MBP 2017

Utilities

KEYSIGHT TECHNOLOGIES

Notices

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause.

Use, duplication or disclosure of Software is subject to Keysight Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Portions of this software are licensed by third parties including open source terms and conditions. For detail information on third party licenses, see Notice**.**

Contents

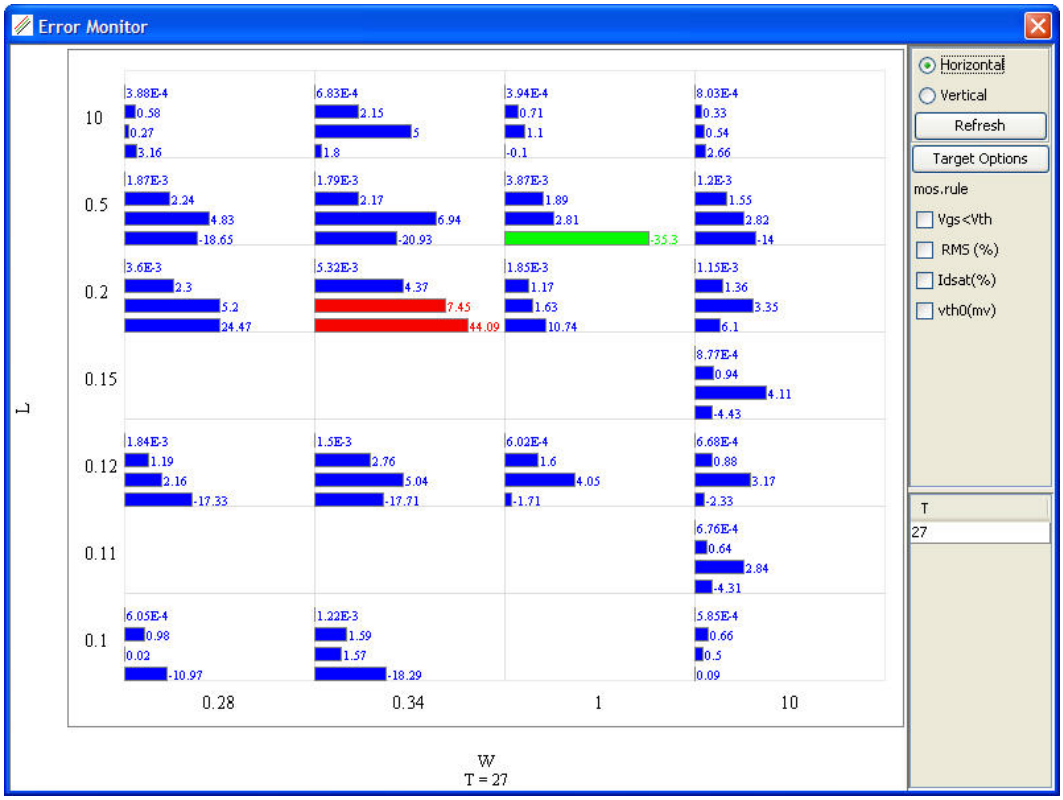
- Error Monitor
- Equation Viewer
- Export Graphic
- Export Sim
- Export to MosDiode Data
- Export to Spectre
- Export Virtual Data
- IMV Pages and DP Implementation
- Lib Parser
- Subckt Template
- MBP Reviewer
- DB Data Loader

Error Monitor

You can have different standard on accepting released model. Most of these standards are made from RMS (Root Mean Square error). MBP enables you to monitor RMS of any devices at specific target in a more intuitive and flexible way by Utilities > Error Monitor.

Error Monitor is also data based module. Use Device Navigator to choose the devices of interest. As shown in following figure, the main window is divided into several panes. Each pane represents a device. Devices are in a descending order of width from the right to the left and a descending order of length from the top to the bottom.

Error Monitor window



In each pane, there are several bars and each followed by a figure. Each bar corresponds to the target that you set. All bars arrange according to legends on the right of the window. For example, from the top to the bottom, the fourth bar stands for target of Idsat. Figure shows its RMS value. The target is defined in MBP rule files. The "rule.index" under `<MBP HOME>\JEF\usr\target` keeps links to the paths of rule files, shown as below:

```
[BSIM3V3]     x=w     y=l
file=mos.rule
......
We cite "mos.rule" shipped with MBP as an example to
explain rule format.
[Title]     value:vth(mv)
[Definition]     value:Vth@vbs=0
[Target_type]     value:vth
[Enabled]     value:yes
[Warning_criteria]     value:0.03
[IsAbsolute]     value:yes
[Scale]     value:1000
[Range]     index:1     page:Ids_Vgs_Vbs|Vds=Vdlin
pvalue:0
......
```

Each rule contains fields of Title, Definition, Target_type, Enabled, Warning_criteria, IsAbsolute, Scale, and Range .

- In Title field, specify the target name. Avoid using the same title for different target.
- The Definition is to help to clarify target definition. It shows one pointing to the corresponding rule, as shown below.

Sample comment

- Target_type should be set as one of the following:
- A variable like Vth or Idsat
- RMS
- MAXRMS
- Yes for Enabled is to adopt this rule. And No for Enable is to close this rule.
- The bar in the Error Monitor turns red if it exceeds Warning_criteria value. Or it remains blue. If a variable is set as Target_type, the Warning_criteria could be either absolute value or relative value, depending on how IsAbsolute is set.

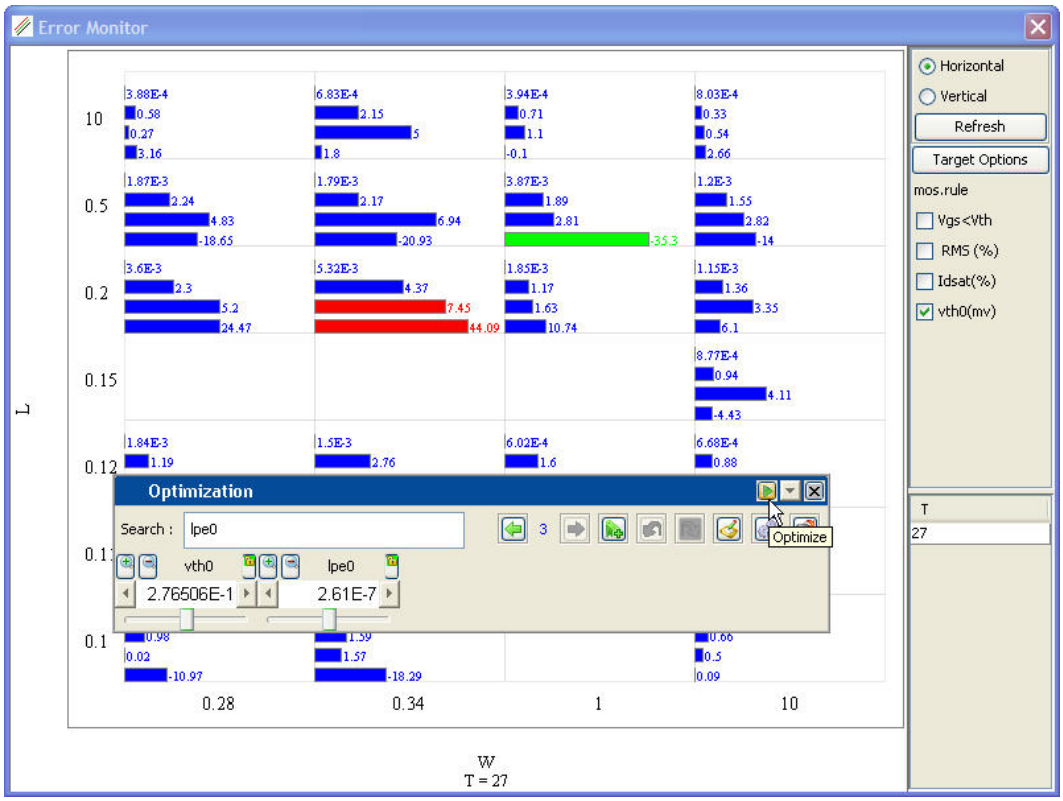
- IsAbsolute only takes effects with a variable set as Target_type. Yes for absolute value, and No for relative value
- Scale takes effect with error plot, for example, when scale = 1000 for vth, it takes mV as Vth unit. However, it does not affect Warning_criteria value.
- Range also contains several functions as follows:
    - Index: An integer value. More than one index for a rule means that it is applied under different conditions.
    - Page: To select one of available device pages.
    - Pindex: An integer value. To identify which curve to select. Use `,` to separate different curves, like `0,1,4`. All curves are selected by default.
    - Pvalue: Also to specify which curve to select. However, it can use built-in variable, like `0,VBB`. All curves are selected by default.
    - Mathfunction: Choose one of these math functions: `y`, `1/y`, `y' , y"`, `dx/dy`. Mathfunction defaults to `y`.
    - XRegion: To specify x-axis region on selected page. By default, it is (-inf, inf). You can also use other variables. For example: (vth, vth+0.2)
    - YRegion: To specify y-axis region on selected page. By default, it is (-inf, inf). Another example: (-inf, 0.8*ymax)

The following lists out variables that you can refer in the rule, as partially seen from above:

- VGG: max VGS
- VDD: max Vds
- VBB: max Vbs
- VDLIN: minor Vds used in Ids_Vgs_Vbs page
- Vth: subthreshold voltage
- Idsat: saturation drain current
- Xmin: the min x value
- Xmax: the max x value
- Ymin: the min y value
- Ymax: the max y value
- -inf: the negative infinite value
- inf: the positive infinite value.

