

NOTICE: This document contains references to Agilent Technologies. Agilent's former Test and Measurement business has become Keysight Technologies. For more information, go to **www.keysight.com**.





Agilent Technologies

ME5

Lib Test in Model Quality Assurance

Application Note

Copyright Notice and Proprietary Information

Copyright © Agilent Technologies, Inc. 2003, 2011. All rights reserved. This software and documentation contain confidential and proprietary information that is the property of Agilent Technologies, Inc. The software and documentation are furnished under a license agreement and may be used or copied only in accordance with the terms of the license agreement. No part of the software and documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Agilent Technologies, Inc., or as expressly provided by the license agreement.

Right to Copy Documentation

The license agreement with Agilent Technologies permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies.

Disclaimer

AGILENT TECHNOLOGIES, INC. AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Agilent, Model Builder Program, MBP, Model Quality Assurance, MQA, Advanced Model Analysis, AMA, Proximity Quality Assurance, PQA are registered trademarks of Agilent Technologies, Inc.

Lib Test in Model Quality Assurance

Application Note

This application note describes how to run quick checking on the model libraries by using the Lib Test feature in Model Quality Assurance (MQA). **Note:** This document was originally released for MQA V2010.2.1.0 in September 2011.

Introduction

The Lib Test option in MQA provides the user a convenient and fast method for checking on model libraries in batch. With this capability, the user can easily find errors (e.g., syntax errors, undefined parameters and wrong links) in the library file. Lib Test supports all types of device models and can recognize HSPICE and Spectre libraries automatically.

The user can run Lib Test in either a graphical user interface (GUI) or command line mode. In this document, the application in both modes is introduced. 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.

File libtest.setup

File “*libtest.setup*” is an optional configuration file. By default, Lib Test creates a device in the netlist file using the default instance in the model. The user can also customize the instance by creating a file named “*libtest.setup*” under the working directory or the directory of libraries. For example, the user can set the instance parameters for given models like:

```
nch1      W=10u L=10u
nch2      W=1u   L=1u SA=0.1u SB=0.1u
```

Here, *nch1* and *nch2* refer to model names in the library file. Lib Test then builds the case using the device with the instance shown above rather than the default instance in the model. The resulting case for this example is as follows:

```
[hspice]
test.lib
[model]
nch w=1e-005 l=1e-005
nch w=2e-005 l=1e-005
nch w=3e-005 l=1e-005
nch w=4e-005 l=1e-005
```

Note that if the user has set multiple geometries for one device, Lib Test will only pick up the last one to build the case, which in this case would mean “*nch w=4e-005 l=1e-005.*”

Run in GUI

To run Lib Test in a GUI, the user first opens the Lib Test window by choosing *Model -> Library Test* from the main menu as shown in Figure 1.

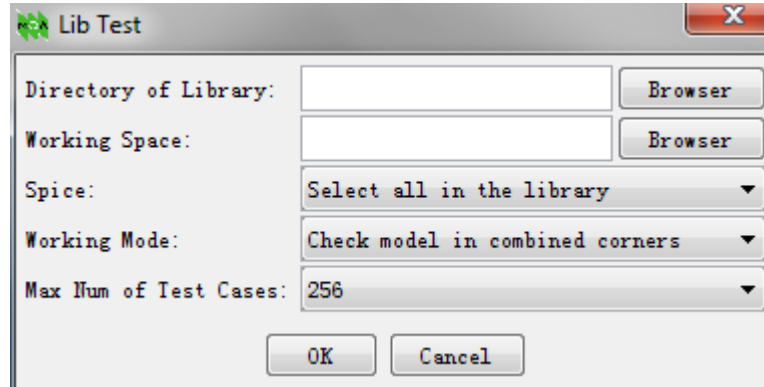


Figure 1. Lib Test

Next, click the *Browser* button of *Directory of Library* to choose the directory containing all model libraries to be tested. Click the *Browser* button of *Working Space* to choose the directory where the final result is saved. The user then needs to choose the library type to be tested from the drop down list shown in Figure 2.

