Agilent Technologies

MQA

Passive Subcircuit Model QA

Application Note

Copyright Notice and Proprietary Information

Right to Copy Documentation

Disclaimer

Registered Trademarks (®)

Passive Subcircuit Model QA

Application Note

This application note describes how to run passive subcircuit (subckt) model quality assurance (QA) in Model Quality Assurance (MQA). **Note**: This document was originally released for MQA V2010.2.0.1 in May 2011.

Introduction

MQA can realize comprehensive QA on various models, including the passive subckt model. In this document, an RF resistor is used as the example to show the steps required to run the QA. For more information go to www.agilent.com/find/eesof or contact your local Agilent office. The complete list is available at: www.agilent.com/find/contactus.

Steps to QA

The user can follow the following steps, one by one, to complete a full QA process for the passive subckt model.

Create New Project

Choose *Project --> Create and Run Project* from the main menu. In the popup window click *New* to create a new project (Figure 1).

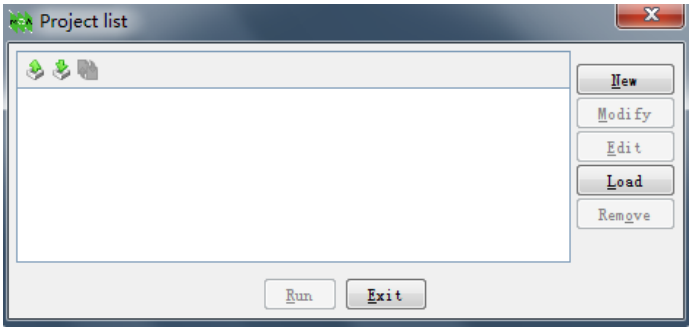

Figure 1. Project list window

The Project Wizard window will pop up. The user must then type in the name of the new project (necessary) and its description (optional), as shown in Figure 2. Click *Next*.

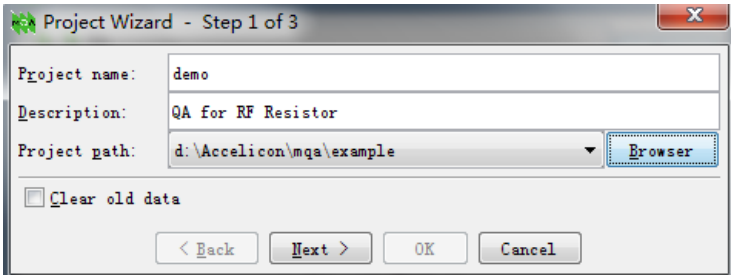

Figure 2. Project Wizard, step 1 of 3

Load Subckt Model

In the window shown in Figure 3, choose *QA* as the action type, *RF* as the application type and *Resistor* as the model type. Then, click *Add*.

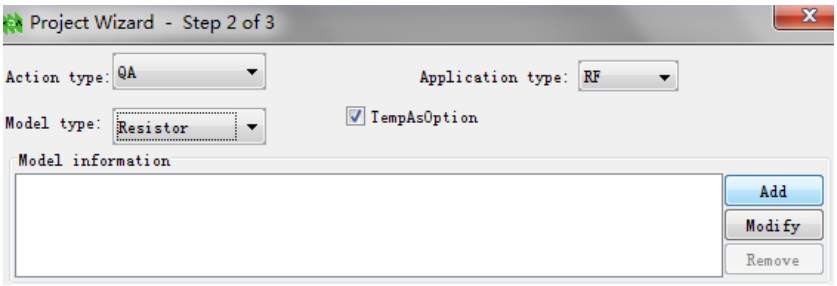

Figure 3. Project Wizard, step 2 of 3

Click *Browser* to choose the model library or model card. The model is automatically detected as a subckt model when it loads. Select *crtmom* in the Subcircuit drop down list. Make sure the Node number is correct. Instance Parameters (the instance name used in the Model and the instance name used in Rule) are normally mapped automatically. The window is shown in Figure 4.