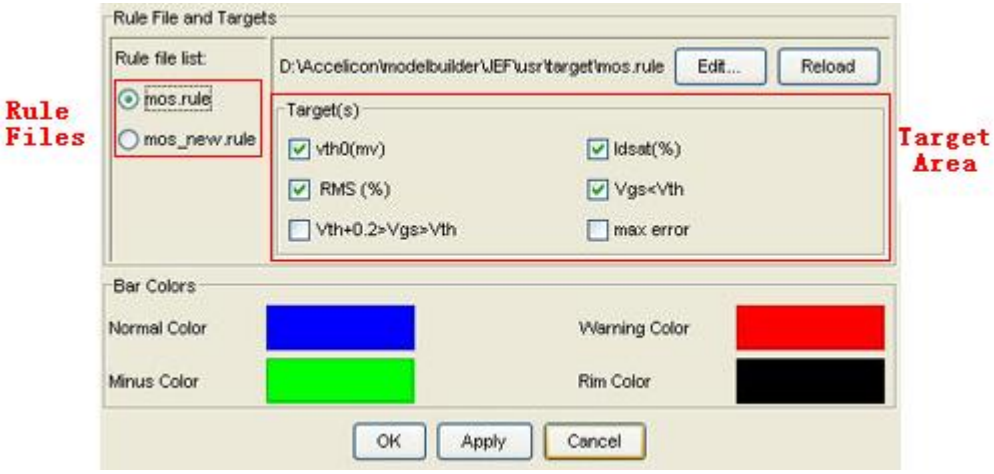
Moreover, in Error Monitor, you can open the Optimization window directly to tune the model. All bars respond immediately upon adjustments. You can even check targets on the right of the Error Monitor, and click Optimize button to perform auto-optimization by our optimizer. Surely, parameters should be manually selected before it, as shown in following figure. It helps to acknowledge overall RMS change by tune.

Optimization with the Error Table



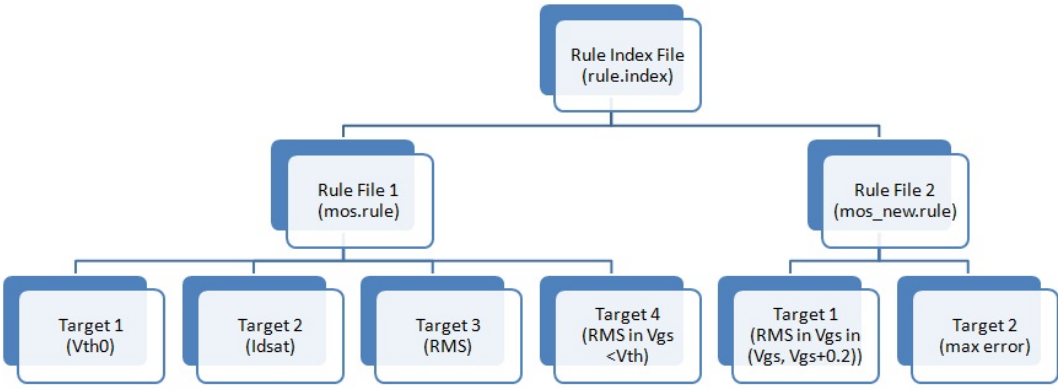
To make it easier to do the customization of the error monitor, MBP now includes the target define interface in. Open it using the button at the top-right of the error monitor.

Target Define Dialog



The hierarchy of the target file is shown as the following pictures. All the rule files are indexed using a rule.index file. And different target can be defined in same rule file. These different targets and rules are linked with different areas in the dialog above.(Shown in the red).

The Hierarchy of the rules and targets



Equation Viewer

This section describes the features, functions and operations of equation viewer, a useful debugging tool of MBP.

Overview

Equation viewer is a unique feature that we offer to help debug model issues and hence improve model quality. As the feature size of modern device continues to shrink, many effects that were once ignored are now dominating upon deep submicrometer transistors. These effects, dictated by device physics, require new device-modeling techniques to predict the resulting circuit behavior. It makes device model quite complex, with intricate equations and quantities of parameters. As a consequence, modeling engineer is facing increasing challenge in terms of extracting accurate models, and always feels difficult to trace the cause for unreasonable model. However, you may find it easier and convenient for model analysis by using the Equation Viewer with its useful features:

- Print out original equations in terms of model type, supports all supported model type in MBP, including BSIM3v3, BSIM4, BSIMPD, PSP, GP, and diode model.
- Print out calculated results of all equations for selected data point.
- Show numeric difference of various data points either from one or different plots.
- Dynamically show numeric difference of various model status, that of the same model type but with different parameter values.
- Output intermediate variables, operation points, and equivalent circuit, which may be of interest to IC designers as well.
- Capable of analyzing macro models. Provide connections to equations of each element.
- Output equations in graph.

All these features have already been proved to be effective in figuring out model problems. Detailed functions, and operations are discussed in the later sections.

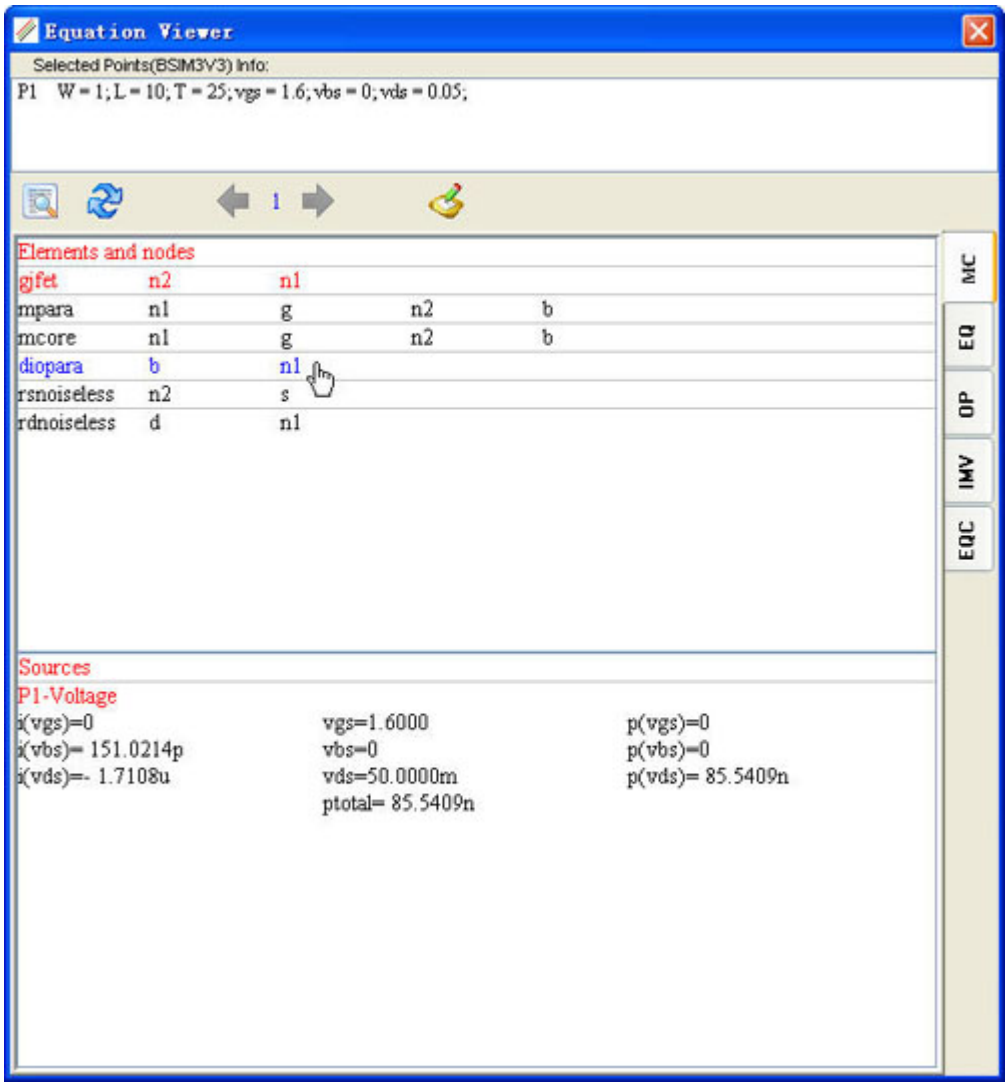
Equation Viewer Interface

Main View of the Equation Viewer

MBP pops up two different windows for compact models and macro models.

