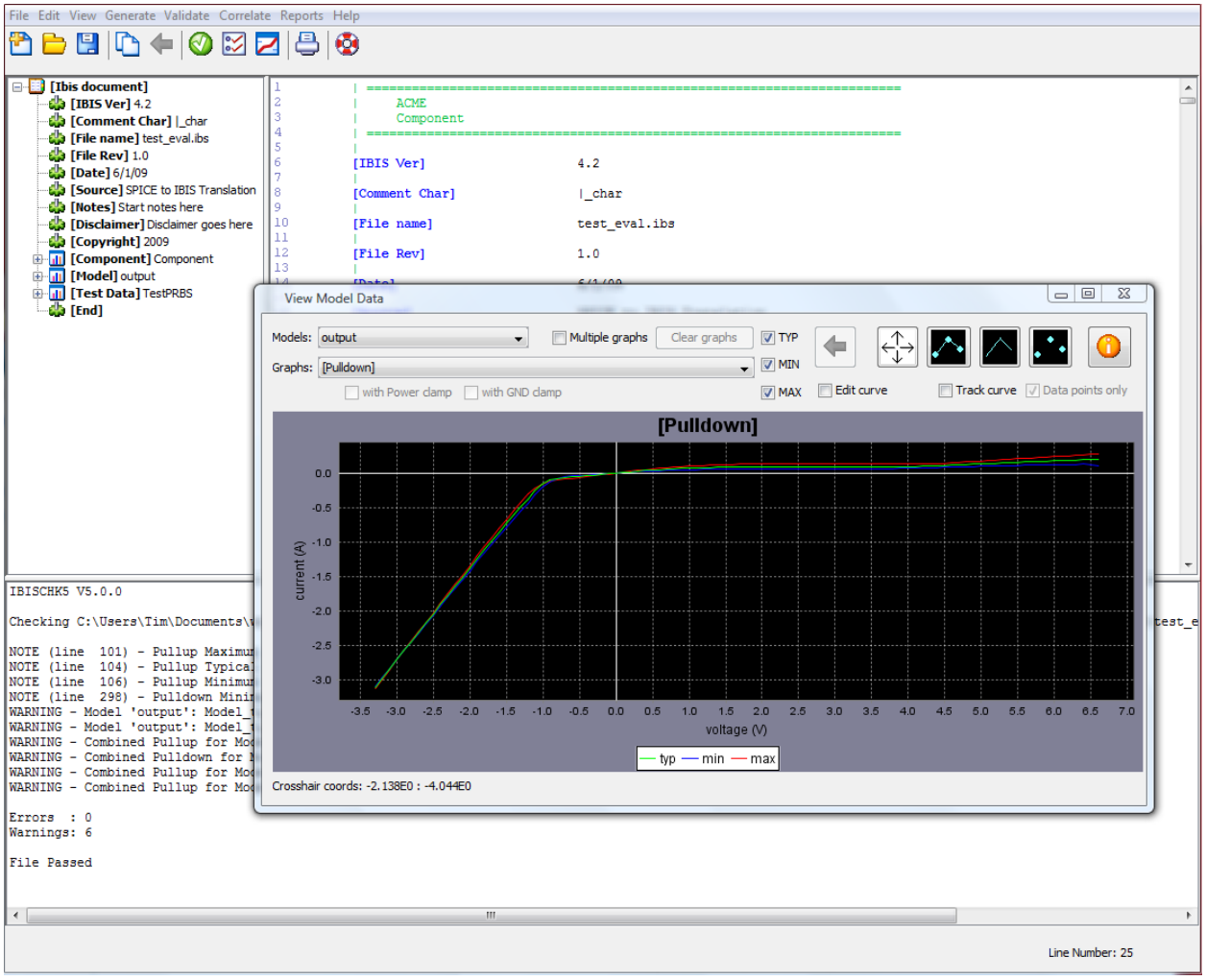
SharkSim Overview

SharkSim is a Signal Integrity modeling and simulation application developed to help engineers generate, edit, and correlate different types of simulation models for system level simulation and analysis.

SharkSim addresses the challenges of generating and correlating IBIS simulation models for PCB simulation analysis. SharkSim provides a complete IBIS modeling solution with the capability to generate IBIS models from SPICE simulations, generate IBIS model files based upon simulation or lab measurement, view/edit/graph IBIS model data, and a unique IBIS quality checker.



Getting Started

Java Installation

In order to run SharkSim the latest version of Java (version 1.6.0_23) must be installed on your machine. You can get the latest java version at http://java.sun.com/ . Only the JRE (Java Runtime Environment) is needed.

Installation Directory

Download and unzip the sharksim_install.zip file. This directory contains all of the necessary files to run SharkSim. The following is a short directory listing:

- **user_guides**: The user guides for SharkSim.
- **lib**: Application library files.
- **keywords_file**: The keywords.txt file needed by the application to parse IBIS model files.
- **SharkSim.jar**: The SharkSim application executable file for Linux is a Java jar file found in the installation root directory.
- **SharkSim.exe**: The SharkSim application executable file for Windows is an exe file found in the installation root directory.
- **spec_files**: The package specification files used for validation in the Package Editor and the device specification files used for validation in the Quality Checker.
- **reports**: User can use this directory to store reports or create a new one. A sample Quality Checker HTML report is included in this folder.
- **sample_ibis_file**: A sample IBIS file with an output model that can be used to test some of the features of the application.
- **quality_checker_batch_file**: Example setup file for batch running Quality Checker.
- **sample_prbs_file**: Sample PRBS file for use in defining test loads in application.
- **ibis_generation_files**: Example component, header, and package files for generating IBIS files.
- **test_data_examples**: Example test data files for adding [Test Data] to existing model in IBIS file.

Running SharkSim on Linux (Legacy License Manager)

If you are using the legacy license manager then use the instructions below to run SharkSim.

SharkSim can be launched from a command line terminal window by using the following command:

'java –jar SharkSim.jar' (legacy license manager)

in the installation root directory. By default the operating system does not allocate a lot of memory to the Java Virtual Machine so by using the alternative command you can allocate more memory in order to open and edit larger IBIS model files:

'java –jar –Xmx300m SharkSim.jar'

This will give 300 MB of memory for the application to open and parse larger IBIS files.

Running SharkSim on Linux

SharkSim can be launched from a command line terminal window by using the following command:

'java –Djava.library.path=./lib –jar SharkSim.jar'

in the installation root directory. By default the operating system does not allocate a lot of memory to the Java Virtual Machine so by using the alternative command you can allocate more memory in order to open and edit larger IBIS model files:

'java –Djava.library.path=./lib –jar –Xmx300m SharkSim.jar'

This will give 300 MB of memory for the application to open and parse larger IBIS files.

Running SharkSim on Windows

SharkSim can be launched from the root directory by double clicking on the SharkSim.exe application executable file. With the native Windows executable file the Java Virtual Machine memory is dynamically allocated so the application will automatically scale the memory usage to open large IBIS files.

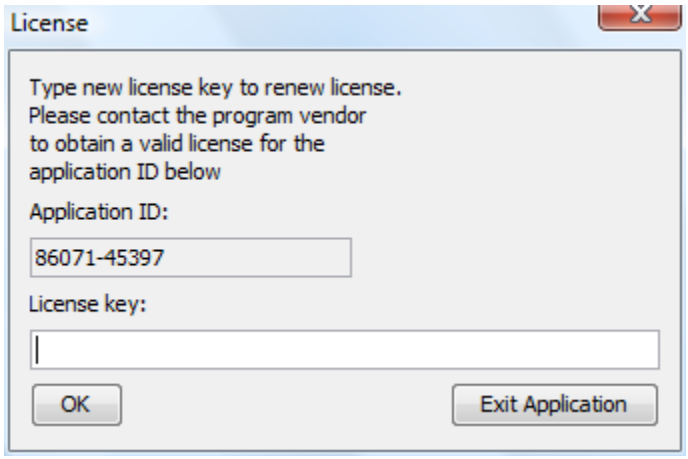
License Agreement

The first time SharkSim is run you need to read and accept the License Agreement. Then the license manager will start.

Node Locked License (Legacy)

If you are using the legacy license manager follow the instructions below.

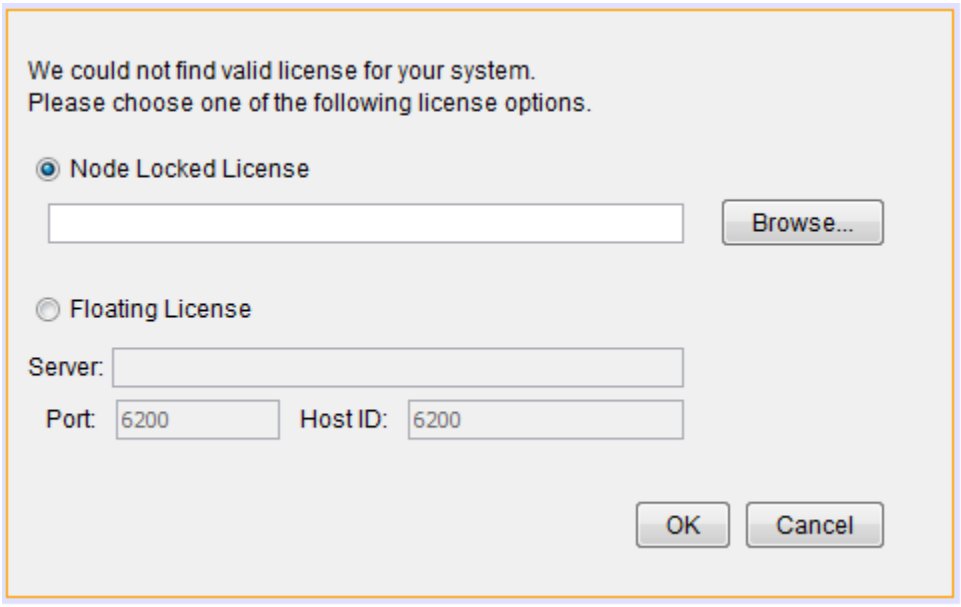
You will be prompted for your license code as seen below:



You need to email your unique Application ID to SharkSim support and than your license key code will be generated for you. When you receive your code you need to enter it in and select OK and the application will start. At any time you can view the License GUI screen by going to **View | License**.

Node Locked License

You will be prompted to select Node Locked or Floating license option as seen below:



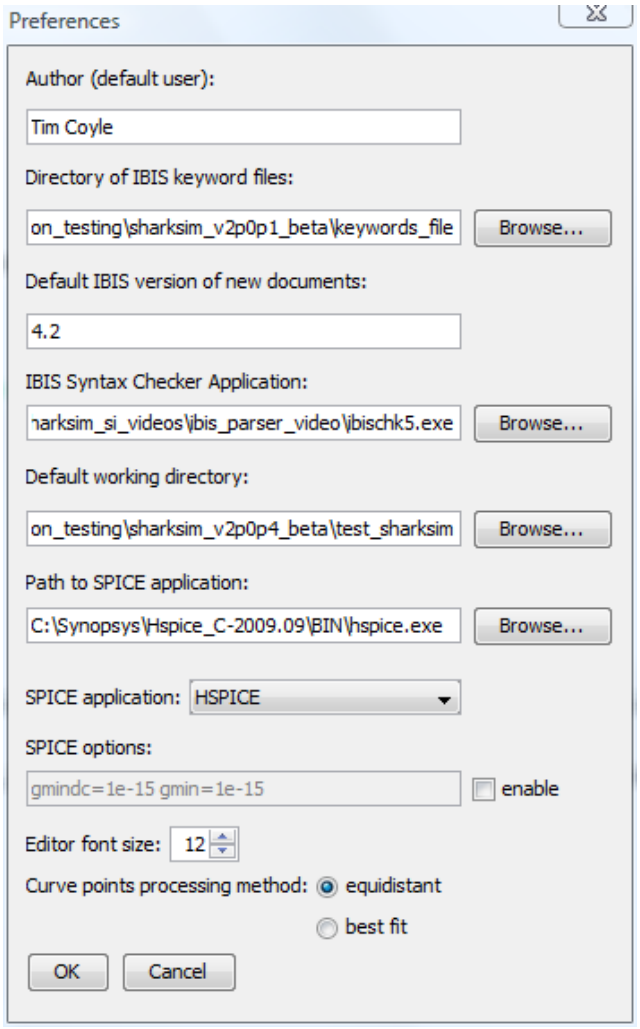
You need to browse to the location of your license file, select it, and then select OK. If successful the application will start. If for some reason the license file does not work please contact customer support. At any time you can view the License GUI screen by going to **View | License**.

Floating License

For a floating LAN license you will need to enter the name of the license server and the port used to talk to the license server. (Port 6200 is the default) Please contact customer support or your system admin for support for floating LAN license support.

