

Scripting How-To: How do I Import Multiple Layer Process Definition (LPF) Files into my Project

In some projects it may be useful to have more than one Layer Process Definition file. For example, suppose you are working on a MMIC chip in a GaAs process that it going to be added into a module and bonded to some boards made from Alumina. You might one process definition file for the chip design and one for the module board design. The AWR Design Environment supports using multiple LPF files in a single project so schematics and layouts within the project can have different process definitions yet still be combined in performance simulations. Let's look at an example of bringing more than one LPF file into the project. Consider the following code:

```
Sub Main

    ' Import first lpf (overwrite = true to replace default)
    Set procDef = Project.ProcessDefinitions.Import("C:\mmic.lpf", True)

    ' Import second lpf (overwrite = false to add and not overwrite default)
    Set procDef = Project.ProcessDefinitions.Import("C:\mic_metric.lpf", False)

End Sub
```

The first statement:

```
' Import first lpf (overwrite = true to replace default)
Set procDef = Project.ProcessDefinitions.Import("C:\mmic.lpf", True)
```

Is the same as we saw in the previous FAQ example for importing an LPF file. It imports the file mmic.lpf and replaces the default LPF file in the project. Next the second line imports a different LPF file:

```
' Import second lpf (overwrite = false to add and not overwrite default)
Set procDef = Project.ProcessDefinitions.Import("C:\mic_metric.lpf", False)
```

But this time the second argument OverwriteDefault is set to false so this one will not replace mmic.lpf but be merged into the project with it. Figure 1 shows the project before and after running this script:

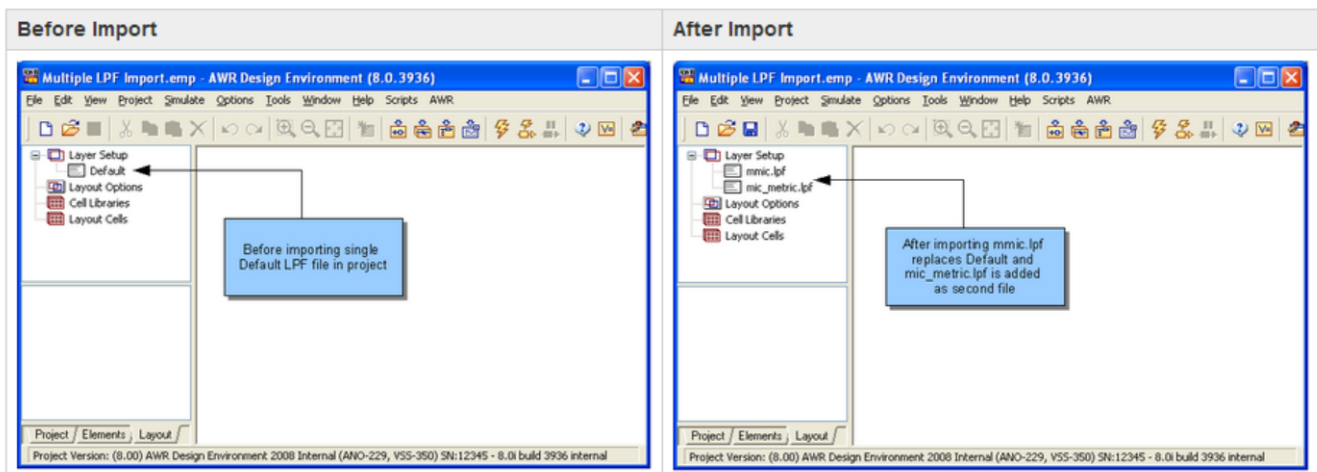


Figure 1 Layout project before and after running the import multiple LPF script.

The key to adding a LPF file is to provide the value False for the OverwriteDefault argument to the Import() method. The first LPF file in the project is considered the default, so anytime this argument is True that LPF file will be replaced.

Now that we have a second LPF file lets look at adding a schematic and setting it to use this second file. Setting it to use the second LPF file involves two steps:

1. Change the schematic so it does not use the project default options. Remember mmic.lpf is the default project LPF file but we want to use the second one mic_metric.lpf.
2. Change the value of the ProcessDefinition property on the schematic to the name of the LPF file we want to use.

Consider the following code:

```
' Code Module
Sub Main
    Dim schem As Schematic

    Set schem = Project.Schematics.Add("MySchematic")
    schem.UseProjectOptions = False
    schem.ProcessDefinition = "mic_metric.lpf"

End Sub
```

The first part is pretty simple. We add a schematic to the schematics collection named "MySchematic" with:

```
Set schem = Project.Schematics.Add("MySchematic")
```

Se also set the variable schem as a reference to the newly created schematic. Next we set the UseProjectOptions property to false so we can overwrite the LPF file from the project default:

```
schem.UseProjectOptions = False
```

Finally we set the process definition file to the mic_metric.lpf file:

```
schem.ProcessDefinition = "mic_metric.lpf"
```

Now when we open the layout associated MySchematic schematic and click on the Layout tab in the project pane we'll see the mic_metric.lpf layers instead of the mmic.lpf layers.