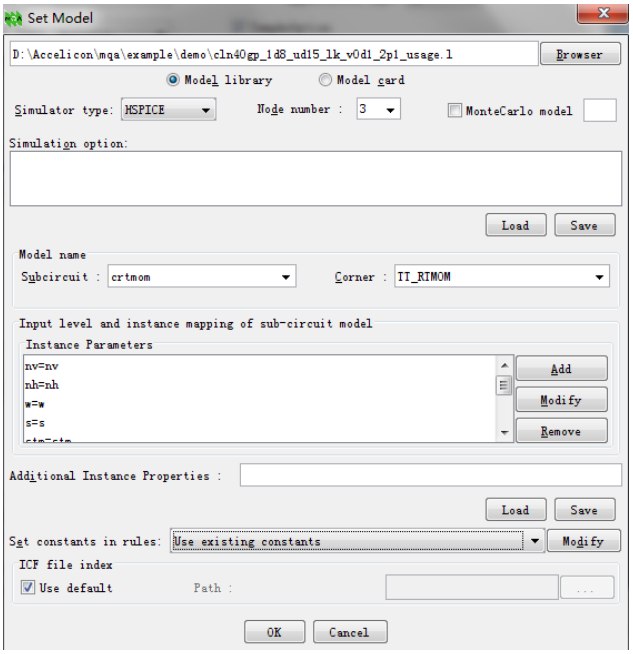

Figure 4. Set model

Click *OK*. The window with model information is shown in Figure 5.

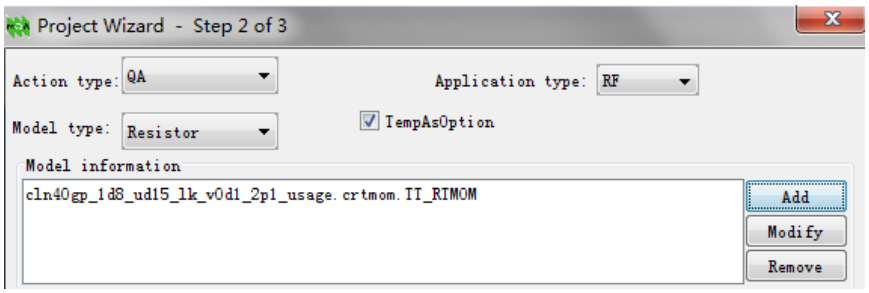

Figure 5. Project Wizard, step 2 of 3

Select Rule

A rule file is needed to run QA. Refer to the rule file used in this example in the section titled "Rule File" at the end of the application note. After loading the rule file, the window in Figure 6 is shown.

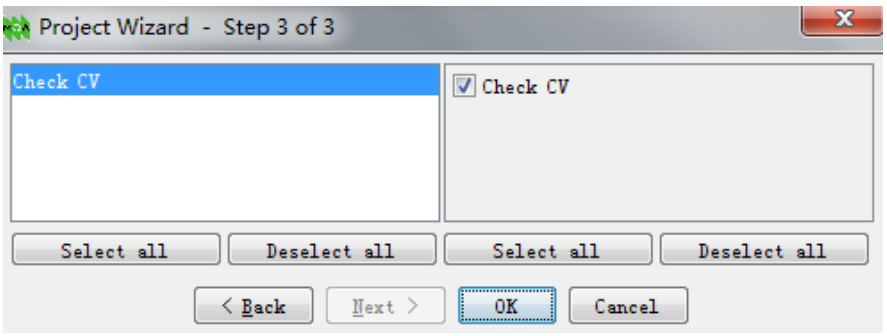

Figure 6. Project Wizard, step 3 of 3

Run

Click *OK* to return to the Project window to find the project just created (Figure 7). Click *Run* to proceed.

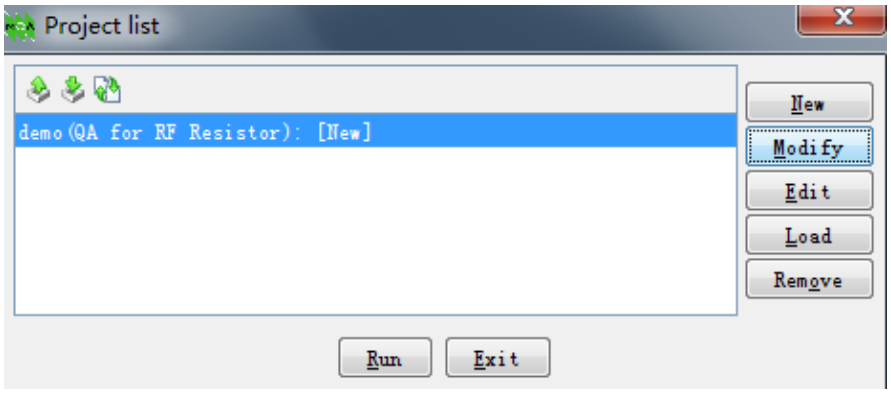

Figure 7. Project list window

Generate Report

After QA is done, right click on the project node and select *Export to Report* (as shown in Figure 8) to pop up the Report Wizard window.