Setting the Preferences

The first time you launch SharkSim you will see a dialog box instructing you to set the keywords in the Preferences menu. Go to **View | Preferences** to get the following screen:

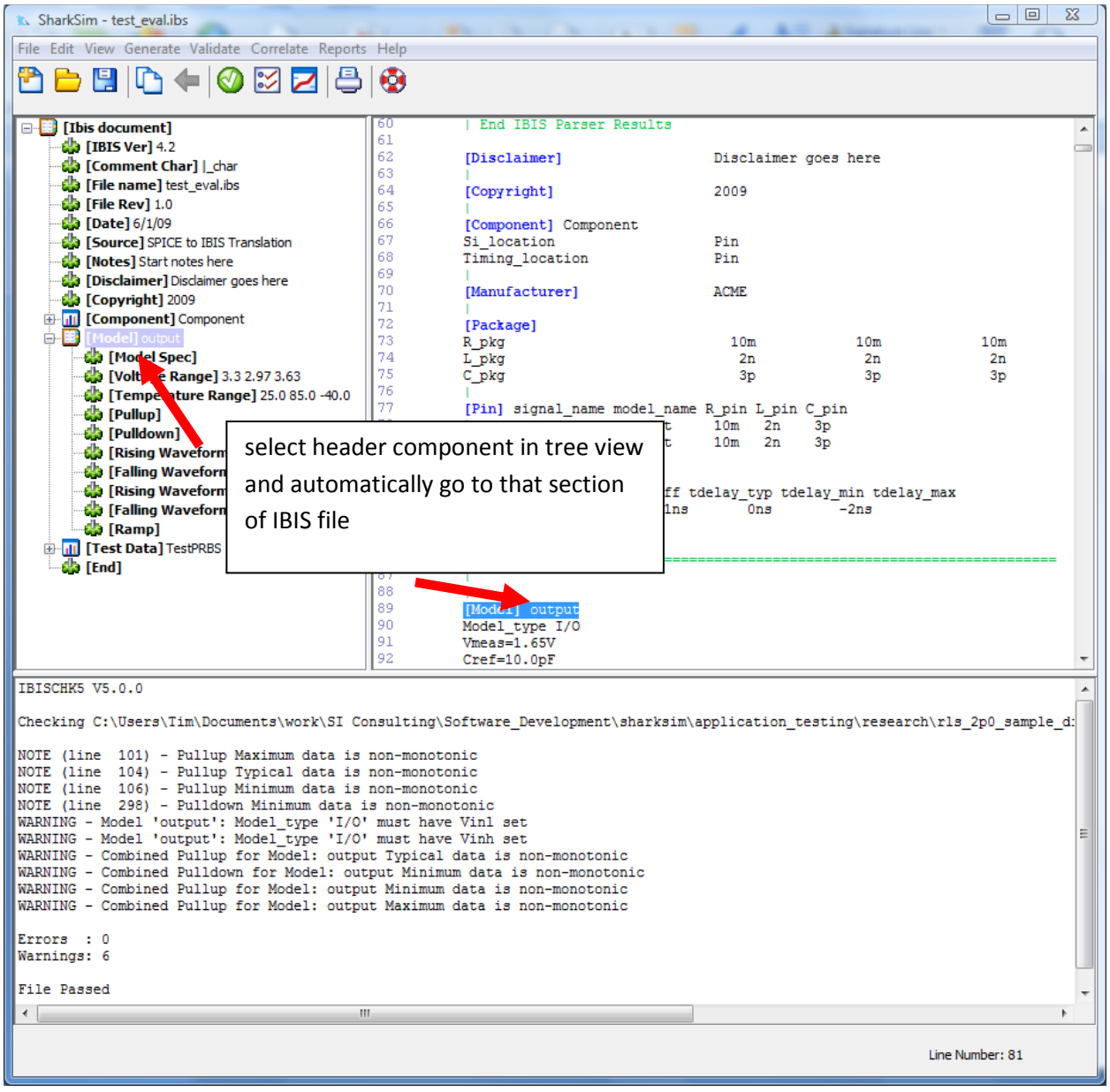


You need to browse and select the **keywords_file** folder located in the unzipped SharkSim directory folder. Set the **default IBIS version of new documents** to be a valid IBIS version (2.1, 3.2, 4.0, 4.1, 4.2, or 5.0) and this will be the default keywords file that is used when editing opened IBIS files. At any time you can change the default version by going to **File | Change Version**. This allows the user to properly edit and view IBIS files with any valid IBIS version. Next you need to download the latest IBIS Golden Parser executable for your operating system and place it in the unzipped SharkSim directory and browse and select the exe file. Links to download the browser are available under the Help section in the application. You can download the parser for free at http://www.vhdl.org/pub/ibis/ibischk4/ . You also need to set your working

directory here as well. You can set your Author Name so that when you use the Notes Manager your name will be automatically filled in. Finally you need to set the path to your SPICE simulator. You also need to select the SPICE simulator from the drop down box. You can also enable global SPICE simulation options for the IBIS generation files. Once you select OK you need to exit the program and will only have to do this once. When you restart the application all of the settings will be in place.

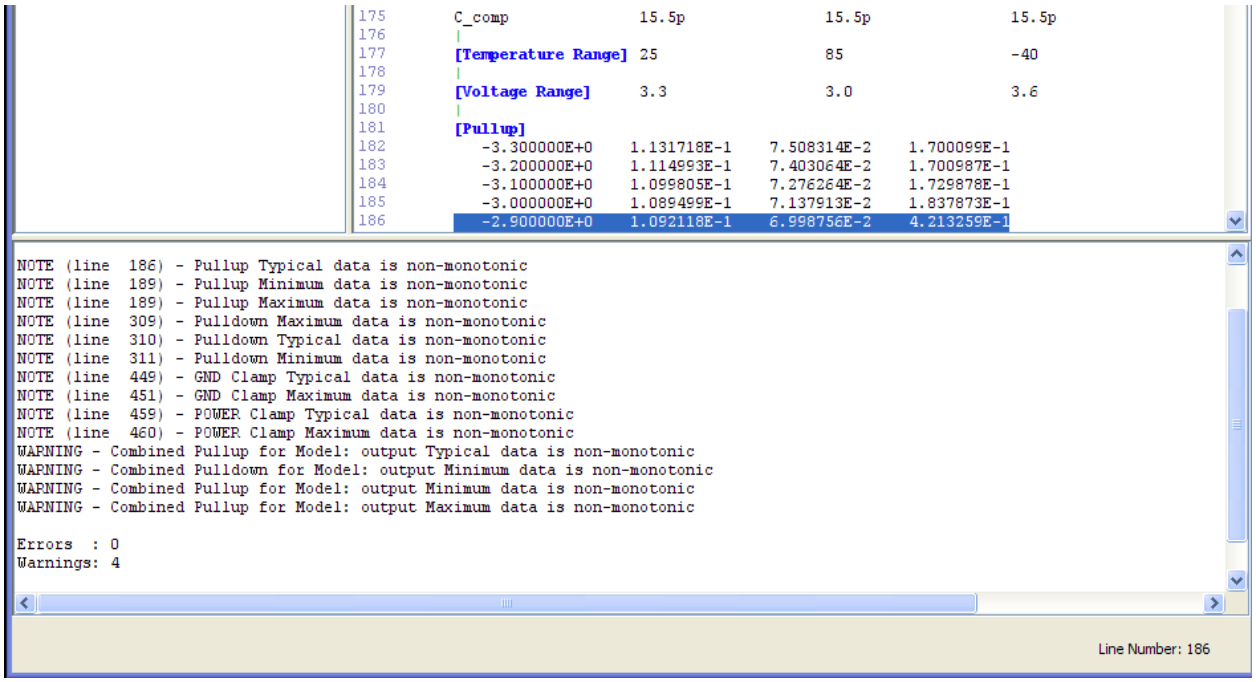
Viewing and Editing IBIS Files

SharkSim supports the latest IBIS Specification version 5.0 and you can open and edit IBIS model files in the main editor window. The major components of the IBIS file are displayed in a tree format in the left pane window with the full text being displayed in the right hand pane window. SharkSim uses the '|' pipe character as a comment by default. By selecting any of the 'headers' in the left hand pane the application will automatically go to that line in the IBIS model file.



Running the IBIS Parser

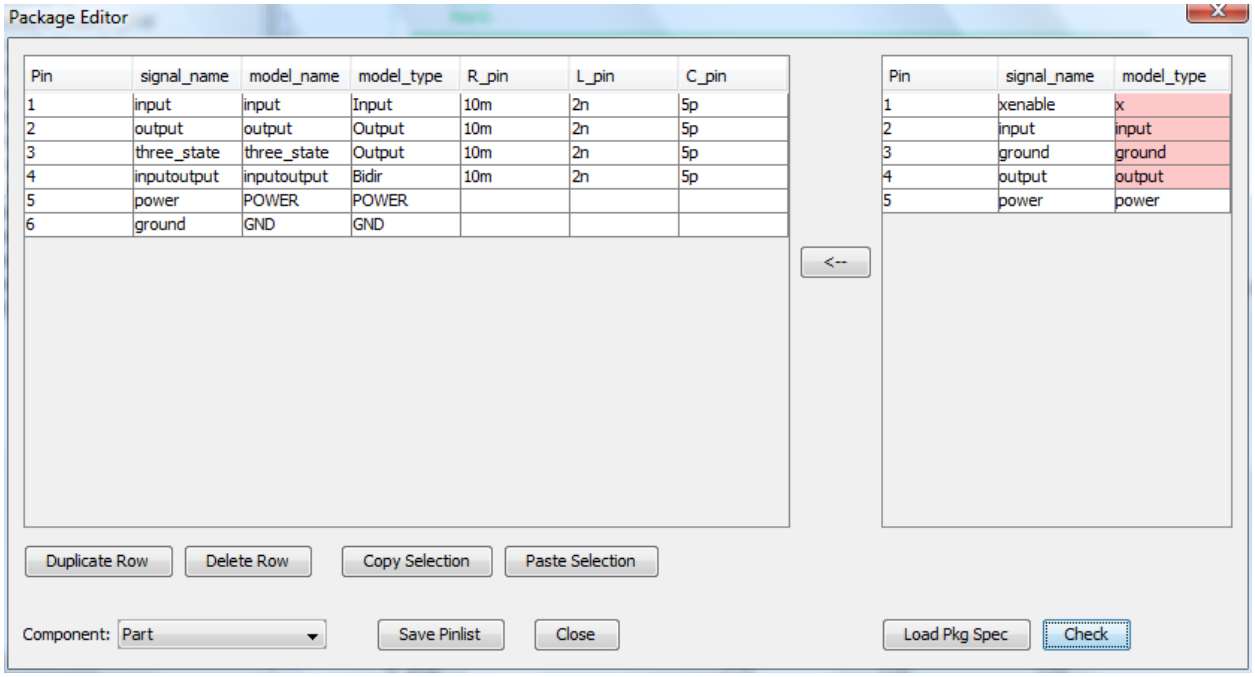
Once configured properly in the Preferences, SharkSim will run the IBIS Syntax Checker (the Golden Parser) on the IBIS file and will report any warning or error messages in the log window. The IBIS Syntax Checker can be run by selecting the green check button or under **Validate | Check Syntax**.



You can double click on any warning or error message with a line number and the editor will automatically go to that line.

Using the Package Editor

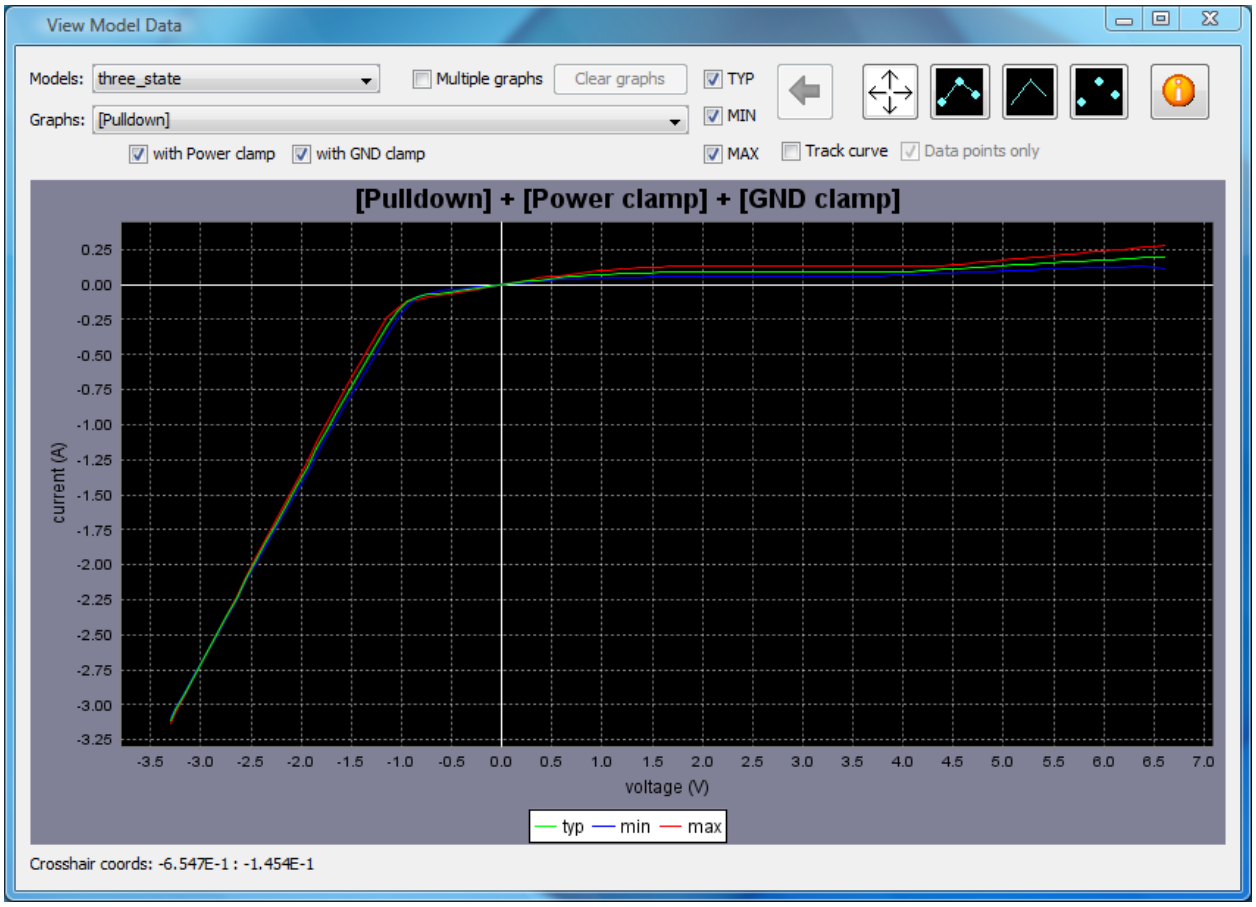
SharkSim comes with a built in editor for the package section of the IBIS model file that can be launched by going to **Validate | Package Editor**.