Equation Viewer (EQV) for Compact Models

Main view of EQV for compact models



As illustrated in above figure, the Equation Viewer window includes the following major elements:

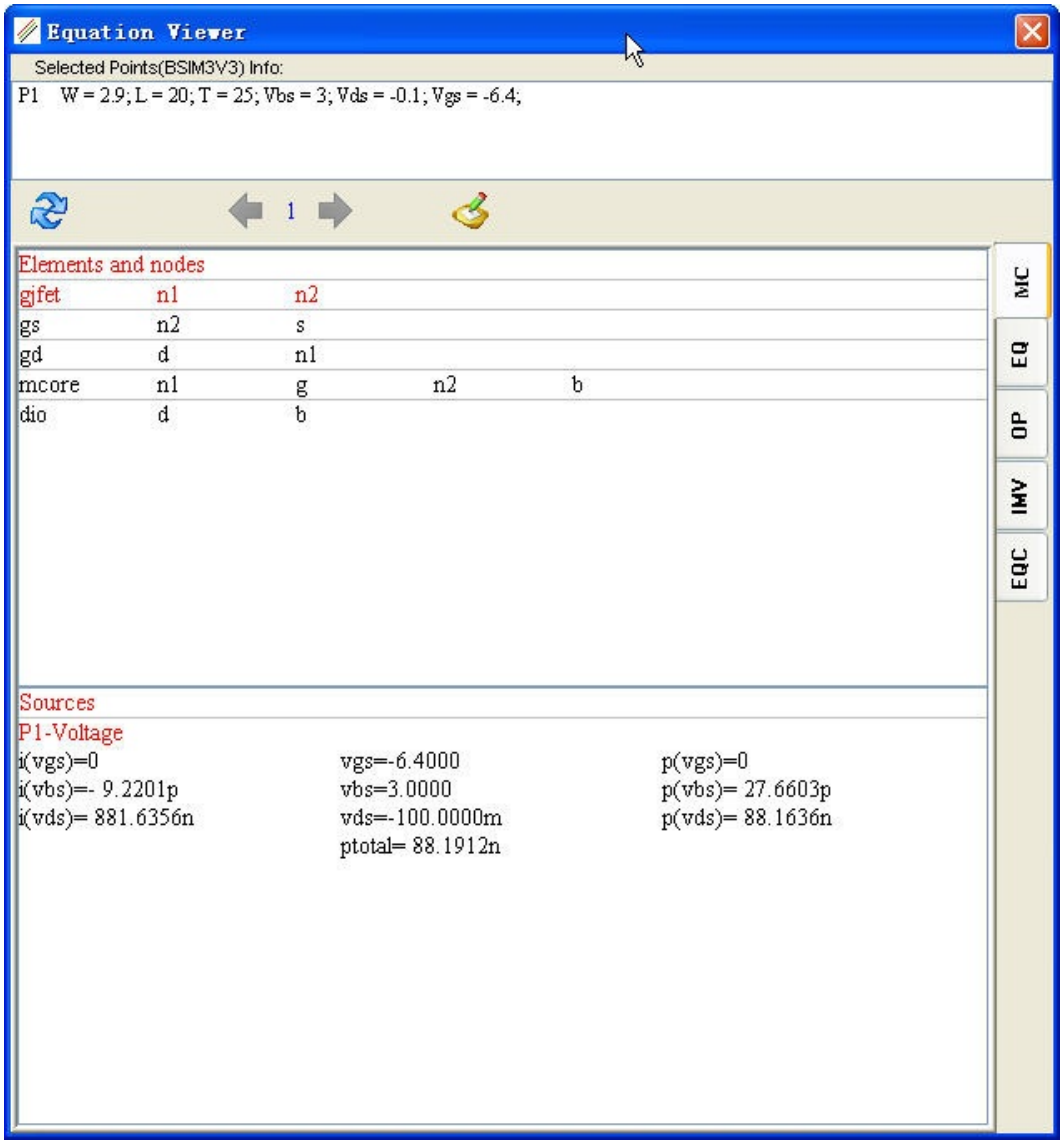
- Point Info Window displays the information of the selected data points. For example, it includes point index (P1), instant parameters (W=1 (um), L=10 (um), T=25(C)) and bias conditions (Vgs=1.1 (V), Vbs=0 (V), Vds=0.05 (V)).
- EV Tool Bar provides the utilities to refresh and save equations as graph. Details are presented in this chapter later.
- You can switch among EQ (Equations), OP (Operating Point), IMV (InterMediate Variable), and EQC (Equivalent Circuit) through Viewer Tab .
- Content Window outputs equations, operating points, intermediate variables, and equivalent circuit depending on EQ, OP, IMV, and EQC selected.

EQV for Macro Models

In the main view of EQV for macro models, MBP analyzes the subcircuit first and print operating points of entire circuit and all elements with node connections, as shown in following figure. One more tag of "MC" (Macro Models) is added to the Viewer Tab. Click elements in the window links to the same view for compact models.

NOTE For the elements like current source and voltage source, only OP is available.

Main view of EQV for macro models

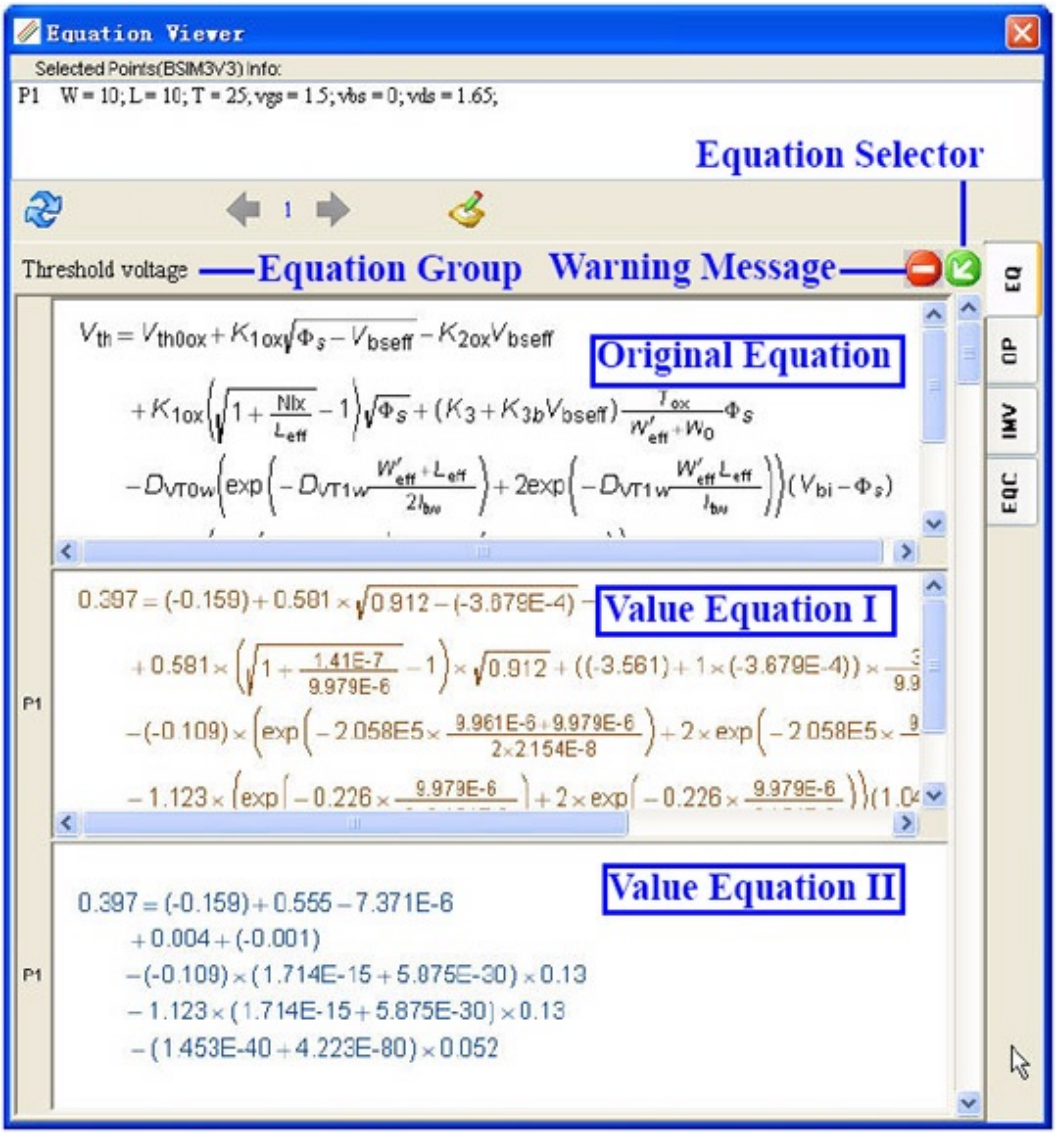


Content Window for EQ (EQuation)

As illustrated in Figure: Content window for EQ, the content window for EQ consists of the following major elements:

In Value Equation I , the parameters and variables are replaced with their actual values. It is easy to figure out that Vth for this point is 0.397, Vth0ox = -0.159, K3= -3.561 (all the model equations use effective values here).

