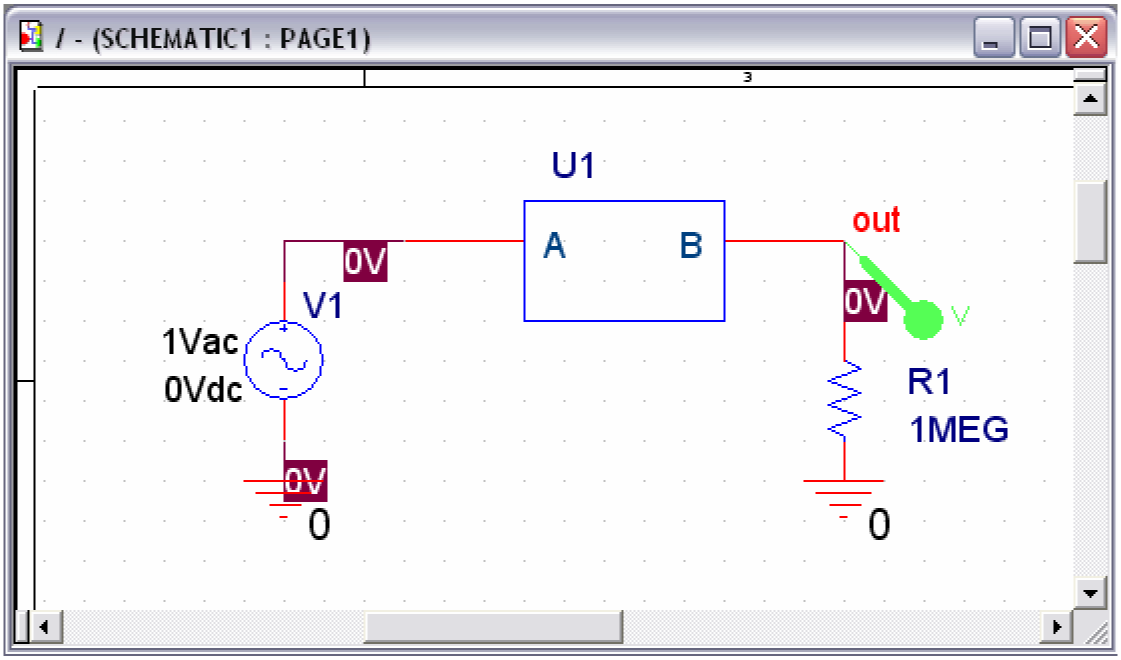
Close everything and start a new project as shown in a previous section.