The Package Editor automatically reads in the package pin data from the IBIS file and places it in a table that allows the user to easily edit the data. The user can sort the column data and add/delete/copy data as well from one row to another. The user can load a package specification file and run a validation check on the pin number and model type. Any errors in the data will be highlighted. The user can also copy a row of data from the package specification section to the IBIS package pinlist section. By selecting the Save Pinlist button the user can save the new data to the package pinlist section of the IBIS model file.

Graphing IBIS Data

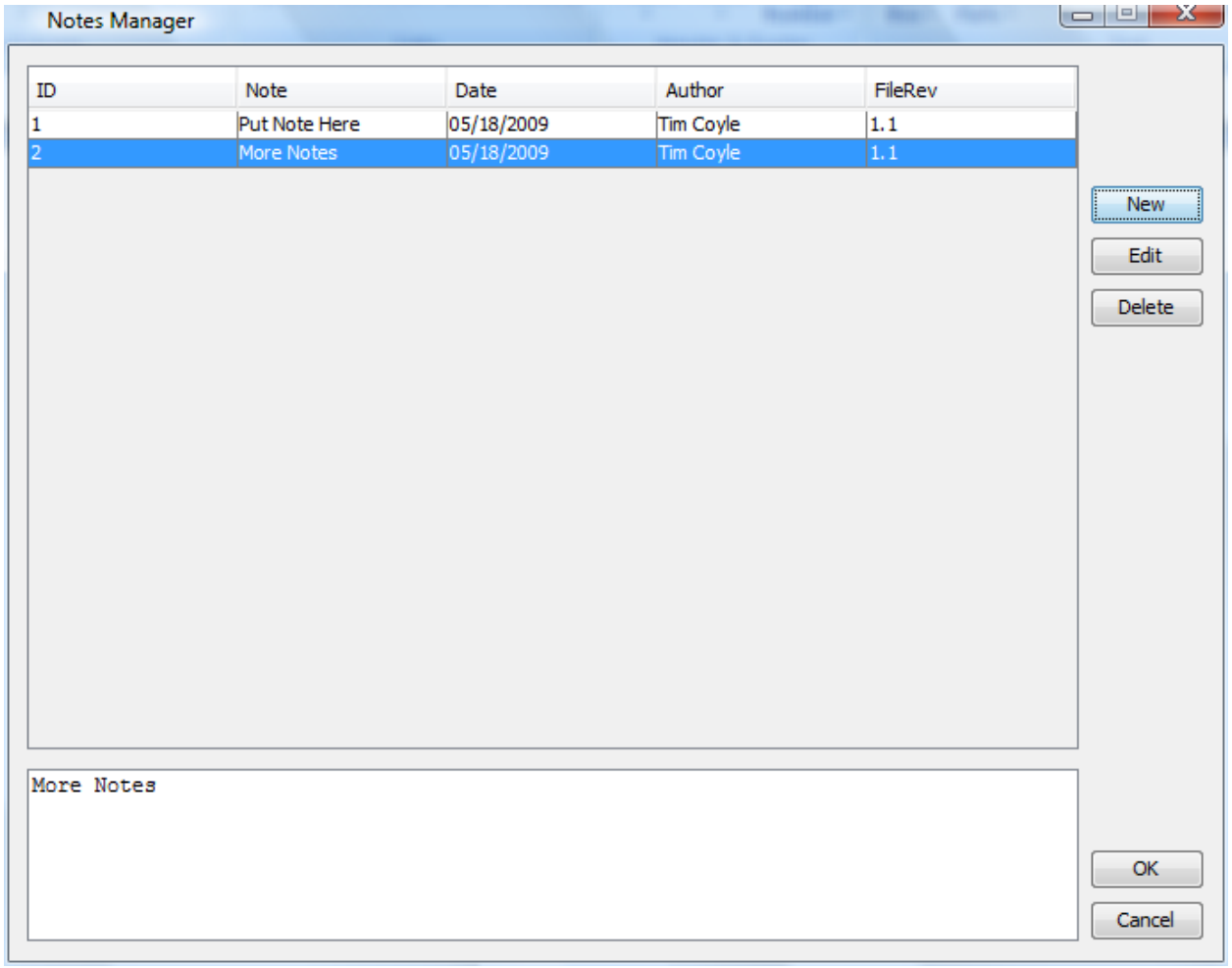
SharkSim allows the user to graph the IV and VT data in an IBIS model file. The graphing feature can be accessed by selecting the line graph icon or under the IBIS menu **Validate | Graph Data**.



The user can select from the different models available in the IBIS model file. Each data set available to graph for each model shows up in Graphs. By selecting the Multiple Graphs option the user can overlay different types of graphs. There are also options to graph the clamp and pulldown/pullup curves together. Finally, there is the ability to edit a point on a graph by moving it along the y-axis. Select the Edit curve option, left click to select a point and a white circle should appear. Without letting go of the mouse, move the point up or down to the new desired position.

Using the Notes Manager

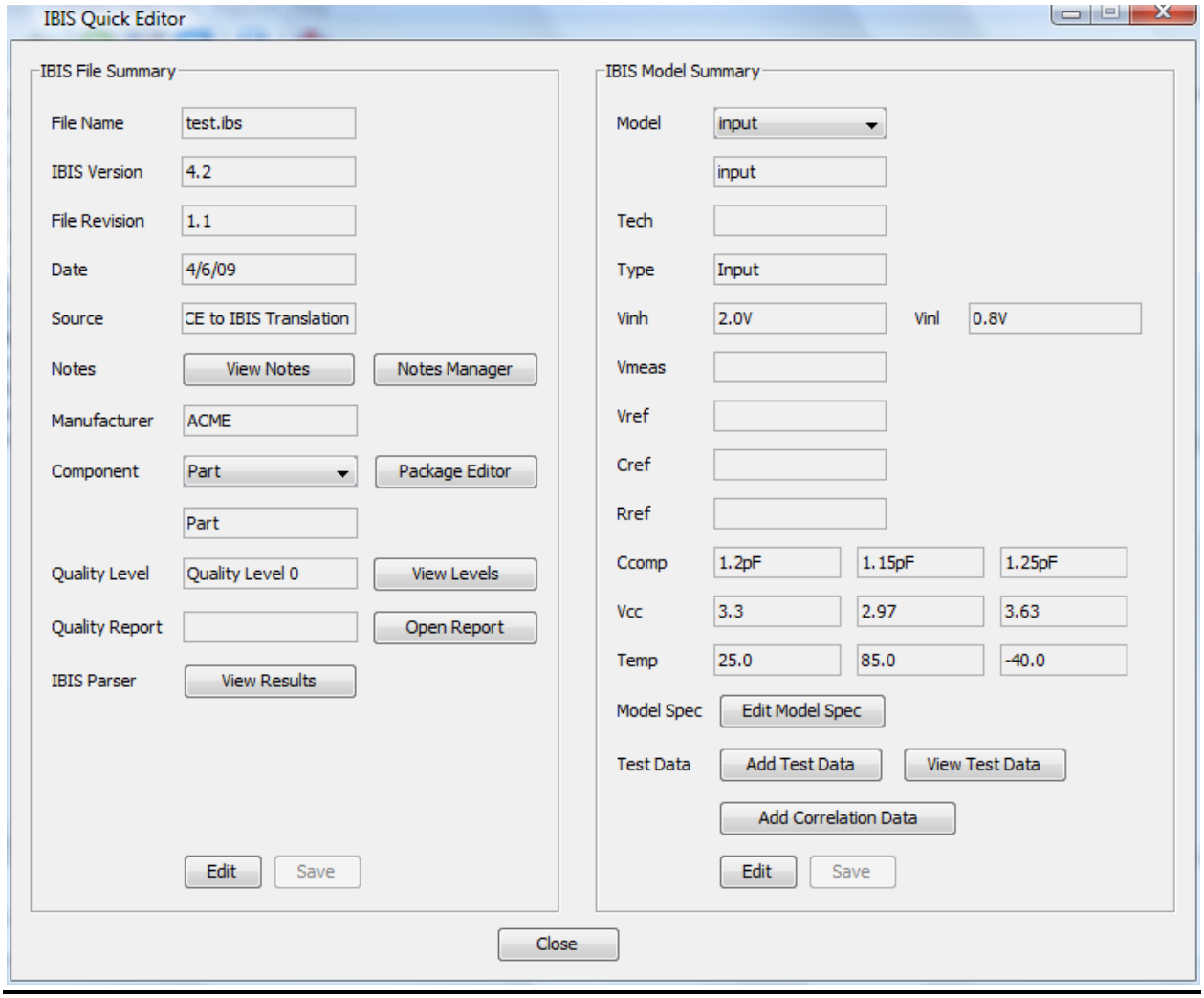
SharkSim comes with a Notes Manager to help users keep track of model changes as well as make working on a file across teams easier. The Notes Manager can be launched from **Edit | Manage Notes** menu.



SharkSim stores the notes in the [Notes] section of the IBIS file. The user can add or delete notes along with automatically updating the IBIS file revision number.

Using the Quick Editor

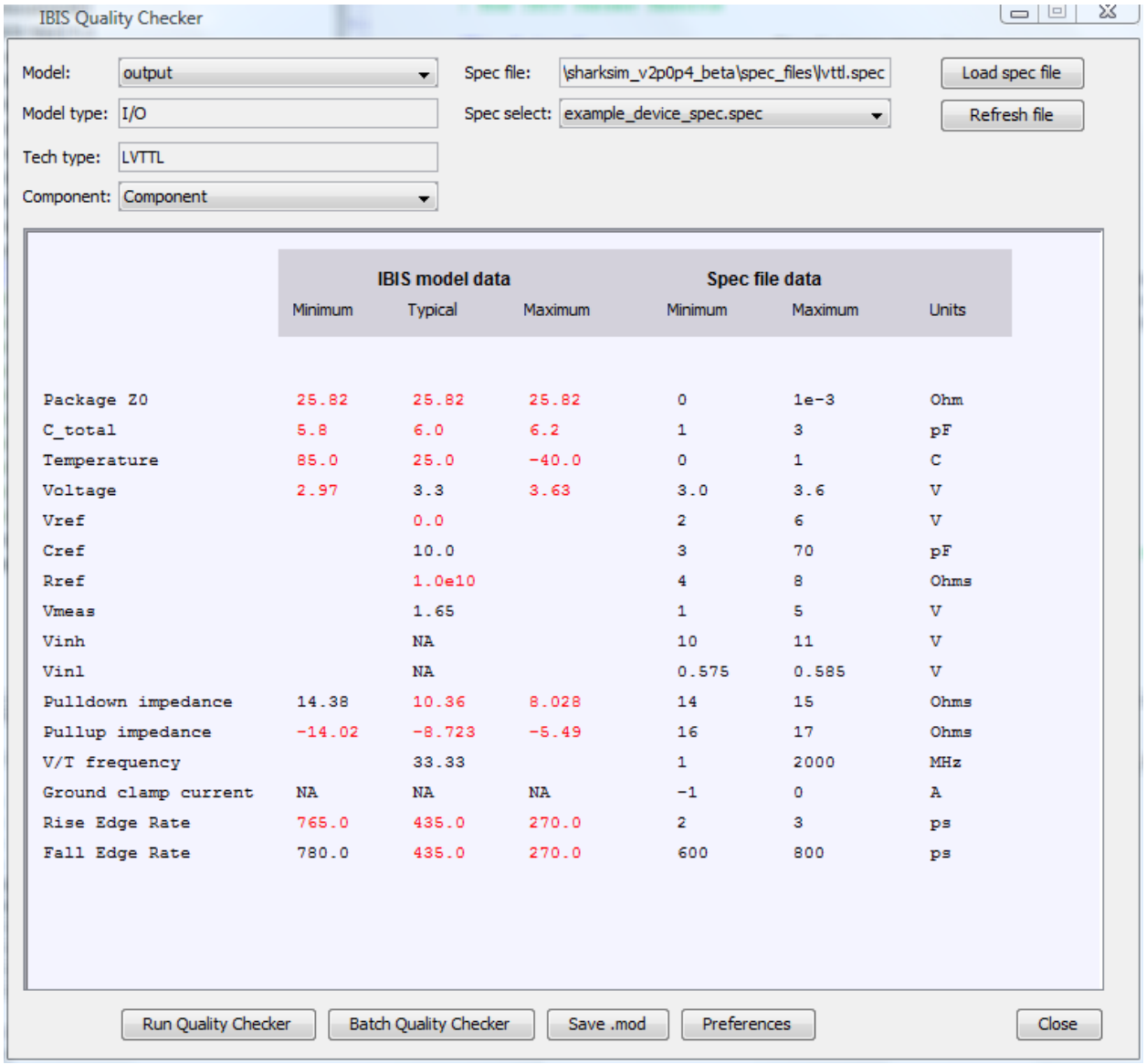
The Quick Editor allows the user to get a quick overview of the different components and models in the IBIS file and easily edit and update them. The Quick Editor can be launched from **Edit | IBIS Quick Editor** menu.