Content window for EQ

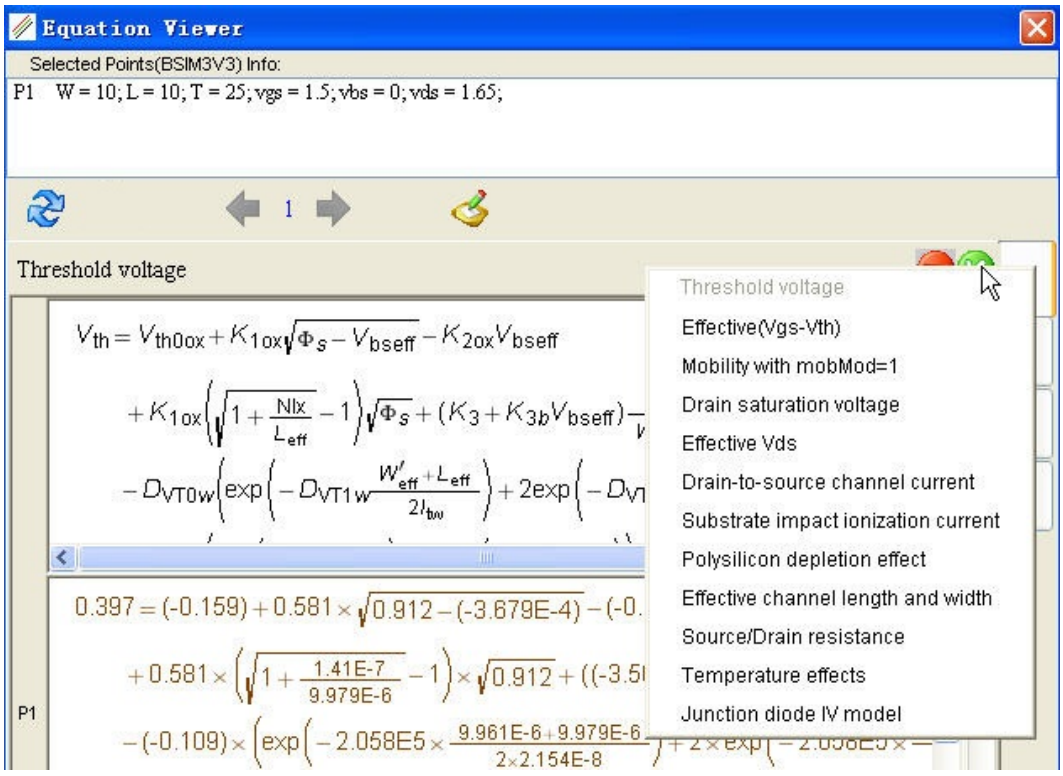


The Value Equation II outputs further calculated equations. For example, you can see the result of following equation is displayed in Value Equation II.

$$K1ox\sqrt{\Phi s - Vbseff} = 0.555$$

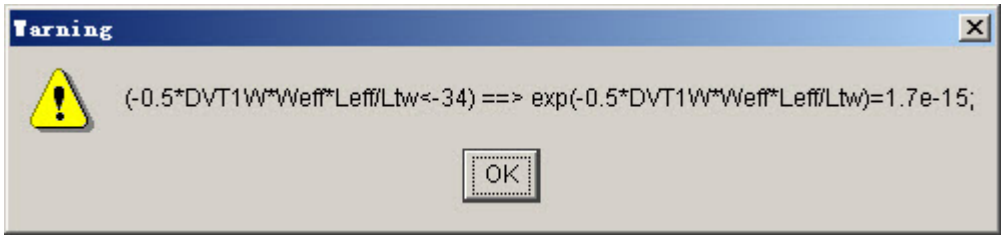
In Equation Viewer, equations classification is based on groups. You can use Equation Selector to select Equation Group, as shown in following figure. And scroll down to locate the equation to diagnose in the content window.

Equation Selector



In general, model equations characterize exact physical effects. But under certain situations, it is not the case. The simulator modifies the equation a little or adopt other method for calculations. This also happens when simulator tries to keep continuity or maintain convergence. If such case happens, the button of Warning Message brightens. You can click it to pop up detailed message. As shown in following figure, since the calculating result of (-0.5*DVT1W*Weff*Leff/Ltw) is less than -34, simulator uses a small value (1.7e-15) to as its exponent value.

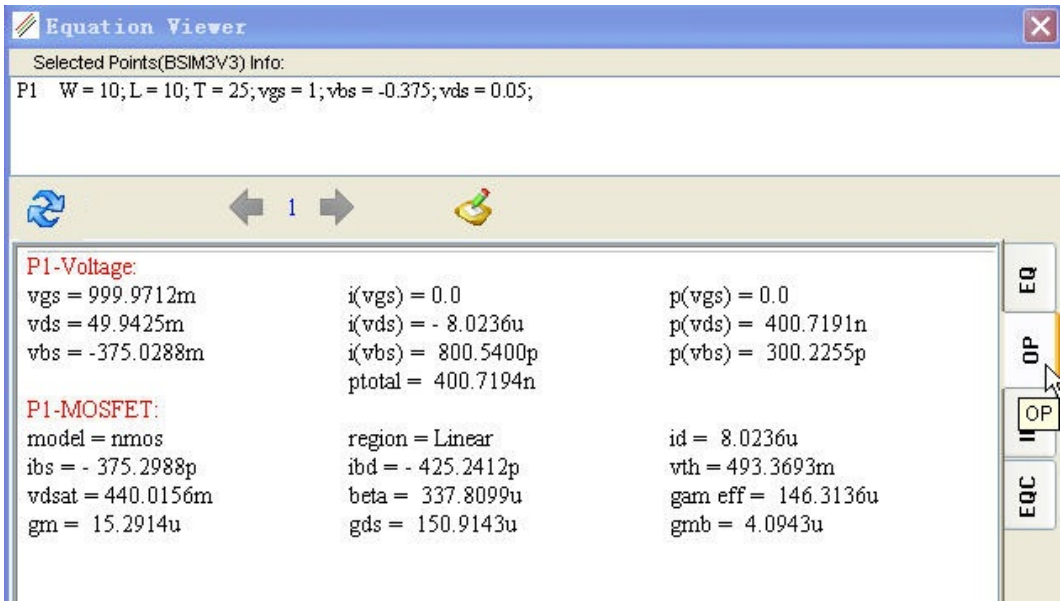
Example of warning message in the Equation Viewer



Content Window for OP (Operating Point)

By clicking OP in viewer tab, the content window displays the operating point information of selected data points, as shown in following figure.

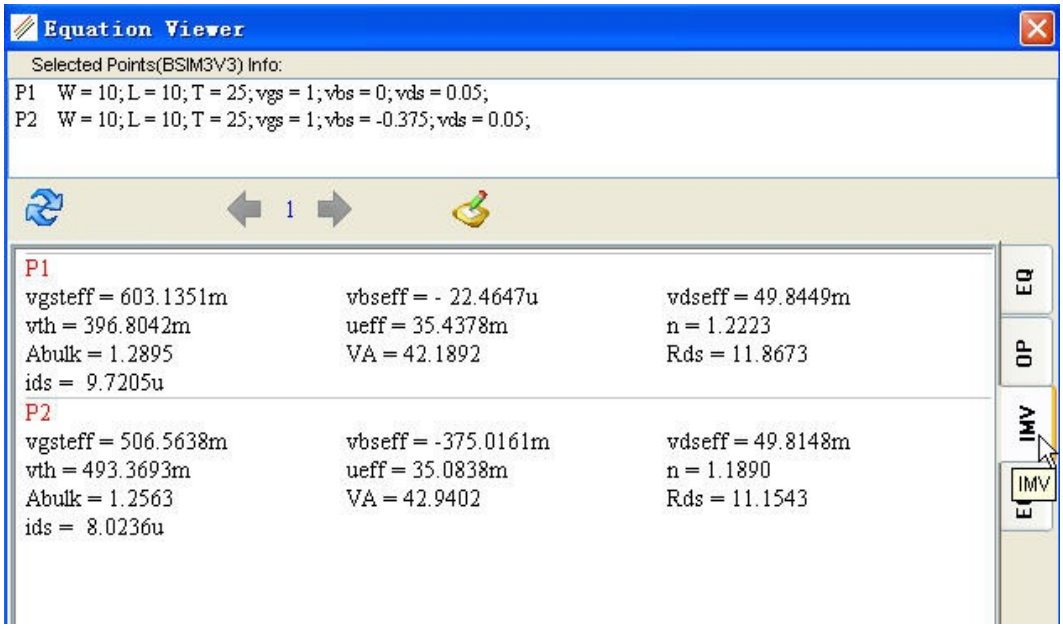
Sample Operation Point in Equation Viewer



Content Window for IMV (InterMediate Variables)

The value of intermediate variables such as Vgsteff, Vdseff, and Abulk may be more valuable for debugging model issues. Click IMV tag in viewer tab to display all intermediate variables with values, as shown in following figure. Moreover, each intermediate variable has a link to the corresponding expression in the EQ window.