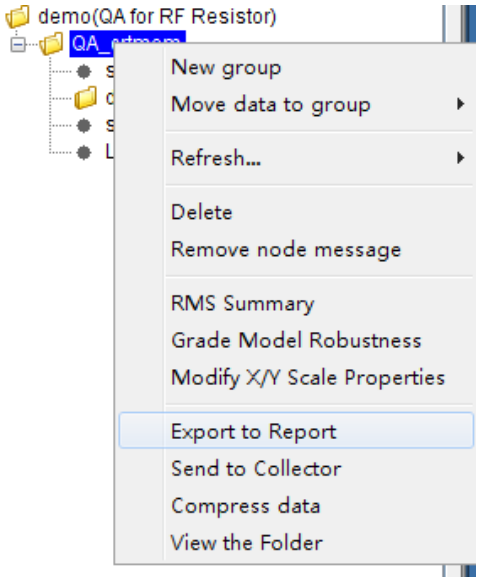

Figure 8. Export to Report

From here the user can customize the report format, as shown in Figure 9.

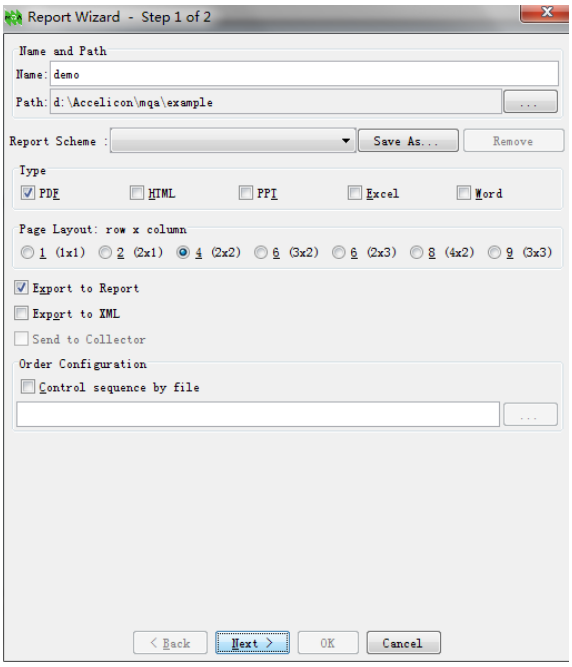

Figure 9. Report Wizard window

Sample Result

A sample report result is shown in Figure 10.

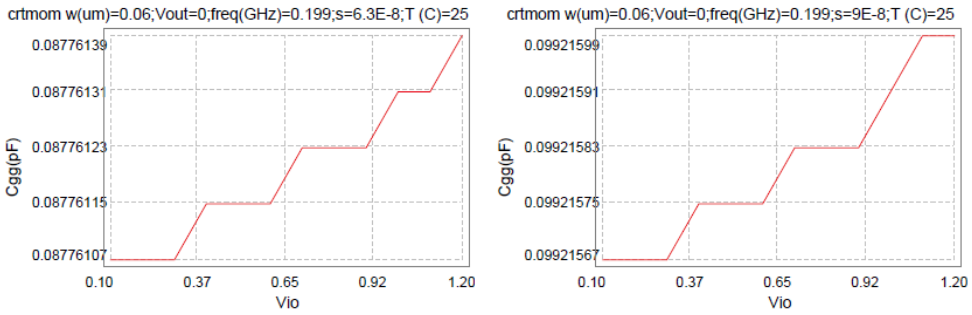

Figure 10. Sample result

Rule File

Below is the rule file with comments used in the example in this application note.

[Label: 3501:title= Check CV]
[Condition:1]
 # defines Vio, T, Vout, freq, w and s in loops section.
 # note: because there is no L in this model, so we use s instead.
 # as you see, we can sweep subckt instance parameters in loops directly, as long as we have them mapped on GUI.
[Loops
:X=Vio(start=0.1,stop=Vin,step=0.1)
:p=T(tnom)
:P1=Vout(0)
:P2=freq(start=1e8,stop=1e10,num=101)
:p3=w(0.063e-6,0.05e-6,0.06e-6,0.07e-6,0.09e-6)
:p4=s(0.063e-6,0.09e-6)
]
[Target: y1=Cgg]