The user can view different IBIS data from the Component and Models section quickly in the Quick Editor. By selecting the Edit button the user can edit and update the IBIS data within the Quick Editor GUI without having to manually go and find the right data in the IBIS file. The user can also add new data to Models in the file.

Using the Quality Checker

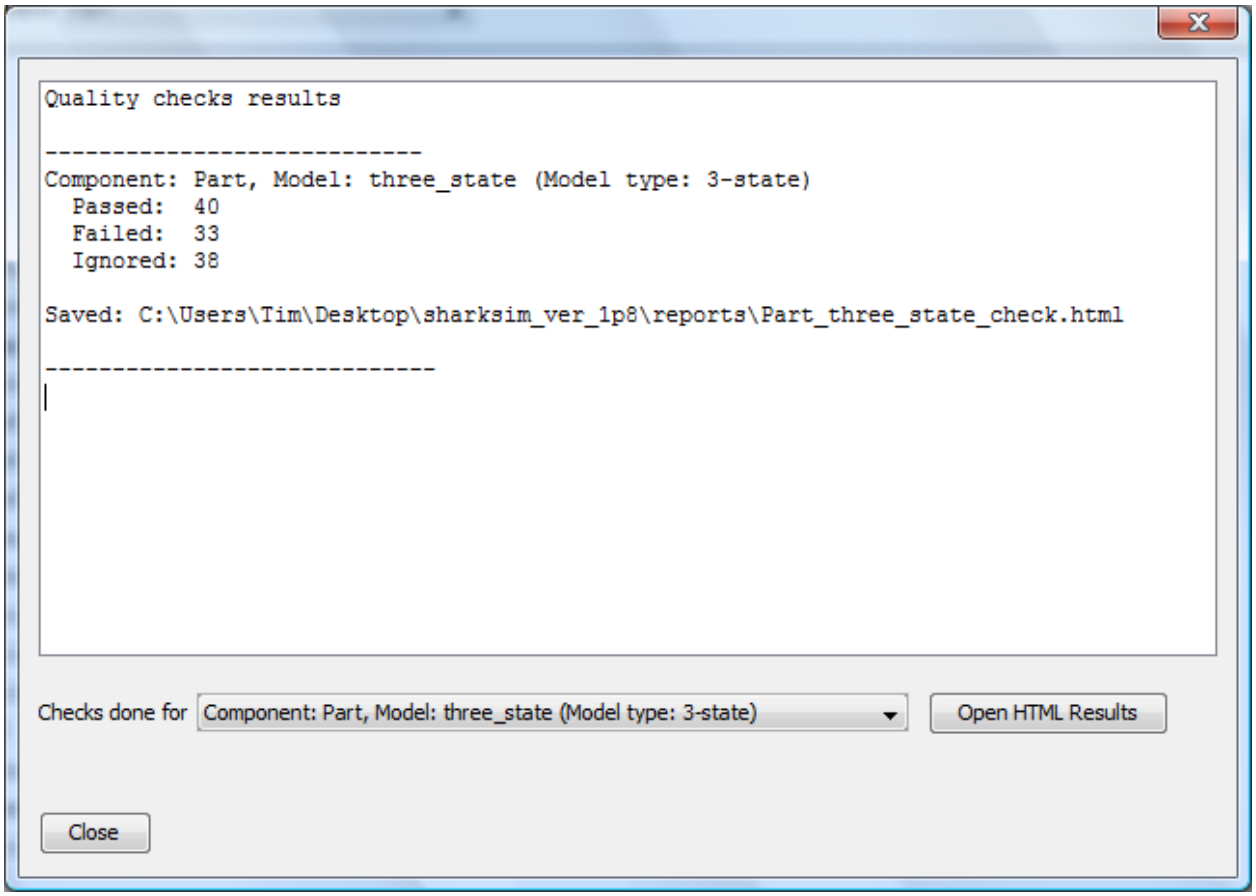
SharkSim comes with a built in quality checker that goes beyond the quality checks that the IBIS Syntax Checker provides. The quality checker can be launched by selecting the checkmark document icon or under the IBIS menu **Validate | Model Quality Checks**.



The quality checker uses a .spec file to compare various parameters in the IBIS file to specified values to see if they are within spec. The values in the .spec file will come from a component's datasheet or an interface specification such as PCI. Once the .spec file is loaded if any of the IBIS values for the selected model component are out of spec they will be highlighted in red. By

using the .spec file in this way it is meant to give the user a quick overview of the quality of the model component in the IBIS file.

To run the full quality checks, the user can select the Run Quality Checker button. The application will then perform a large set of different quality checks. Some of the checks will use the .spec file. The final results are displayed in an output window:



In order to keep track of quality issues for different model components in an IBIS model the user has the option to view the quality check report as an HTML file as seen below. The HTML Report shown is only a clipping of the full report.

Quality checks results for model **three_state** (model type: 3-state) and component **Part**

Time stamp: 2009/05/18 05:17:19

Passed: 40

Failed: 33

Ignored: 38

Check	Typ	Min	Max	Spec min	Spec max	Units	Typ Result	Min Result	Max Result	Info
Package Z0	20.0	20.0	20.0	0	1e-3	Ohm	Fail	Fail	Fail	Typical value is greater than specified. Minimum value is greater than specified. Maximum value is greater than specified.
C_total	7.8	7.7	7.9	1	3	pF	Fail	Fail	Fail	Typical value is greater than specified. Minimum value is greater than specified. Maximum value is greater than specified.
Temperature	25.0	85.0	-40.0	50	NA	C	Fail	Pass	Fail	Typical value is less than specified. Maximum value is less than specified.
Voltage	3.3	2.97	3.63	3.0	3.3	V	Pass	Fail	Fail	Minimum value is less than specified. Maximum value is greater than specified.
Vref	0.0			2	6	V	Fail	Ignore	Ignore	Typical value is less than specified.
Cref	10.0			3	70	pF	Pass	Ignore	Ignore	
Rref	1.0e9			4	8	Ohms	Fail	Ignore	Ignore	Typical value is greater than specified.

This HTML report easily identifies what quality check was performed and the pass/fail/ignore status. If a .spec parameter was used the min/max spec parameters are listed.

The user can also run the Quality Checker in batch mode to validate all of the buffer models in an IBIS file and create HTML reports.

Defining A Spec File for Quality Checks

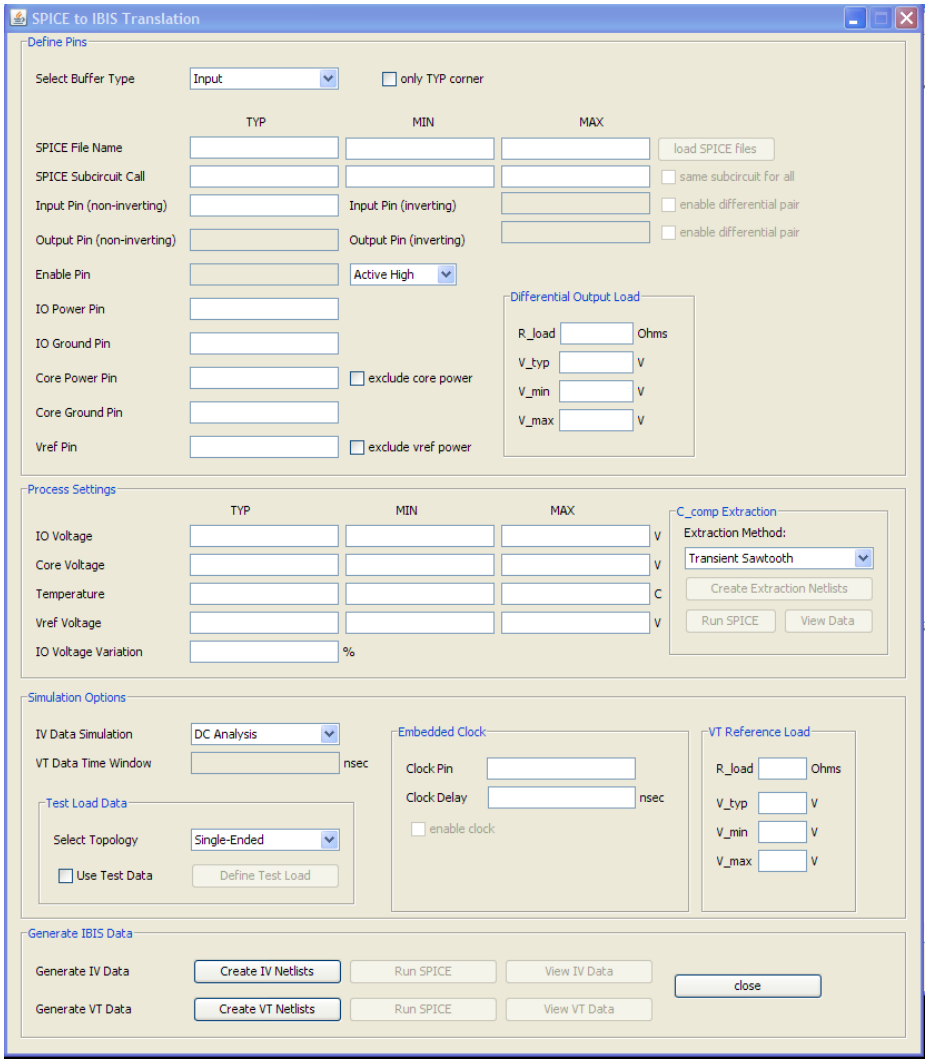
SharkSim uses a .spec file to run quality checks on an IBIS model file. The .spec file is an ANSI text file and a user can create a custom .spec file. The following is a sample of checks that are available:

```
# comment lines like this and empty lines allowed

# Format:
# [check_name][unit_name] = min,max
# The special value NA is allowed for min or max if not applicable

[Voltage Range][Volts]      = 3.0, 3.3
[Temperature Range][C]      = 50, NA
[C_total][pF]               = 1, 3
[Package Z0][Ohm]           = 0, 1e-3
[Vmeas][Volts]              = 1, 5
[Vref][Volts]               = 2, 6
[Cref][pF]                  = 3, 70
[Rref][Ohms]                = 4, 8
[Vinh][Volts]               = 10, 11
[Vinl][Volts]               = 12, 13
[Pulldown impedance][Ohms] = 14, 15
[Pullup impedance][Ohms]    = 16, 17
[V/T frequency][MHz]        = 1, 2000
[Ground clamp current][A]   = -1, 0
[Rise Edge Rate][ps]        = 2, 3
[Fall Edge Rate][ps]        = 600, 800
[Pin Z0][Ohms]              = 84, 85
[Pin C_total][pF]           = 10, 20
[Model Spec Vinl][Volts]    = 0.5, 0.6
```

More spec parameters and checks will be added in future releases but the Quality Checker is currently compliant with the IBIS Quality Specification 2.0.

SPICE to IBIS Translation

SharkSim allows the user to translate SPICE models to IBIS models by taking a SPICE model and extracting the correct IV and VT data to create an IBIS model file. The first step is to launch the SPICE to IBIS interface that will run the SPICE simulations to generate the IBIS IV and VT data. The SPICE to IBIS interface can be launched by going to **Generate | SPICE to IBIS Translation**.