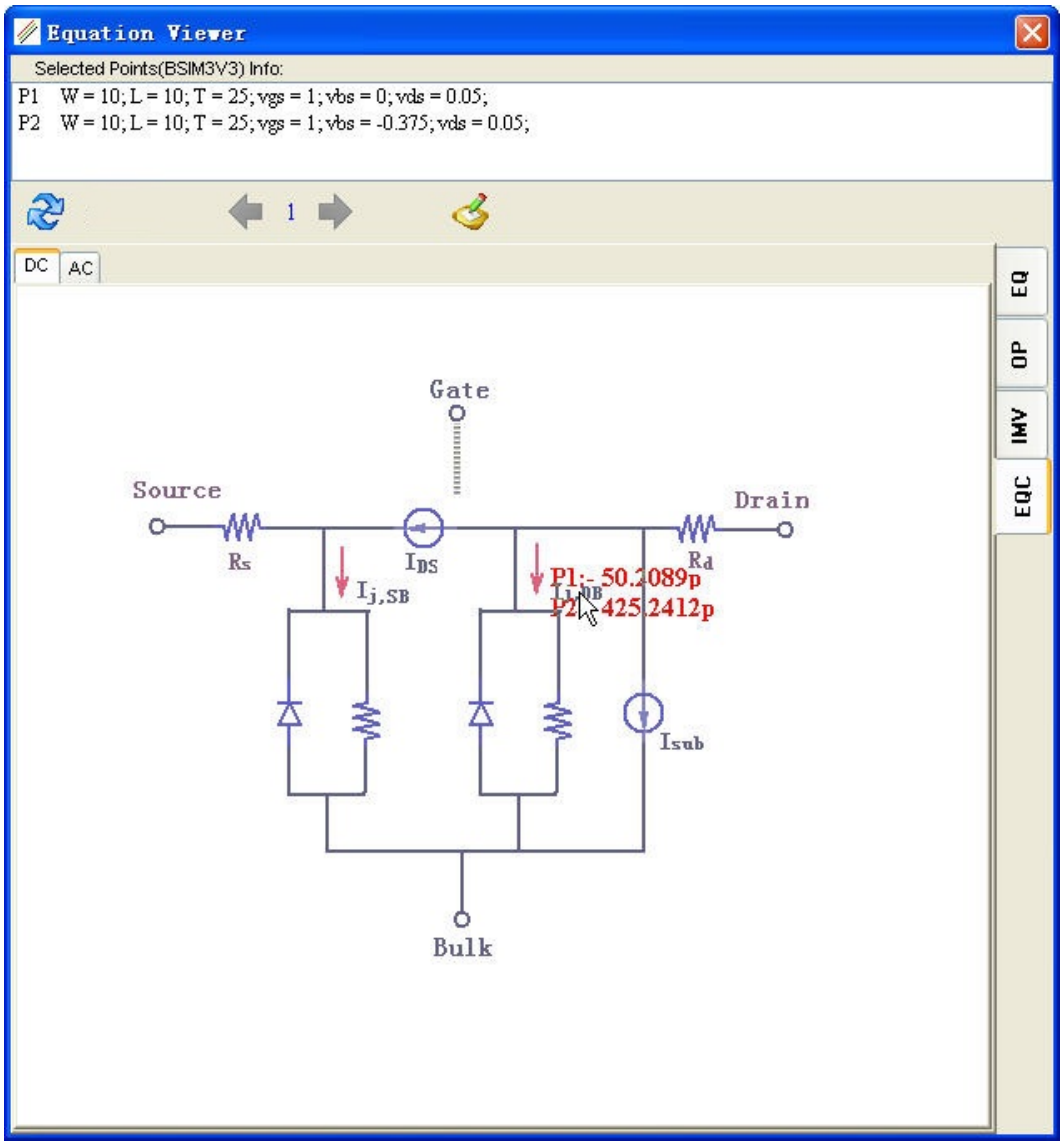
Sample intermediate variables in Equation Viewer



Content Window for EQC (EQuivalent Circuit)

Equivalent circuit of current model gets displayed in the content window by clicking EQC in the Viewer Tab. Another convenient feature is that you can read operating values by moving cursor close to given items. As show in following figure, "Ij, DB" represents the current flow through drain to bulk diode. When pointing to it, 50.2089 pA and 425.2412 pA values get displayed for data points P1 and P2, respectively.

Sample equivalent circuit in Equation Viewer



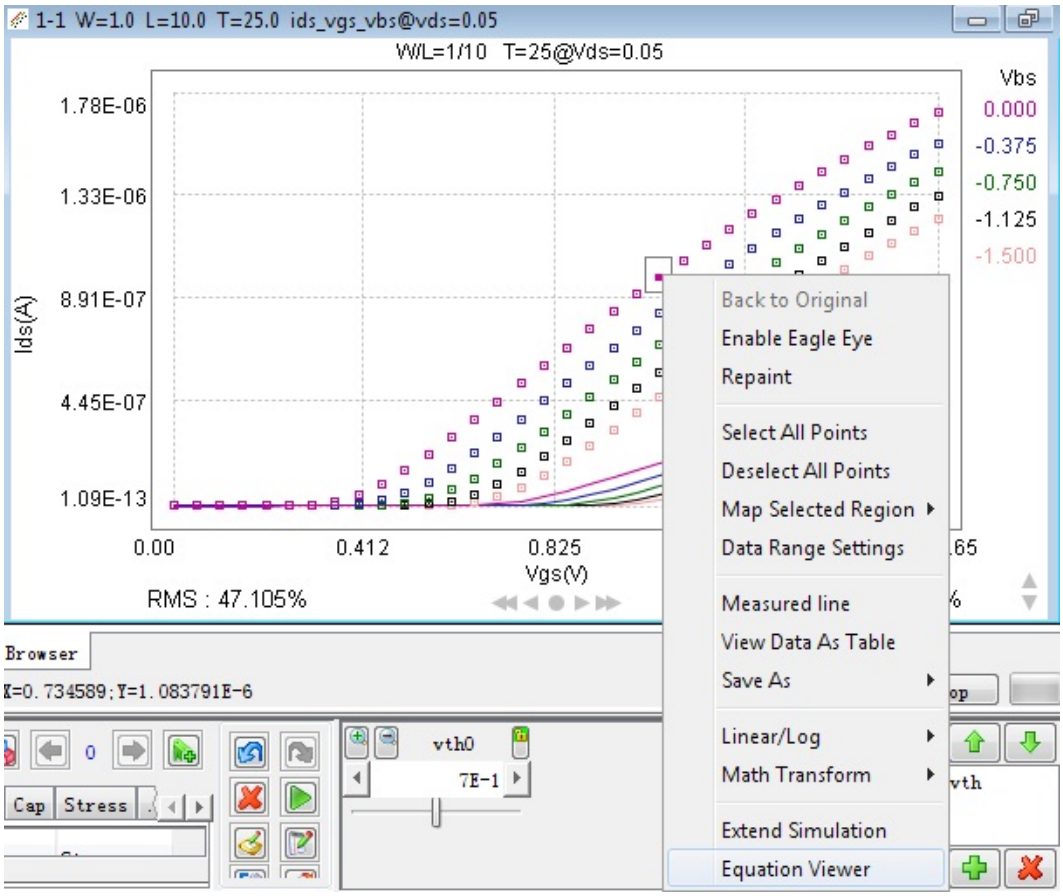
Function and Operation of Equation Viewer

The section explains how to operate Equation Viewer. Equation Viewer usage involves two steps: selecting data points and popping up the Equation Viewer.

Diagnose Single Data Point

Hold Ctrl key and use left button to draw a rectangle to cover one data point. Then, right-click on the plot and choose Equation Viewer (as shown in following figure), the Equation Viewer pops up.

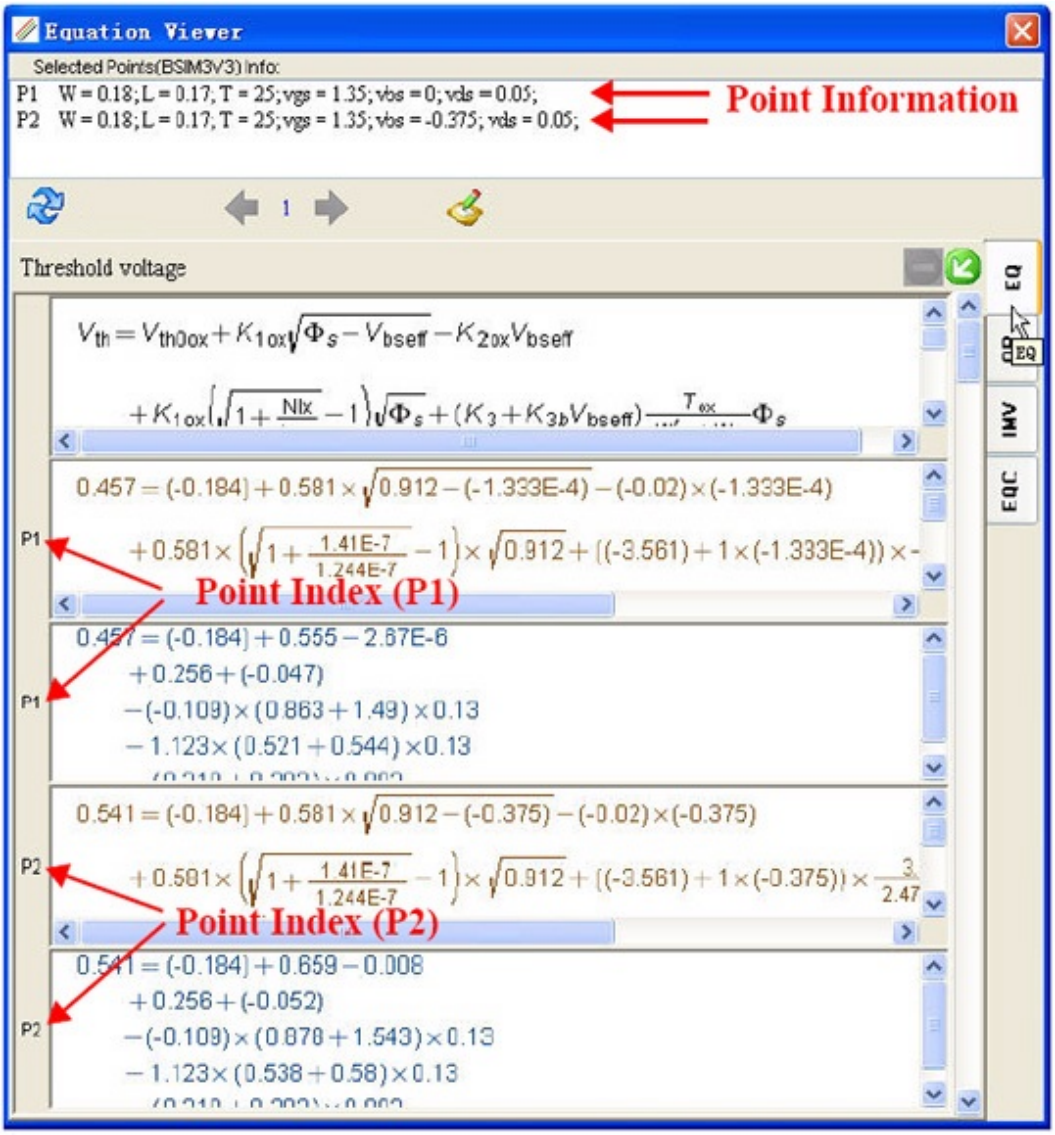
Select single point for Equation Viewer



Diagnose Multipoints from One Plot

Choose Optimization > Data Selection > Multiple from the main menu or click the Multiple Region icon  in the tool bar to activate multiple selections. Then, it allows to select multiple data points from one plot. Right-click on the plot and choose Equation Viewer, the Equation Viewer window pops up as shown in following figure. These points and corresponding equations are differentiated by Point Index.