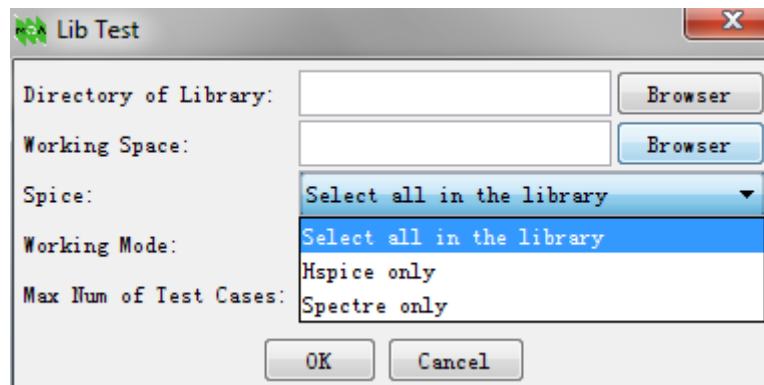


Figure 2. Choose library type

For *Working Mode*, the user has two options: to check the model in either combined corners or a single corner. When *Check model in combined corners* is selected, the program checks the combined corners from different devices. If *Check model in single corner* is selected, the program only checks on a single device corner.

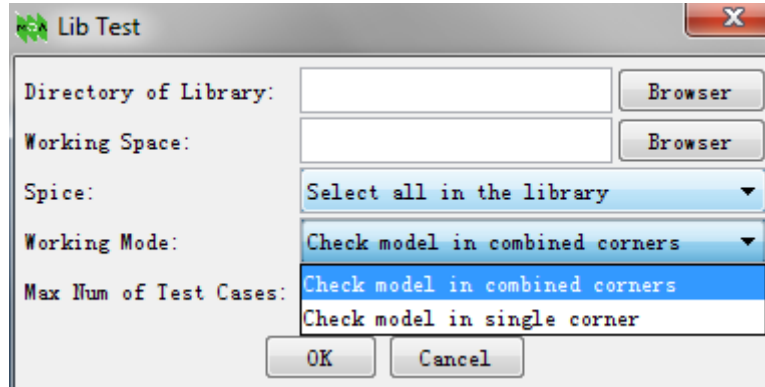


Figure 3. Working mode

For the *Max Num of Test Cases*, the user can set any positive integer desired, although the default number is 256.

After the configuration is complete, the user simply clicks the *OK* button to proceed. An HTML output file named “*summary.html*” is then automatically created under the path designated in *Working Space*.

Next, click *OK* to view the output file “*summary.html*” as shown in Figure 4.

Lib dir	models
Date	9:59:36 26/1/2011
Cmd	LibTest.exe models
Total 254 test cases are cancelled automatically to save time.	

Spice	Corner	Netlist	Error/Warning
HSPICE		th001.sp th001.lis	**error** unknown subcircuit name: **error** difficulty in reading input **error** unknown subcircuit name: **error** difficulty in reading input **error** unknown subcircuit name: **error** difficulty in reading input **error** unknown subcircuit name: **error** difficulty in reading input

Figure 4. Output file

Run in Command Line

Lib Test can also be run with a command line. The syntax is:

libtest [OPTION]... <dir of library>

In this case, the *OPTION* can be:

- *spice*: to run the given simulator(s)
 - *spice hspice* // to test HSPICE libraries only
 - *spice spectre* // to test Spectre libraries only

- *spice all* // to test all supported libraries (default)

- *level*: to define the items to be checked
 - *level 0* // to check corner by corner
 - *level 1* // to check combined corners (default)

- *maxcasenum*: the maximum number of test cases for one lib
 - *maxcasenum 256* // to build up to maximum 256 test cases (default)

After the checking is done, the output file “*summary.html*” is saved under the current directory.