The SPICE to IBIS interface works by the user creating a subcircuit wrapper around their current SPICE model. Then the user enters the necessary information in the GUI window and the tool will create the IV and VT netlists that a SPICE simulator will run to generate the output listing data. The user can select DC or AC Transient simulation methods.

Creating a Subcircuit Wrapper

The SPICE to IBIS interface needs to have a subcircuit defined in the following form:

Xcall In Out Enable IOPower IOGround CorePower CoreGround Vref subckt

Depending on the buffer type of the circuit some of the nodes may not be needed. The subcircuit nodes have to be in this exact order in order for the tool to read it properly. The core power and ground nodes and vref node is optional. The user can select the exclude checkboxes to not include these node definitions in the tool.

The protocol is the following:

- **Input** => Xcall In IOPower IOGround (CorePower CoreGround Vref)
- **Output** => Xcall In Out IOPower IOGround (CorePower CoreGround Vref)
- **Input-Output**=> Xcall In Out Enable IOPower IOGround (CorePower CoreGround Vref)
- **3-State Output**=> Xcall In Out Enable IOPower IOGround (CorePower CoreGround Vref)
- **Differential**=> Xcall In+ In- Out+ Out- Enable IOPower IOGround (CorePower CoreGround Vref)

For each model type generated, a separate subcircuit wrapper is needed. For example, an input buffer will require a separate wrapper from an output buffer type. Also, there needs to be a separate file for each process corner.

The following example will demonstrate how to create a subcircuit wrapper.

Steps to Create A Subcircuit Wrapper For Output Buffer

1. Create a top-level SPICE netlist file that can properly simulate the output buffer into the manufacturers specified test load. This netlist file should include all other necessary subcircuit calls and library/process calls if they are not embedded into the file.
2. There should be one top level instance call that does not include a package model. An example would be: **Xdie Input_A Sense_B Output_C VCC GND buffer**. From the previous specified protocol for an output buffer we need an input, output, power and ground. We have an extra node, Sense_B that is not needed for the IBIS translation. Also, the protocol specifies that the node order needs to be input than output as well. In the next step we will create a new subcircuit wrapper to address this
3. A subcircuit wrapper should be created around the current subcircuit call with the correct format and ordering.

Here is the old toplevel subcircuit call:

```
Xdie Input_A Sense_B Output_C VCC GND buffer

.subckt buffer Input_A Sense_B Output_C VCC GND

* circuit definitions and process info goes here

.ends buffer
```

Now we will create a new wrapper:

```
Xdie Input_A Output_C VCC GND buffer2

.subckt buffer2 Input_A Output_C VCC GND

Vsource Sense_B 0 DC 0
Xdie Input_A Sense_B Output_C VCC GND buffer

.ends buffer2


.subckt buffer Input_A Sense_B Output_C VCC GND

* circuit definitions and process info goes here

.ends buffer
```

We did two things here: we changed the node name order and set the extra node that was needed to the appropriate logic level. A new subcircuit buffer2 was created and the previous subcircuit call buffer was invoked from within this subcircuit.

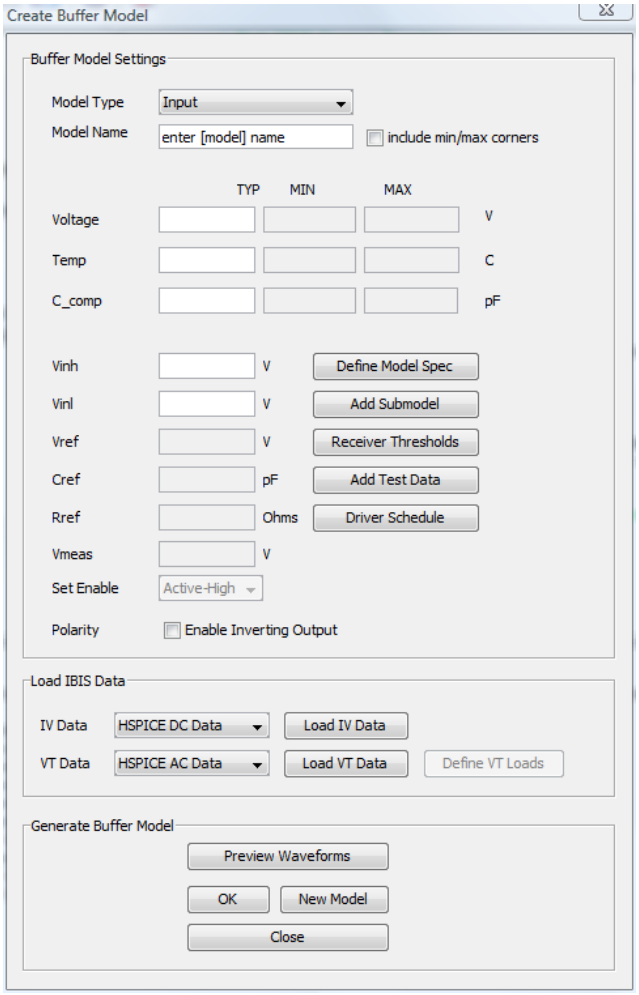
There are many different ways to create subcircuit calls to format an existing SPICE netlist to work properly with SharkSim and this is only one method.

Creating An IBIS File

Once a user has either simulated data or lab measurement data a full IBIS file can be generated using SharkSim. This is done in two steps: first the model buffer component is generated and then the complete IBIS file with header and component information is created.

Create a Model Component

Simulated or lab measured data can be used to generate a model buffer component [Model] to be included in a full IBIS file. This can be launched by going to **Generate | Create Buffer Model**.

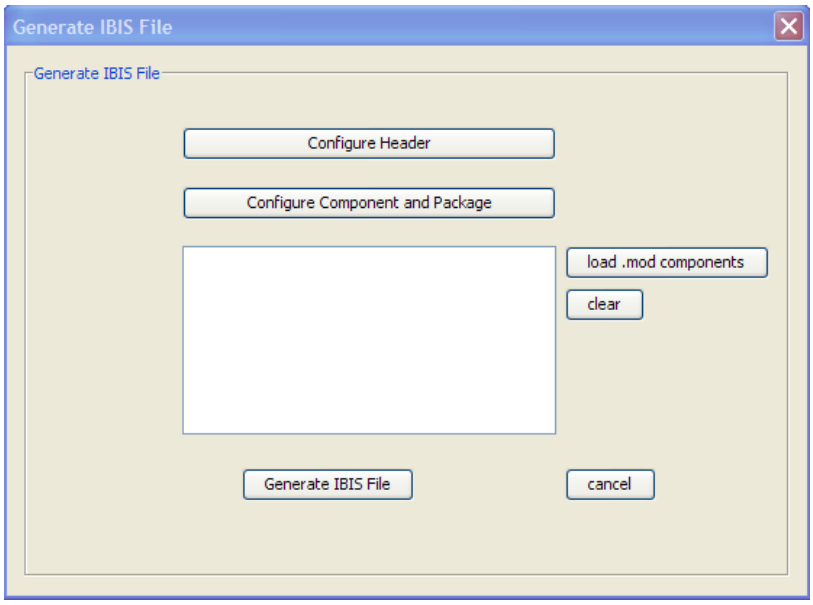


The user selects the model type and enters the various information. The user can select whether the simulation data was from DC or AC Transient or if from lab measurement. If the model type requires VT data the user will need to select the define VT loads button and enter the appropriate information. The preview waveforms button will allow the user to view the formatted IV and VT

data. The user can select ok to create a .mod file to be used later in the IBIS file generation process.

Generate IBIS File

Once the user has created a .mod file that represents a model buffer component the full IBIS file can be generated. This can be launched by going to **Generate | Generate IBIS File**.



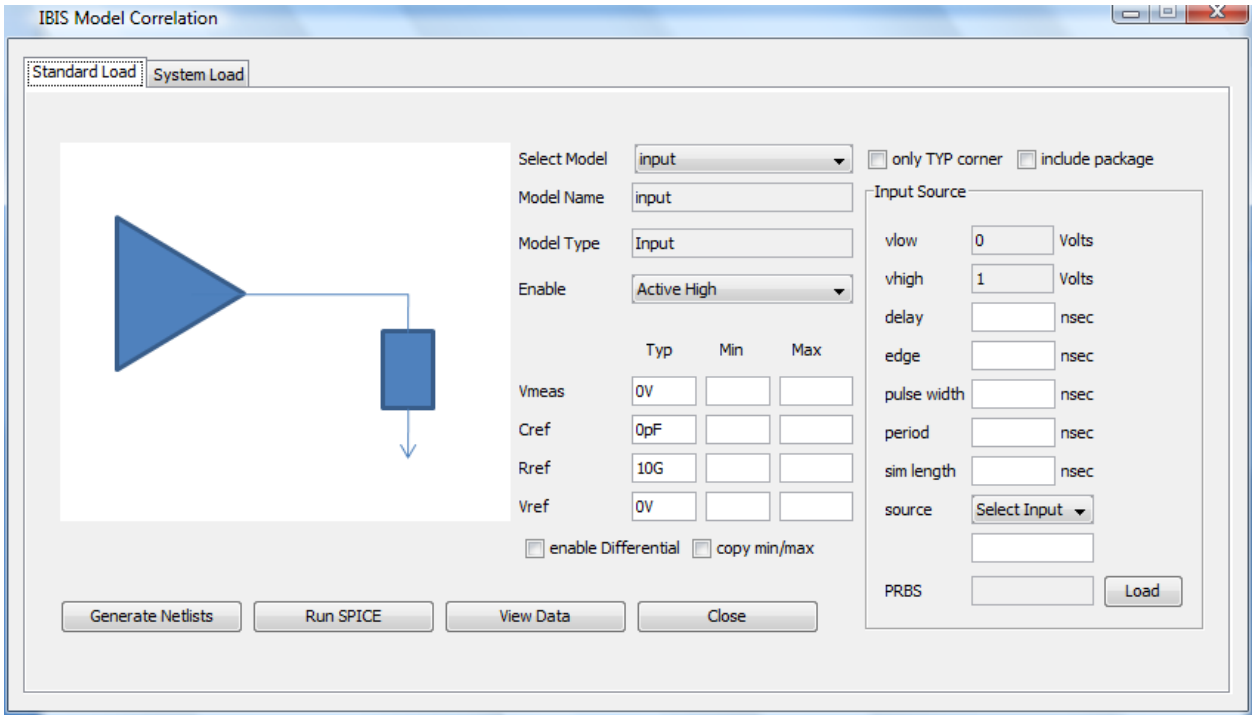
The user needs to select the configure header button to fill out the various data fields and the same with the configure component and package section. For this section, the user can upload various package and pin files to define the IBIS file. The pin file represents the RLC of each package pin and must be in a .pin file. An example is below:

[Pin]	signal_name	model_name	R_pin	L_pin	C_pin
1	enable	input	32m	1.01n	0.13p
2	input	input	30m	0.90n	0.13p
3	ground	GND	22m	0.91n	0.13p
4	output	output	22m	0.91n	0.13p
5	power	POWER	32m	1.01n	0.13p

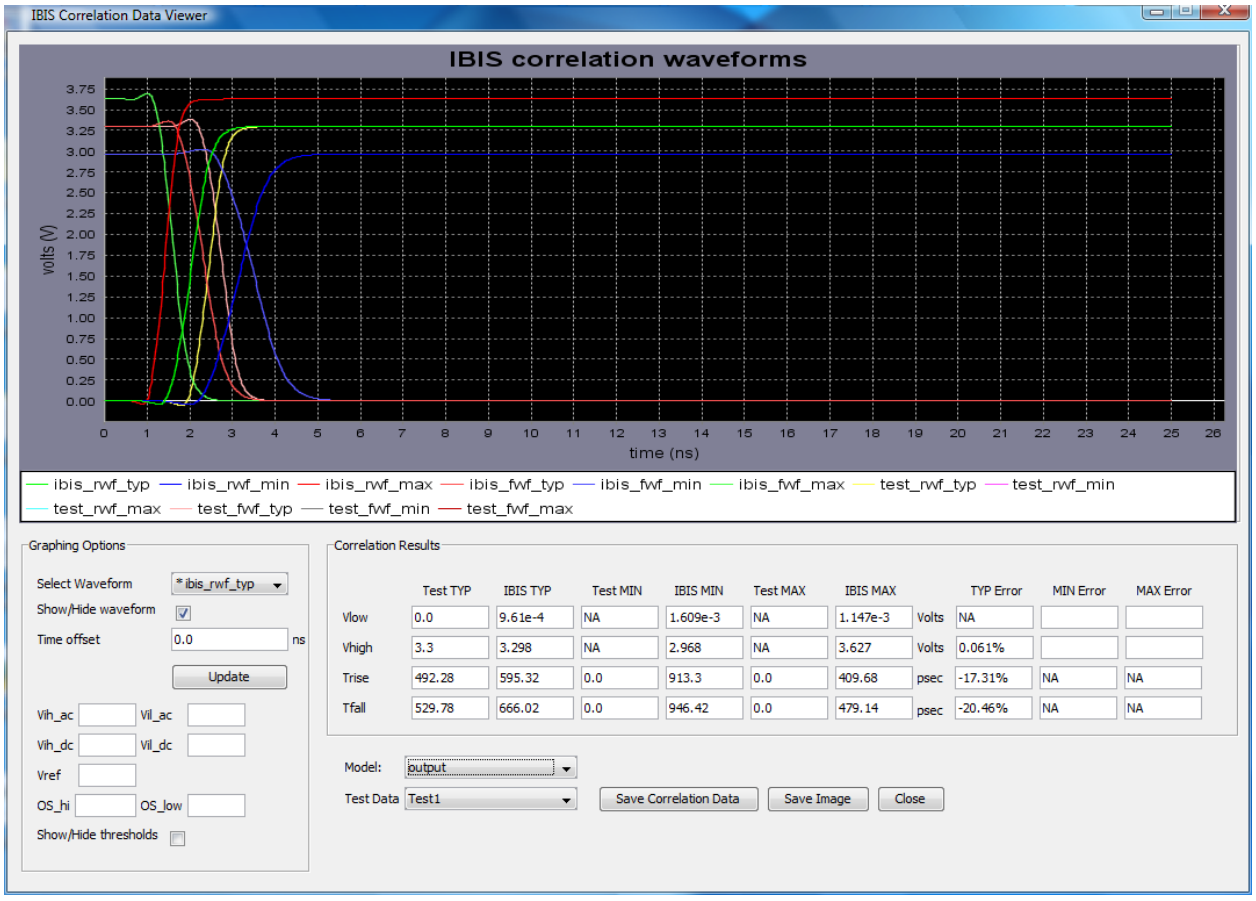
The users can than browse and select various .mod files to be included in the final IBIS file.