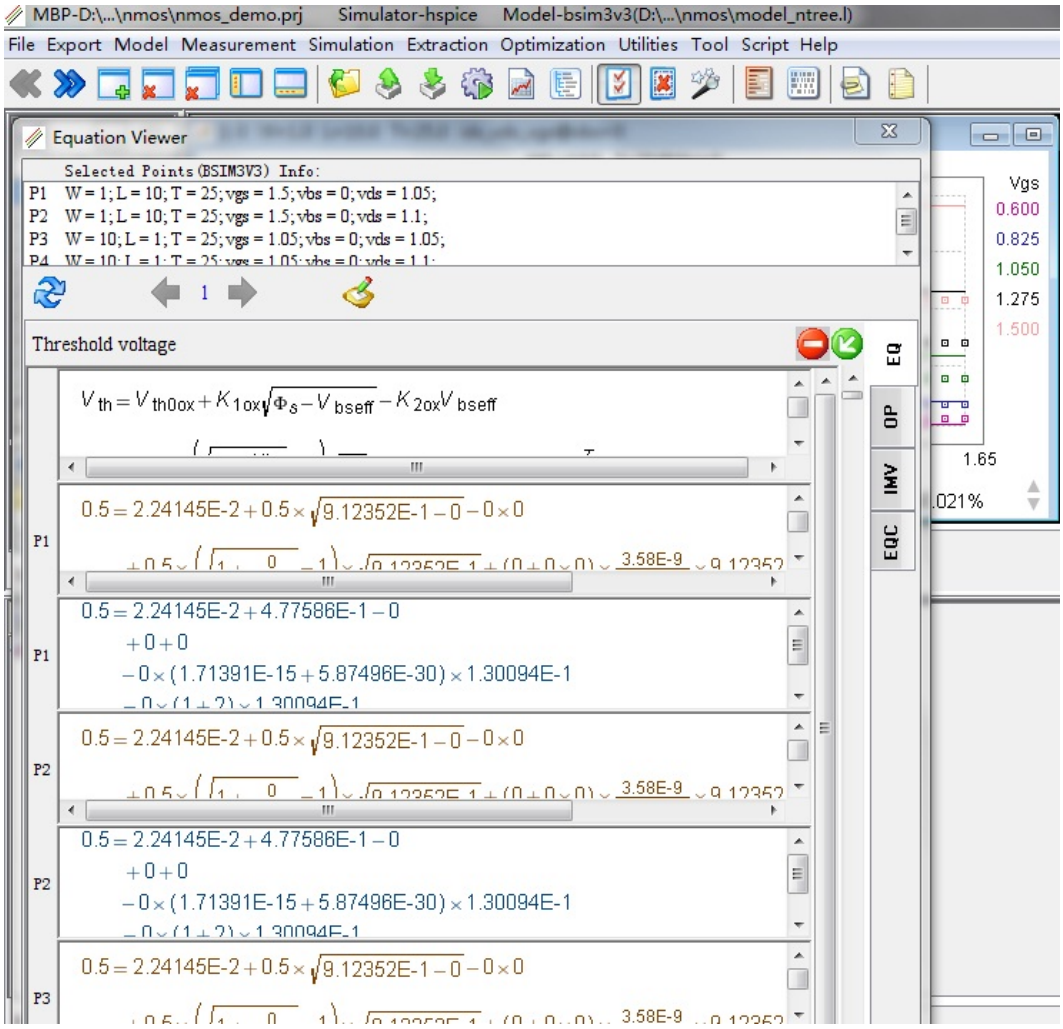
Differentiate multipoints in Equation Viewer



Diagnose Multipoints from Different Plots

MBP also allows comparison of various data points from different plots. These points and their equations are differentiated by point index, as shown in following figure.

Equation Viewer of multipoints from multi-plots

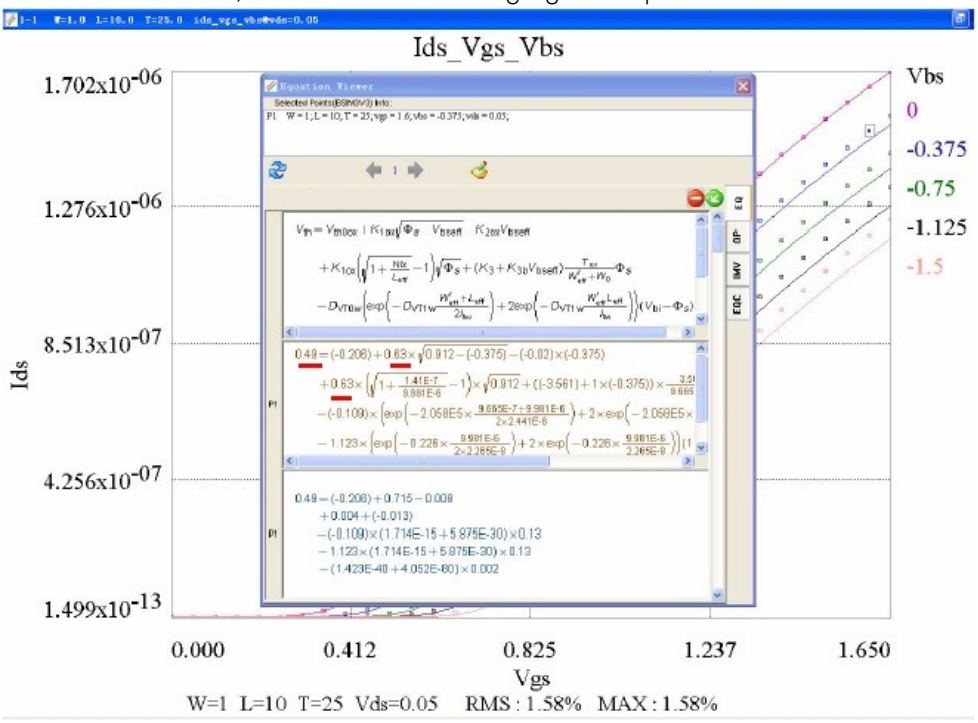


NOTE Remember to activate multiple selection first if it is required to select more than one point from a plot. As for selecting data points, refer to Parameter and Optimization for details.

Shift Model Status by using EV Tool

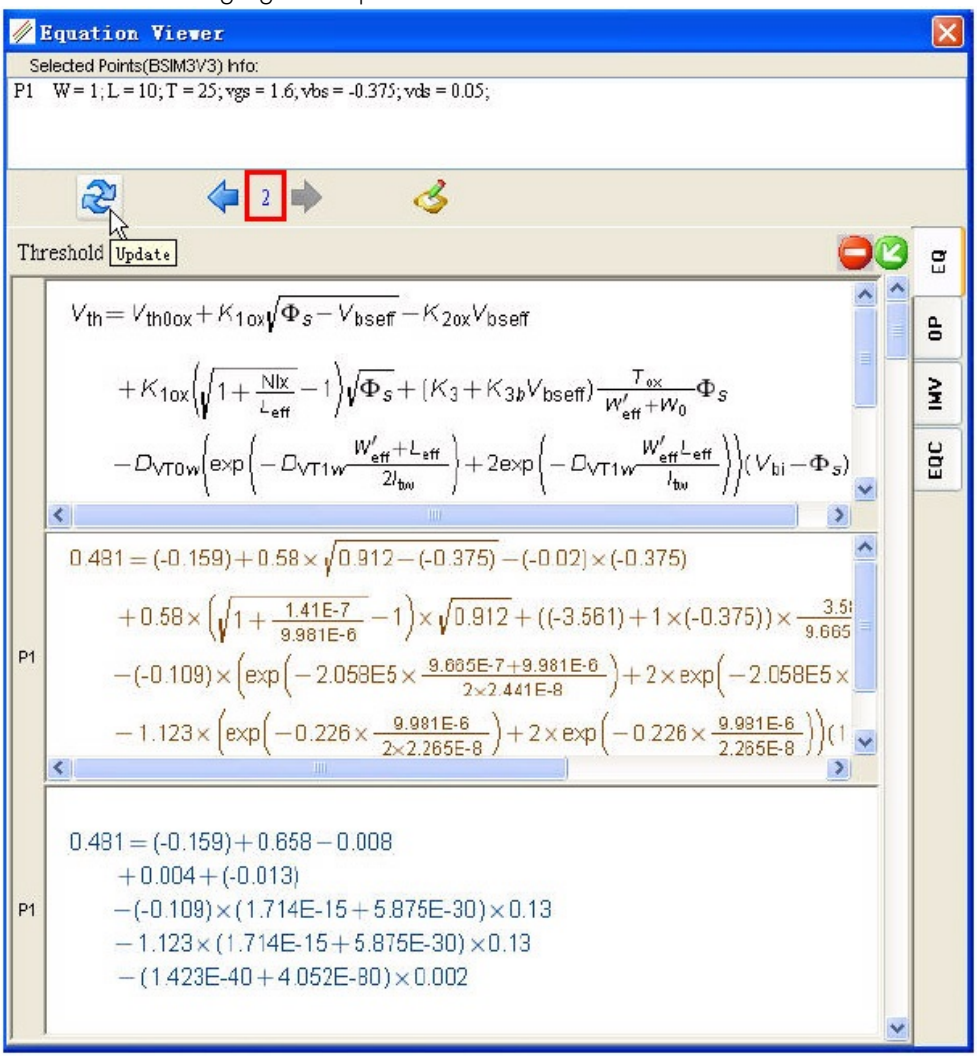
The capability of shifting model status provides a short cut directly to debug model issues. Here, model status is defined as a model with a specific set of parameter values. In other words, different model status means that model parameters have difference in values. When fine-tuning parameters, it is helpful to see how much influence it brings to the target value such as Vth and Idsat. And sometimes it's puzzling why some adjustments do not have any effects on the fitting results. In such cases, use EV Tool to switch between different model statuses. Following sample flow describe the operations in detail.