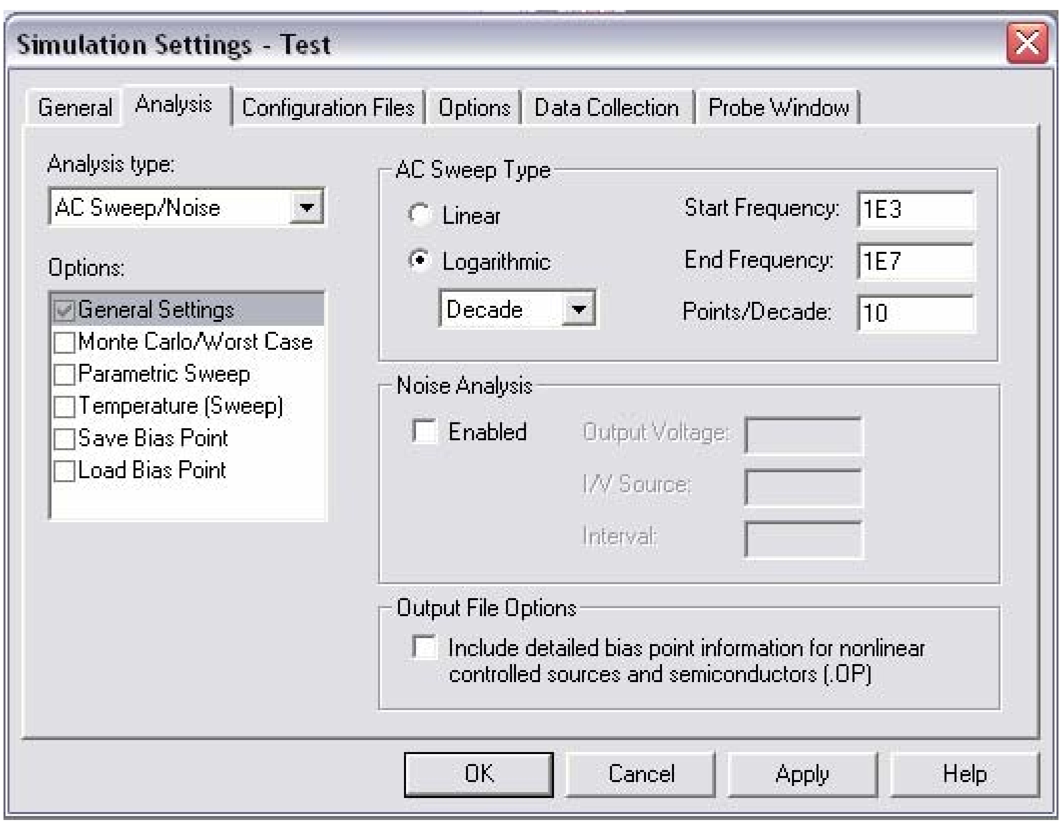


When placing parts, include libraries **analog.olb**, **source.olb** and **enetut.olb**, the created library. The *Place part* window should look something like this:

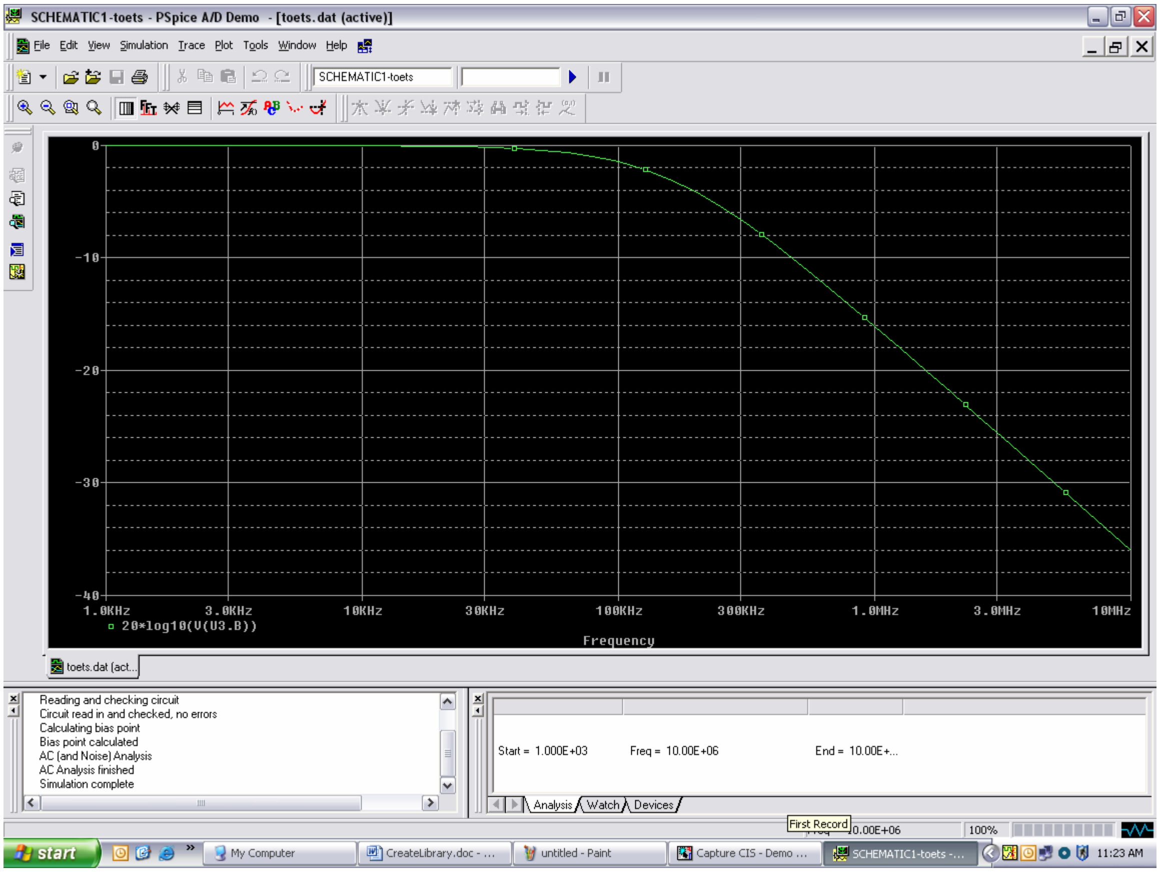


Construct a test circuit for the component. A good choice would be to construct an AC-analysis. The time constant of the circuit (using R = 1 kΩ and C = 1 nF) dictates a -3 dB frequency of about 159 kHz. Therefore, make sure an AC analysis is set up to have this point clearly visible.



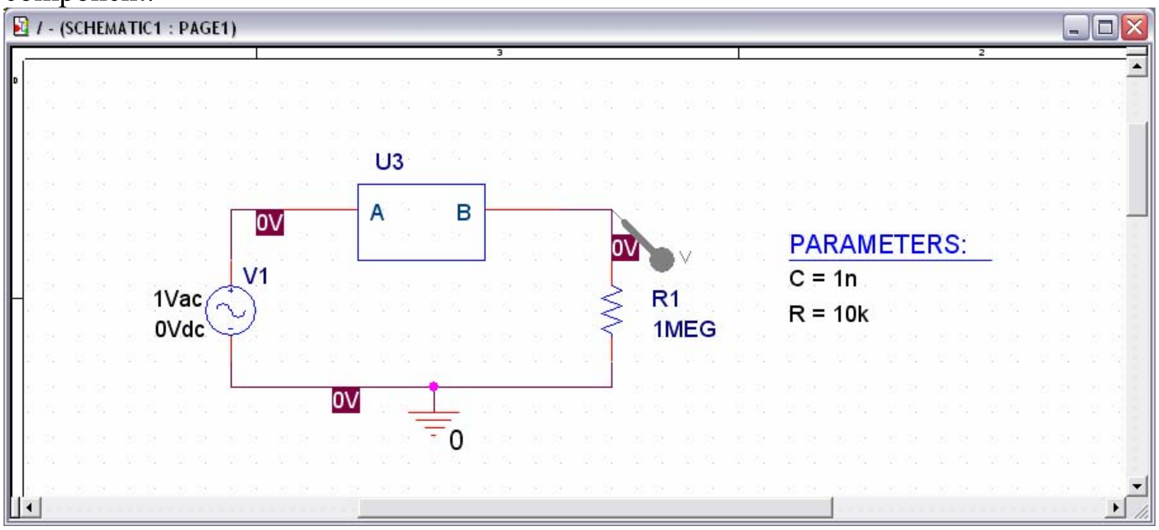


Results


SCHEMATIC1-toets - PSpice A/D Demo - [toets.dat (active)]
File Edit View Simulation Trace Plot Tools Window Help
SCHEMATIC1-toets
0
-10
-20
-30
-40
1.0KHz
3.0KHz
10KHz
30KHz
100KHz
300KHz
1.0MHz
3.0MHz
10MHz
20*log10(V(U3.B))
Frequency
toets.dat (act...
Reading and checking circuit
Circuit read in and checked, no errors
Calculating bias point
Bias point calculated
AC (and Noise) Analysis
AC Analysis finished
Simulation complete
Start = 1.000E+03
Freq = 10.00E+06
End = 10.00E+...
Analysis
Watch
Devices
First Record
100%
start
My Computer
CreateLibrary.doc - ...
untitled - Paint
Capture CIS - Demo ...
SCHEMATIC1-toets -...
11:23 AM

Additional: Parametric Control

If you want to externally control the parameters inside the component, one can define the resistor and capacitor values as **R={R}** and **C={C}** respectively when creating the part schematic. When instancing the part in another schematic, the **PARAM** part from the **special.olb** library can then be used to control the R and C values on the inside of the component:



Obviously, the flaw with this would be the limit of ONE component per schematic page.