IBIS Model Simulation and Correlation

Using a supported external simulator like Synopsys HSPICE the user can select different test loads and automatically simulate the IBIS buffer model and correlate the data back to the original source. The IBIS Model Simulation GUI window can be launched from the **Correlate | IBIS Model Simulation** menu.



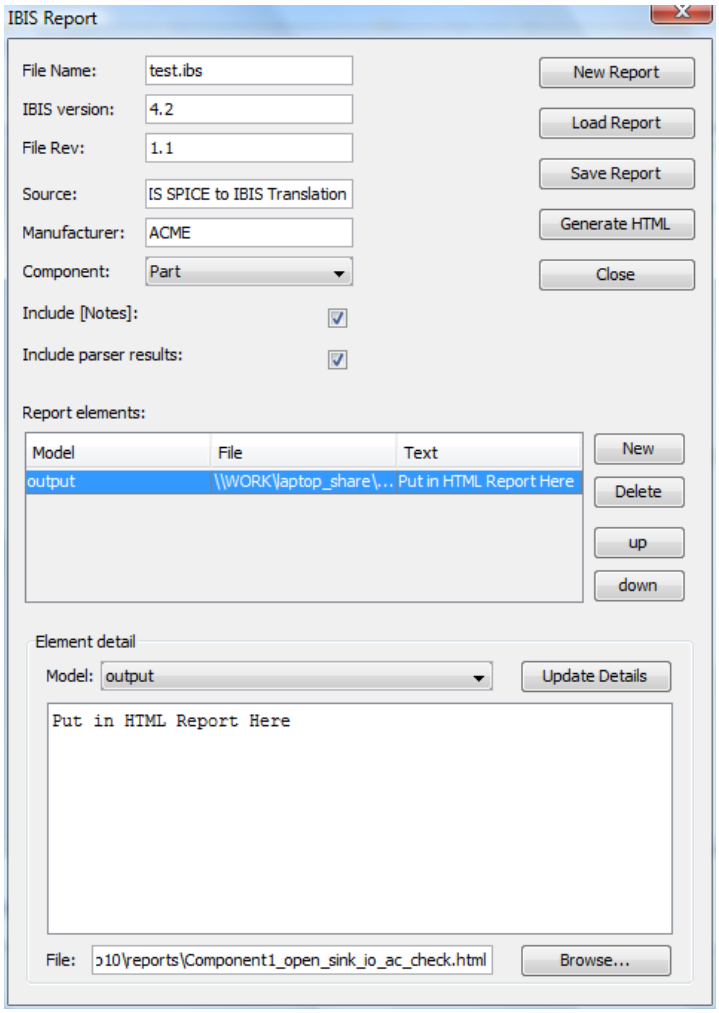
The user can use the Standard Load from the IBIS file or select a System Load and enter in their own test load including transmission line loads. When the user views the data if there is existing Test Data in the IBIS file (for correlation) the user can correlate the IBIS data to another source.



SharkSim automatically generates the correlation metric data based upon the output voltage high and low levels as well as the rising and falling edge rates. The user can save this data for later user as well as save screenshots. The user can also turn different waveforms on and off and add threshold markers.

Generating Reports

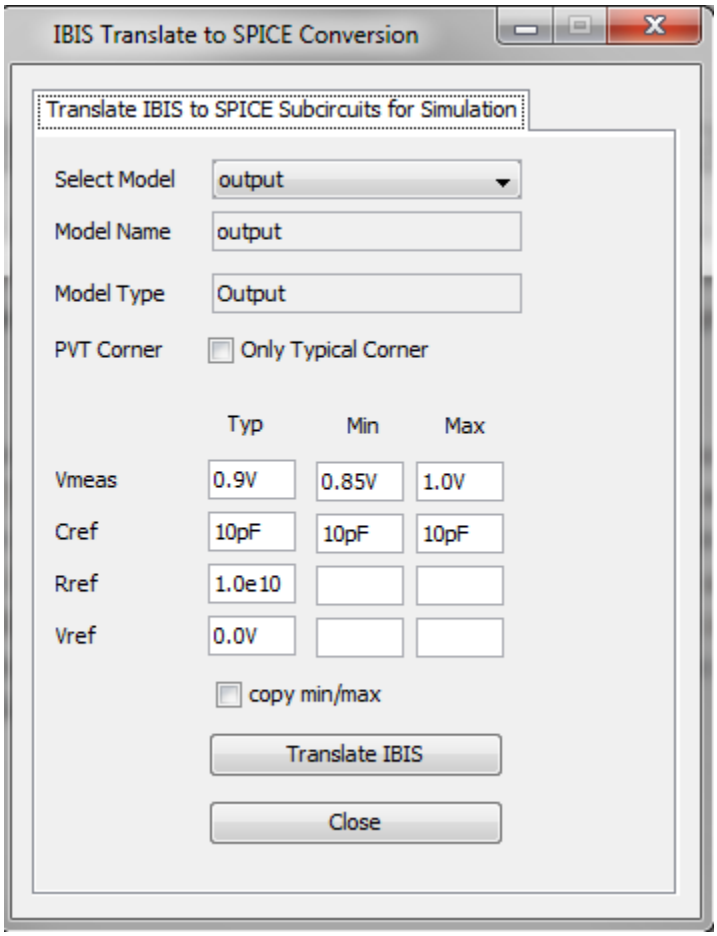
Sharksim gives the user the ability to generate an HTML report on the overall IBIS file by including any type of data like the model quality HTML report, IBIS simulation correlation data and screenshots and so on. The reporting feature can be accessed by going to **Reports | Generate Custom IBIS Report**.



The IBIS Report allows the user to save the IBIS header information and include the Notes and IBIS Parser results. Than the user can select different data to include with each model buffer in the IBIS file like the quality HTML report, screenshots, saved data text files and so on. The user selects the Generate HTML button and all of the data is compiled into one single HTML report for the whole IBIS file. This is useful to give to external customers and for internal process controls. An IBIS Quality Checklist Report can also be generated using the same method by going to **Reports | Generate IBIS Quality Checklist Report**.

Translating IBIS to SPICE Subcircuits

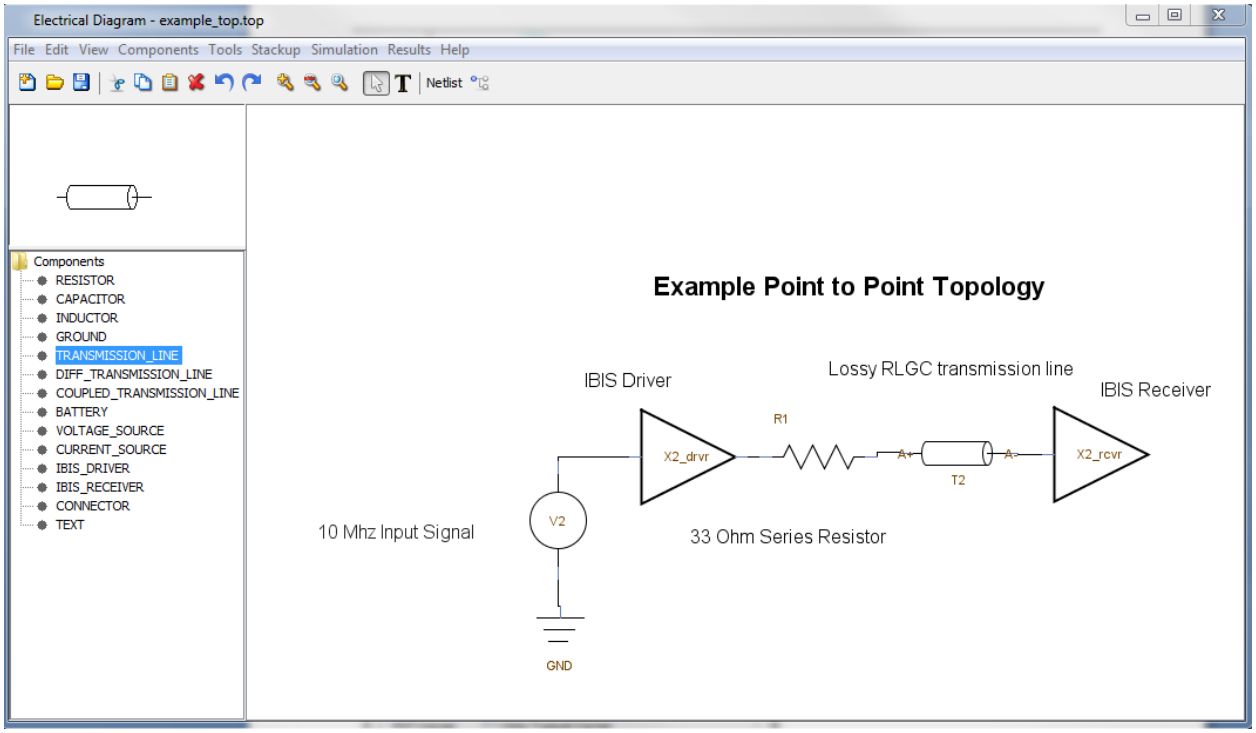
In order to use an IBIS model in SharkSim PCB the IBIS model needs to be translated to a SPICE subcircuit. This feature can be accessed by going to **Translate | Translate IBIS to SharkSim PCB Format.** For other simulators the appropriate translator option can be selected.



The IBIS timing load parameters are automatically displayed for you and will be stored with the SPICE subcircuit. You need to translate each model from the IBIS file that you want to simulate. There are options to include all three process corners and modify the timing test load. The copy min/max button will apply the same timing test load to the min/max PVT corners that the typical PVT corner has.

Schematic Topology Editor

The schematic topology editor allows you to quickly build your interface topology, assign models, and simulate and view the results. From the topology editor you can also access the transmission line editor (with built in 2D field solver) and the waveform viewer with noise and timing calculator. This feature can be accessed by going to **Simulate | Launch Schematic Entry and Simulation.**

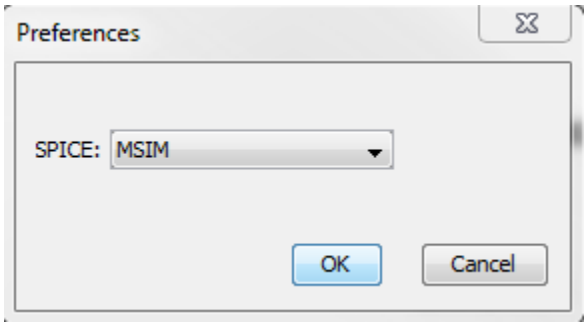


The topology editor has an easy to use drag and drop interface and will be very familiar to engineers who have used PSPICE type tools before.