1. Select a data point on page Ids_Vgs_Vbs and enable the Equation Viewer. We define the current model as status 1, with parameter K1(K1ox)=0.63 and resultant Vth=0.49, as shown in following figure. Equation Viewer on status 1

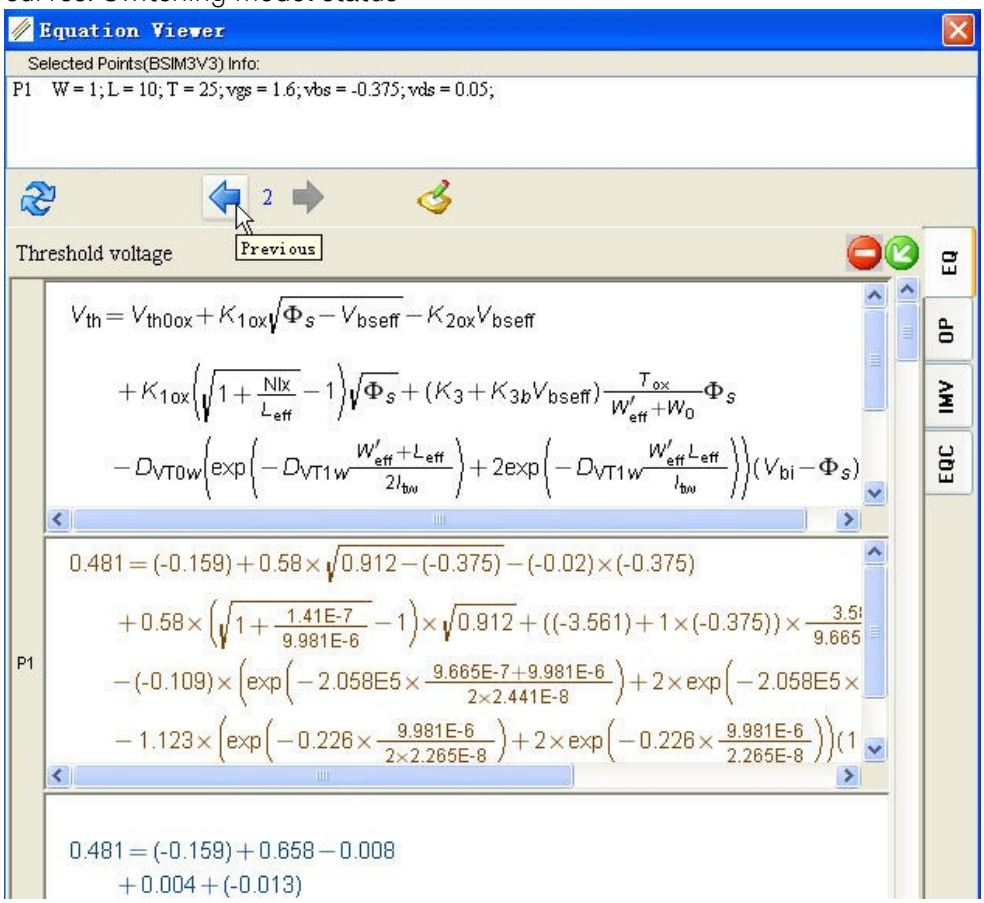


2. Adjust the value of parameter K1 from 0.63 to 0.58.

3. To update this change to the Equation Viewer, i.e. to define a new status, click Update icon  in EV Tool Bar. The number 2& denotes status 2. For verification, we can see K1(K1ox) is already set to 0.58 in the viewer. Correspondingly, we get Vth of 0.481 instead of the preceding value 0.49, shown in following figure. Equation Viewer on status 2



4. As shown in figure below, by clicking Previous or Next button, you can freely switch models between status 1 and status 2, or any other newly defined status. EQ, OP, IMV, and EQC changes accordingly together with simulated curves. Switching model status

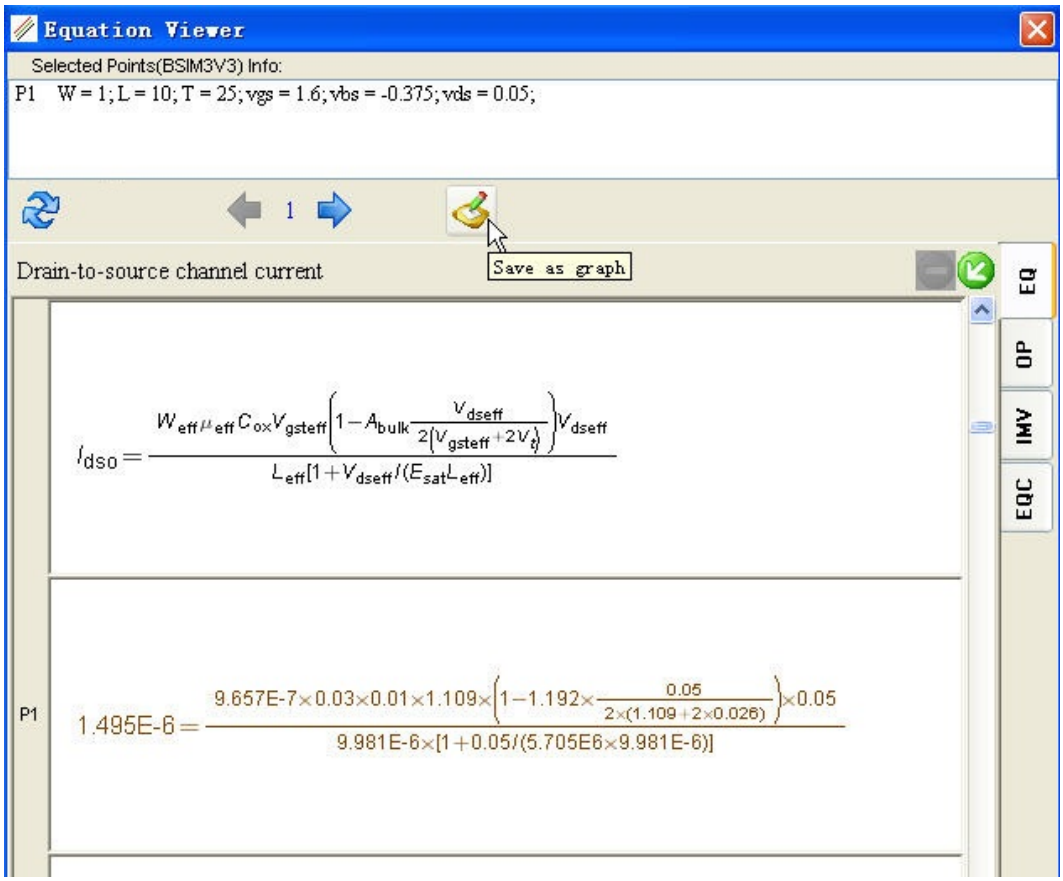


NOTE The Equation Viewer does not limit the amount of user-defined model status. However, to keep the Equation Viewer running smoothly, we recommend not to define more than five status.