Start or Open a Project

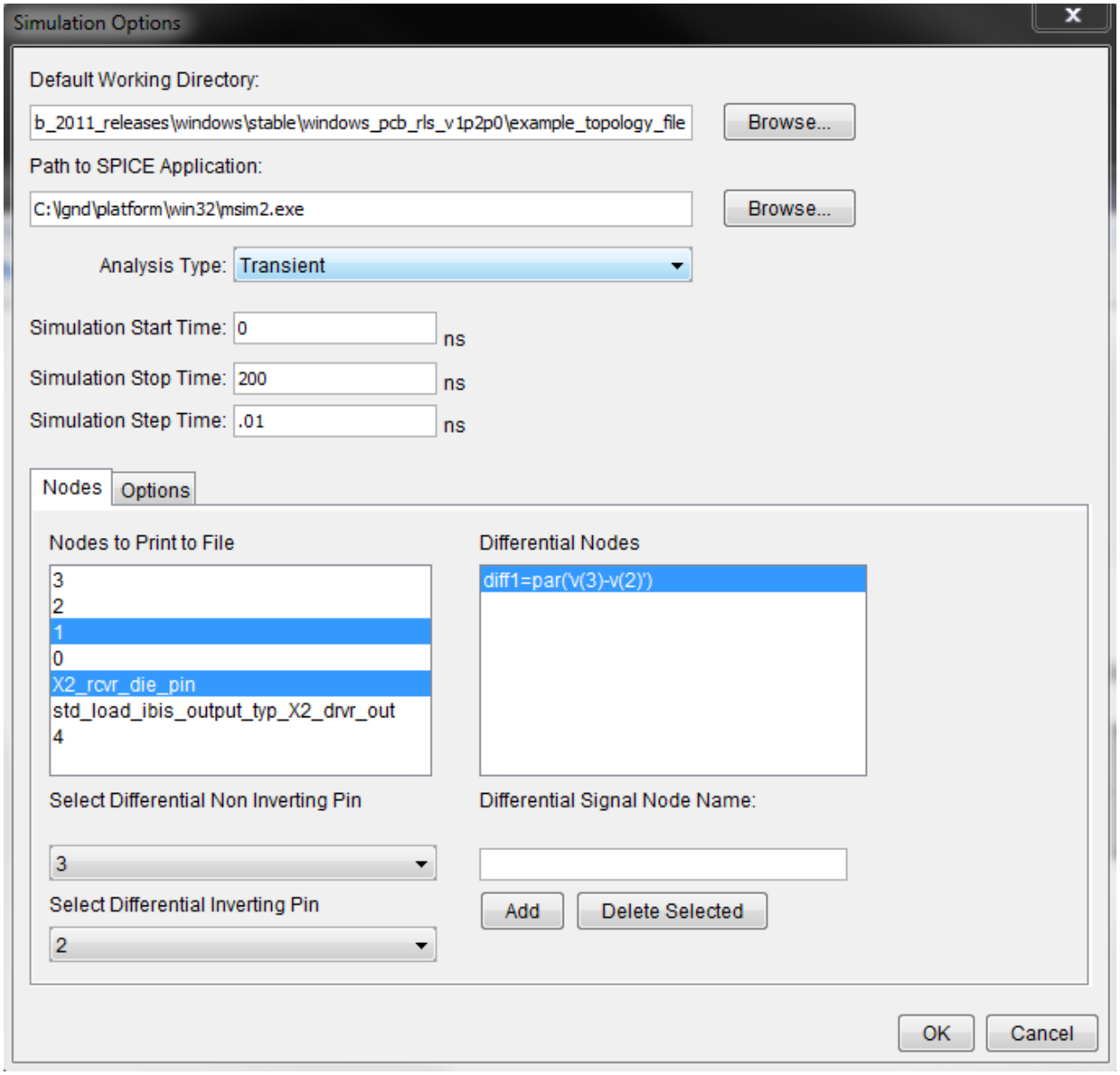
To open an existing project go to **File | Open**. To start a new project go to **File | New**. The topology tool saves the schematic files with a *.top extension.

Set Simulator Preference



SharkSim PCB Windows versions come with the embedded MSIM simulator by default. You can also select HSPICE or LTSPICE as an alternative simulator but you must have a valid license to use any other simulator. To set your simulator preferences go to **Tools | Preferences**. To use the embedded MSIM simulator select **MSIM**.

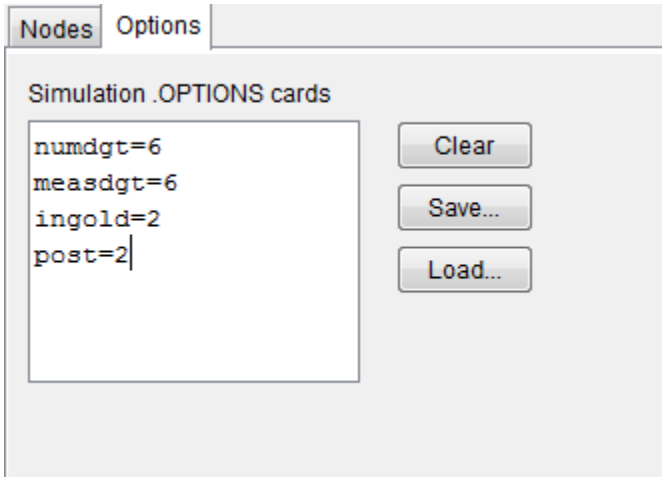
Set Simulation Options



To set your simulation options go to **Simulation | Simulation Options.** You should first select a folder on your computer that you have read and write access as your default working directory. This is where all of your simulation files will be stored. If you are using an external SPICE application such as HSPICE you will need to browse to and select the location of the exe file.

Next you can select the simulation analysis type and the settings. Under **Nodes** you can select from the available schematic which nodes to print. You can also create differential node signals

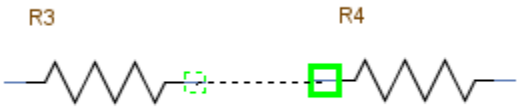
by selecting the inverting and non-inverting pins, giving it a name, and selecting **Add**. Your simulation options will be saved along with your .top file.



You can also click on the **Options** tab and enter in the simulation option commands based upon your simulator. For embedded MSIM simulator leave the defaults as shown below. If for some reason the defaults are not listed you will need to enter them for the tool to work properly. You can save and load different option settings as well.

Selecting and Connection Components

To build a simulation topology simply left click on a component from the toolbar on the left and left click on the canvas to place the component. You can keep placing components once a component is selected. To deselect a component or when you are done adding components click on the mouse cursor tool bar item.



To connect components together you need to use the wire component. Select any component so it is highlighted and then left click on the node to select it. Then right click once to start the wire connection. You can right click again to route the wire where you want it to go. When you want to connect the wire to a component just connect the component node with the wire and use a left click to finish the connection. So it's left click, right click, draw wire, right click, left click.

Setting Component Properties

Any component properties can be edited by right clicking on the component and selecting properties.

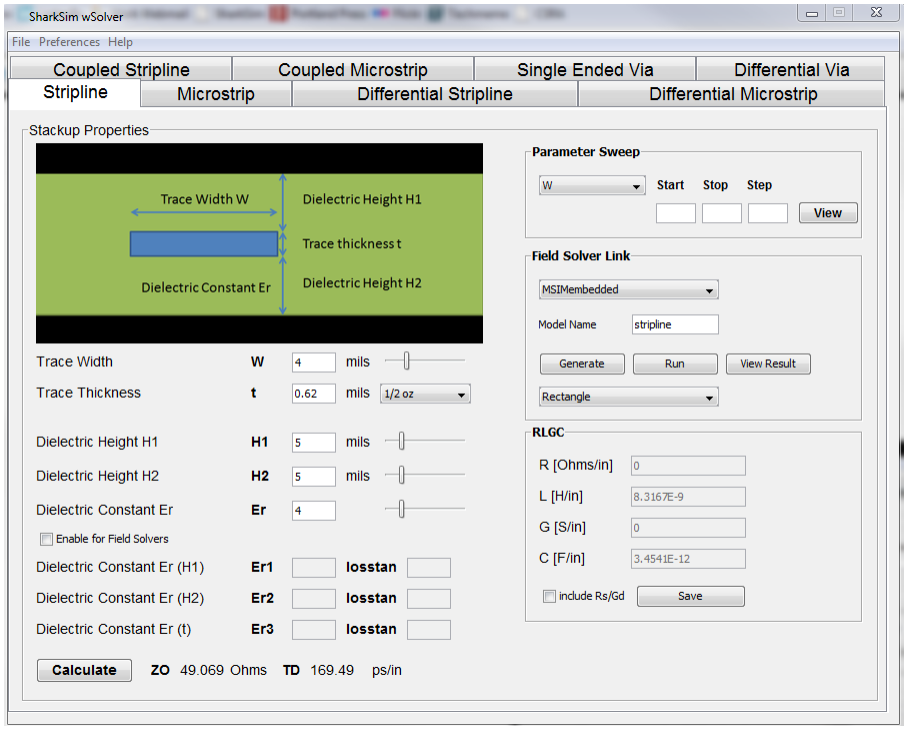
For an IBIS Driver or Receiver when you right click you will need to browse and select the translated IBIS to SPICE subcircuit generated from the main SharkSim PCB IBIS editor window.

For transmission lines the default is a lossless ideal transmission line. If you select Lossy then you will need to browse to and select an RLGC file. For single-ended transmission lines you can manually enter the lossy RLGC values as well. You can also select the PCB Stackup Editor button to launch the Transmission Line Editor tool with a field solver.

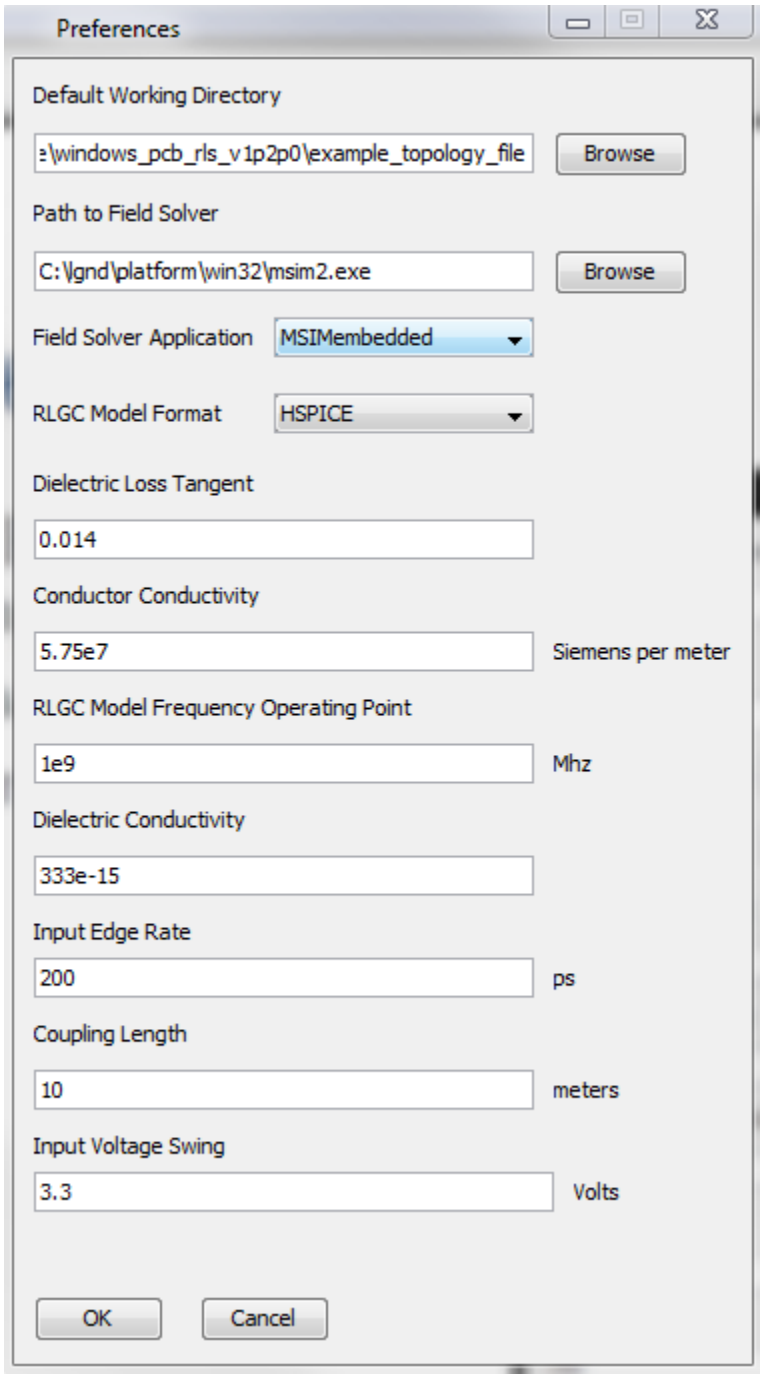
It is required that you have at least one ground component in your schematic. If you don't the tool will give you an error.

Using the Transmission Line Editor

You can launch the Transmission Line Editor from any of the transmission line components using the button or from the menu from **Stackup | Transmission Line Editor**.



The Transmission Line Editor tool includes the eight most common PCB structures. It allows for quick stackup analysis by using analytical equations to calculate and graph stack parameters against impedance. You can also use the embedded MSIM simulator or external field solver to generate RLGC models to use in SharkSim PCB.



The first thing to do is to go to **Edit | Preferences** to set the field solver preferences. The working directory is where all of the files will be stored. If you are using an external field solver you will need to point to the exe path location. For the field solver application leave the MSIMembedded as the default if you are using the embedded MSIM simulator. Otherwise you can choose HSPICE or the open source TNT field solver application.

The rest of the preferences fields are for the field solver and calculations. Some of these values will come from the dielectric material of your PCB such as FR4.

Dielectric Loss Tangent: Loss tangent of dielectric material (usually between 0.014 and 0.022 for FR4)

Conductor Conductivity: Conductivity of metal used for trace such as Copper (standard value for Copper is 5.75e7)

RLGC Model Frequency Operating Point: Define frequency of simulation model (default should be 1 Ghz)

Dielectric Conductivity: Conductivity of dielectric material (standard value for FR4 is 333e-15)

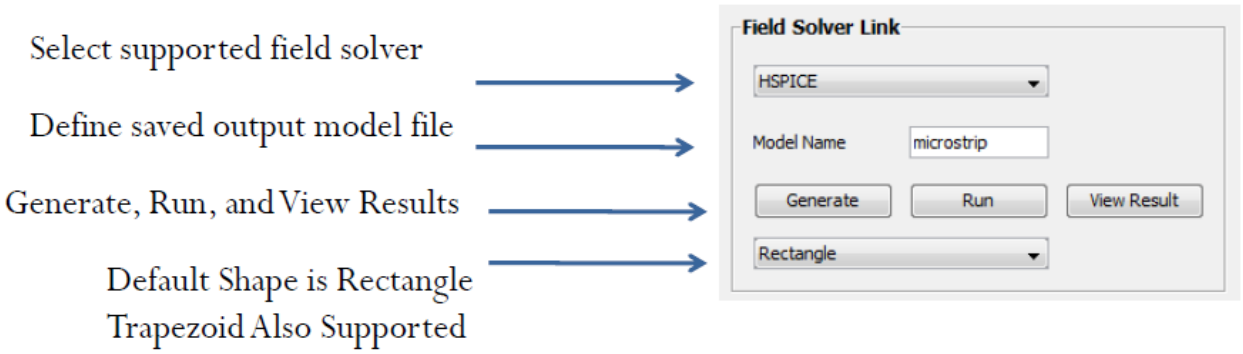
The following values are used for crosstalk calculations:

Input Edge Rate: Edge rate of input signal

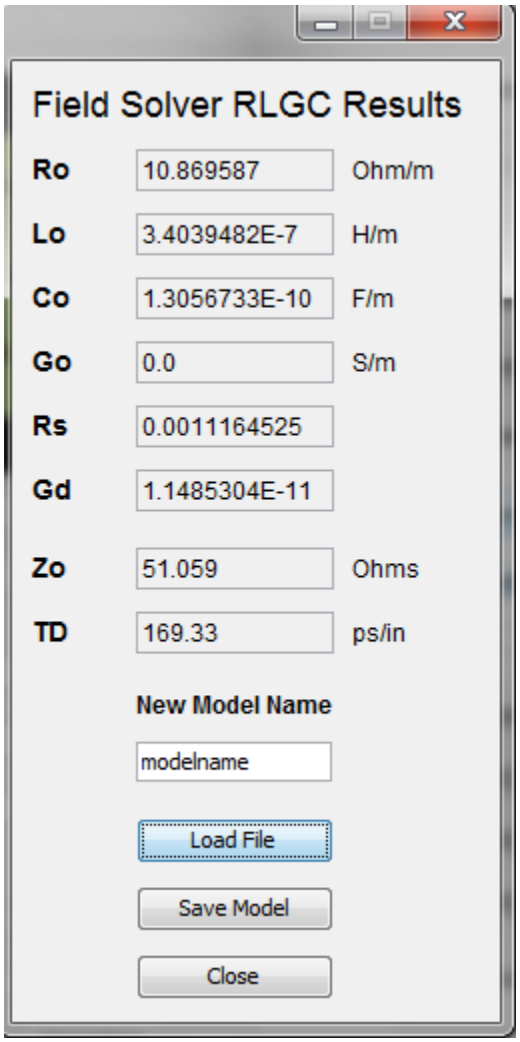
Coupling Length: The parallel coupled length of two traces in PCB layout

Input Voltage Swing: Voltage amplitude of input signal

To use the field solver option to generate an RLGC file of the PCB structure simply select the supported field solver, enter a model name, and generate and run the file. Then you can view the resulting RLGC file in the data viewer.



The RLGC data viewer reads the RLGC matrix and automatically calculates the single-ended impedance and delay. For differential and coupled lines it will also calculate differential impedance and even and odd modes.



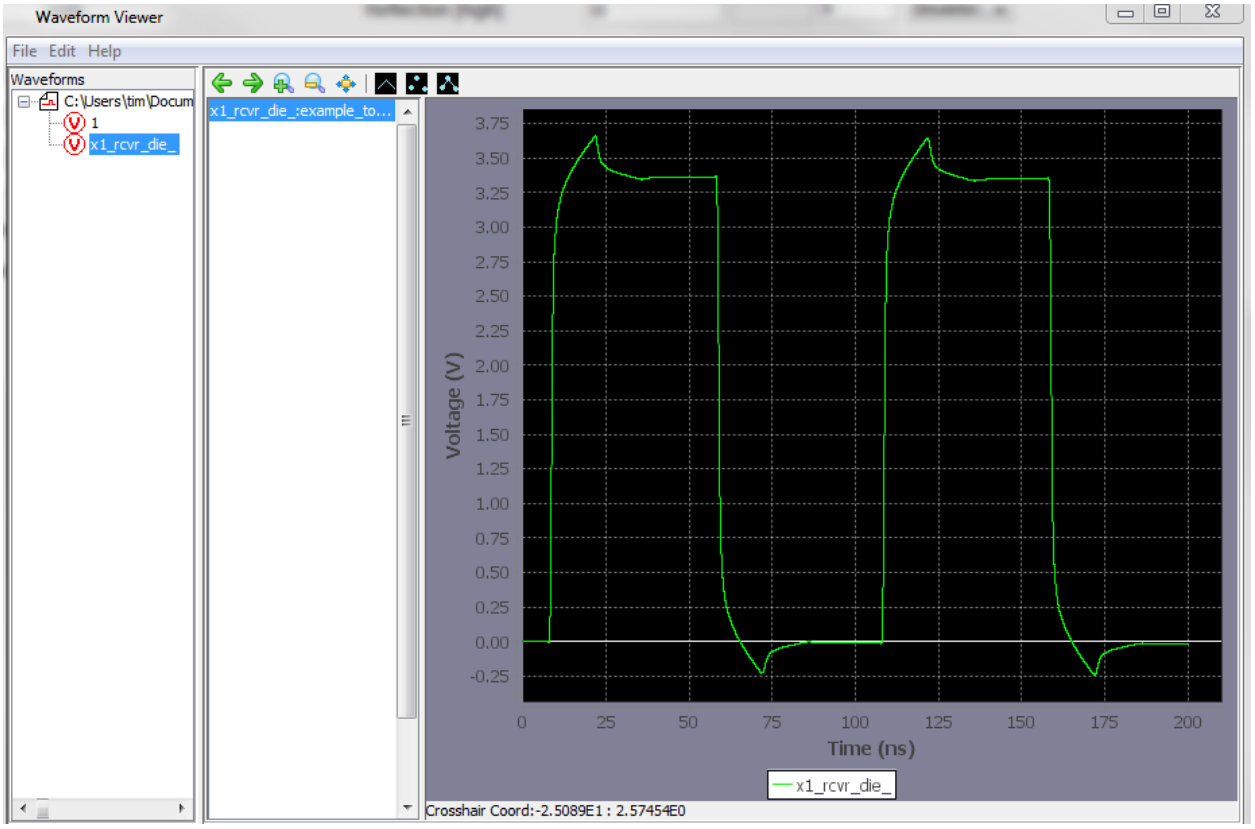
Running a Simulation

Once your preferences have been set and models assigned to your components in your connected topology you can run your simulation. To do this go to **Simulation | Run**. This will automatically generate a SPICE netlist and run your selected simulator with this netlist. You can also use the Netlist button to save a netlist and modify it or save multiple netlists. You can then use the **Simulation | Run Batch Mode** to select one or multiple netlists to run at once.

Viewing Results

To view your simulation results you can launch the waveform viewer and calculator tool. To do this go to **Results | Waveform Calculator and Viewer**. The main screen is the noise and timing calculators. To open the waveform viewer go to **Waveform | Launch Waveform Viewer**.

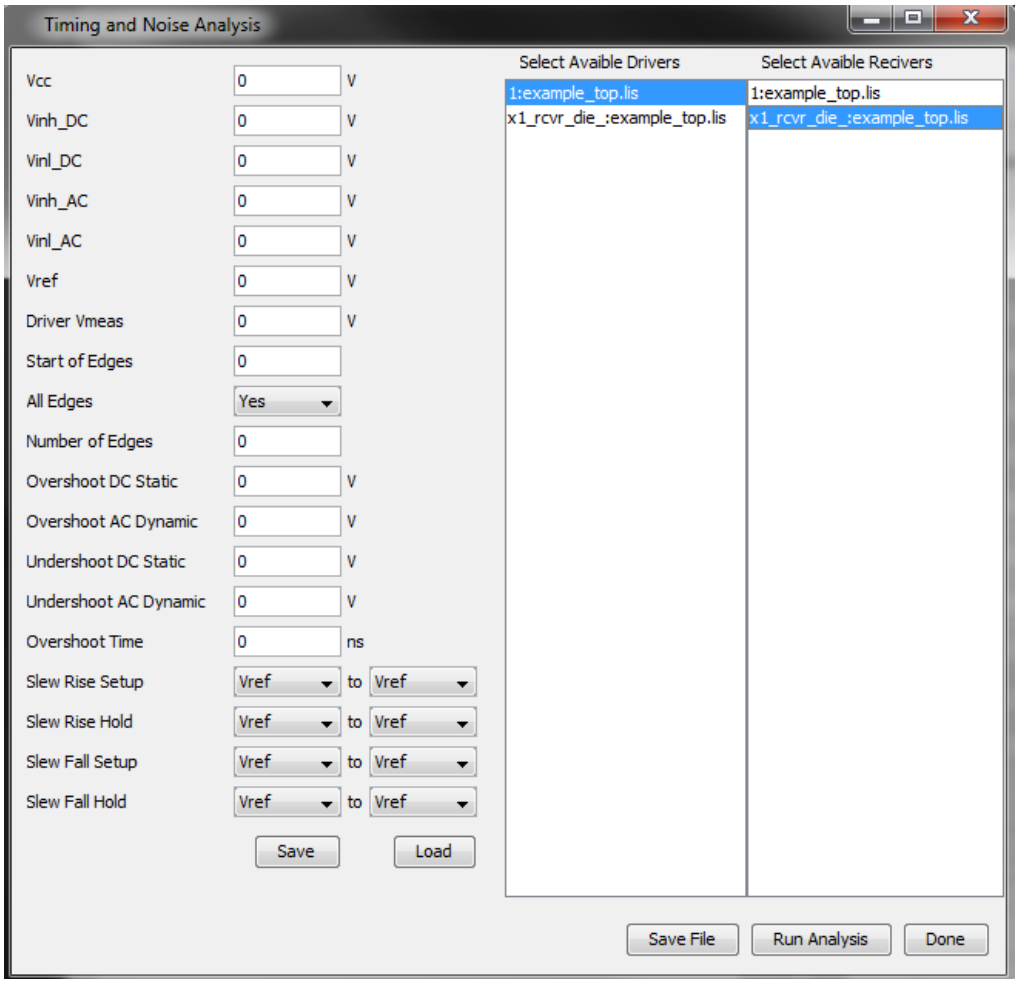
To open a waveform file go to **File | Load File** and open the *.lis simulated file. You can then plot the selected nodes you printed by dragging and dropping them in the waveform viewer.



Noise and Timing Analysis

The main screen allows you to create and save a noise budget for your interface. You can also create a timing budget for a common clock interface. Once you simulate and analyze your results you can update these budgets to reflect your actual system. Then you can save the budgets as HTML table files for your reports.

You can also run an analysis on the selected waveforms to automatically calculate the noise and timing margin analysis for you and create an Excel spreadsheet of the results. (Note: This feature may not work on all Linux systems.) To do this go to **Analysis | Run Timing and Noise Analysis**. Your printed nodes will automatically be loaded for you. **However you always have to first load the file in the waveform viewer**.



You first need to select which nodes are the driver and which nodes are the receiver. In general you will want to select the Driver standard load node and any receiver node for the flight time calculations to come out properly.

Next you need to enter the various AC and DC threshold settings for your noise and timing analysis.

When you are done select **Save File** and you will be asked to save a batch file with an * .wcb extension. This allows you to create different analysis templates for the same set of signals.

Finally you can select **Run Analysis** and you will be prompted to select one or multiple *.wcb files. The analysis will run and create a separate Excel spreadsheet for each driver/receiver pair. You can then go to your working directory and view the Excel spreadsheets.