Output Equations as Graphs

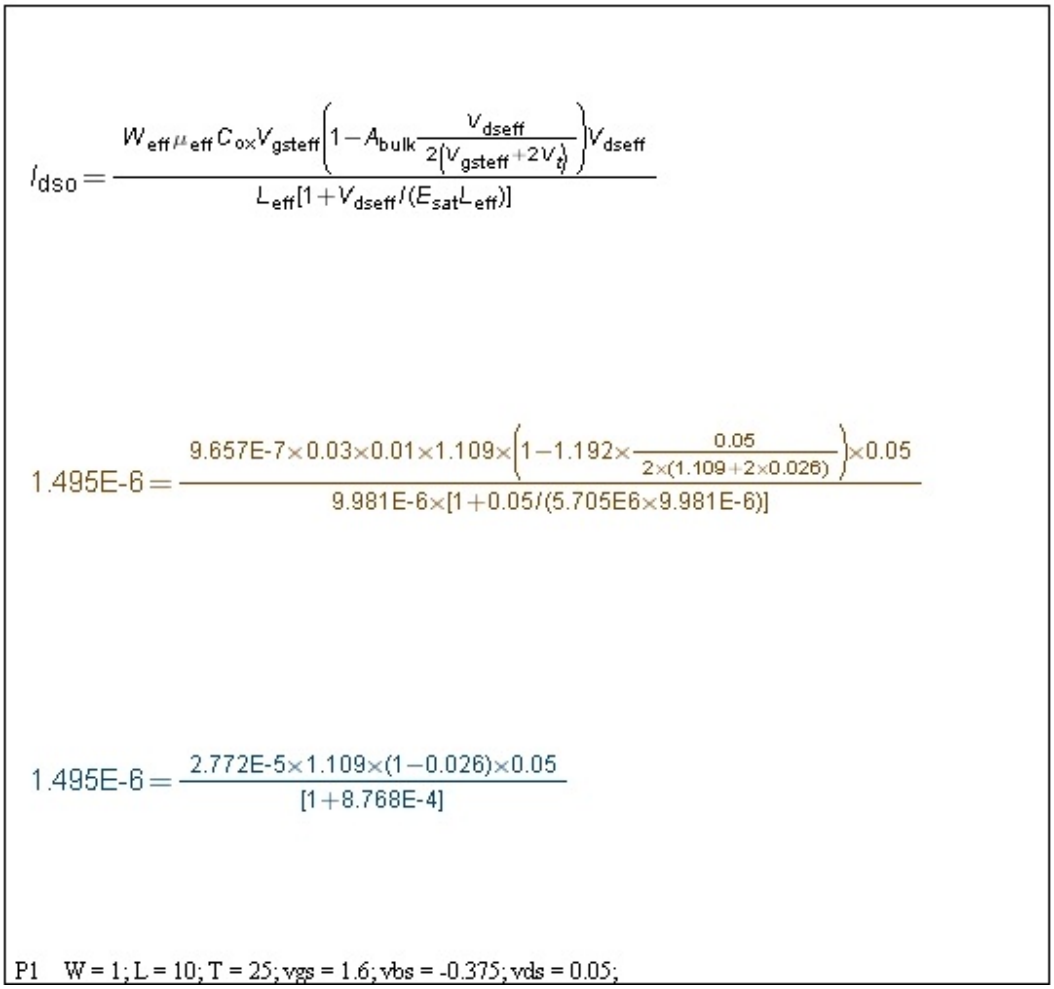
To output equations in current view, click Save as graph icon  in EV Tool Bar, as shown in following figure.

Icon of Save as graph in EV Tool Bar



An example output graph is also shown in following figure. It keeps information of both original equations and calculating results. We call them value equations. And the instance parameters and bias conditions are printed at the bottom of the figure.

Example of outputted Equation Viewer



NOTE This function is only applicable to EQ.

Export Graphic

Graph is necessary for good reporting. There are four options available to export graph in MBP. You can click Export > Export Graph and choose the corresponding options.

All Pages

If you select All Pages, all the pages of all loaded devices will be output in linear scale. MBP assigns name to each page; specify the directory for saving the plots. The plots will be stored in JPG format.

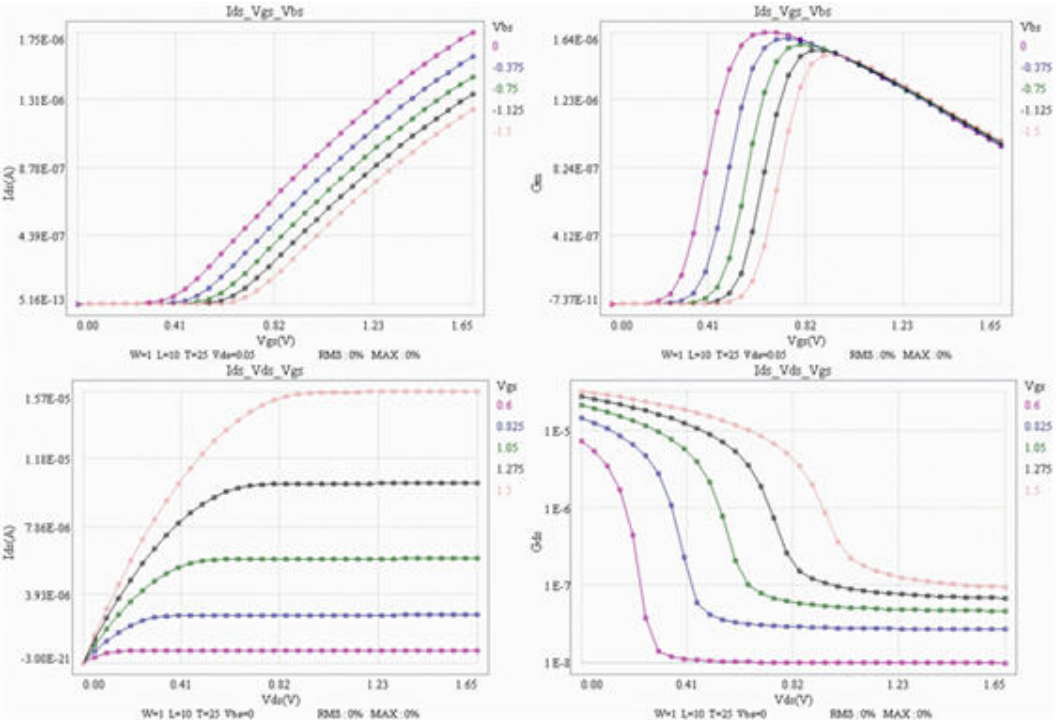
Current View

MBP could also output the images displayed with math function and scale applied by choosing Current View. However, it only outputs all the pages within current screen, not all loaded devices. The plots will be stored in GIF format.

Screen in One

This option Screen in One means to save all the pages within current screen as one image file. See following figure for an example. The plot will be stored in GIF format.

Sample outputted picture with Screen In One option



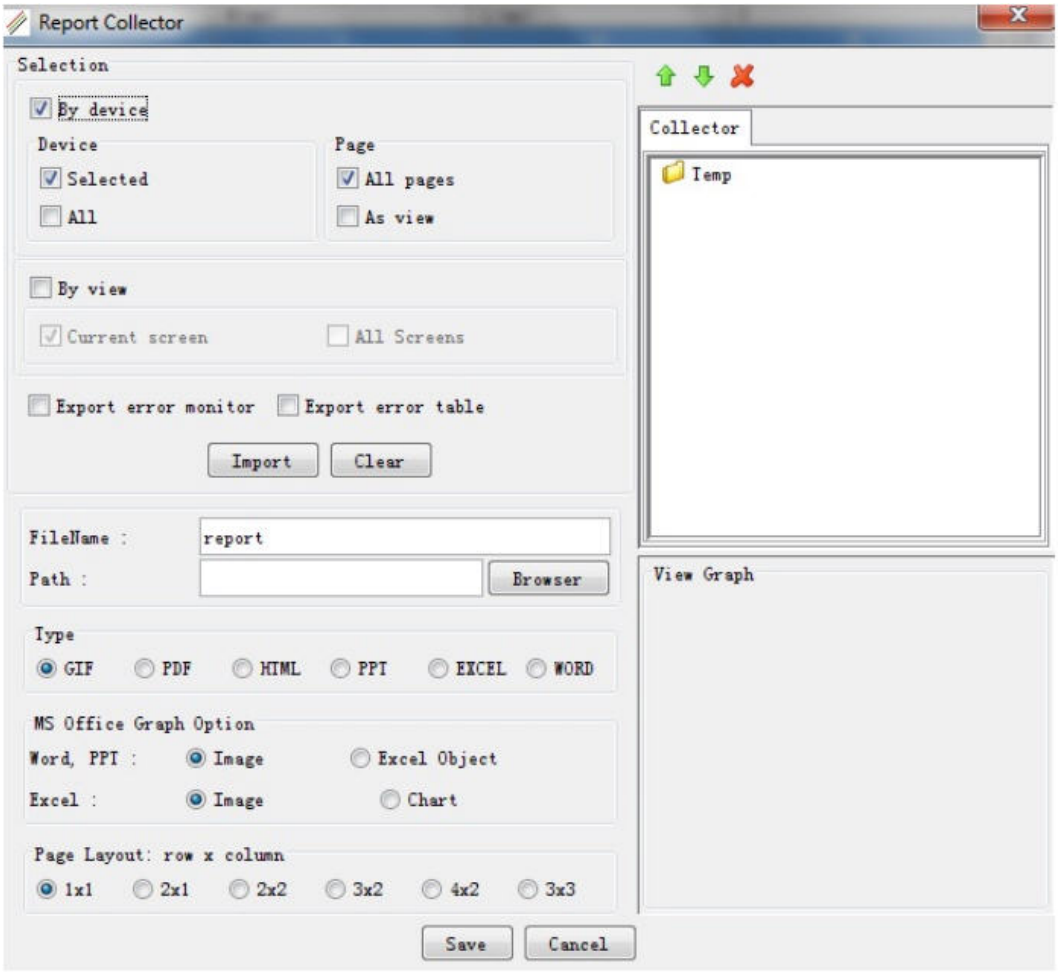
NOTE

For all the three options above, if you do not want to show RMS and MAX values, check **Hide RMS and MAX** options in RMS settings (**Tool > GUI Options > RMS**) before exporting.

Advanced

Choose Export > Export Graph > Advanced to pop-up Report Collector window. You can customize the report format (such as adding comments to the graph, adding /removing graph, adjusting graph sequence...), as shown in following figure. If the exported type is Word, Excel or PPT, you have the option to save the file as image or Excel object. See application note Advanced Graph Export for more details.

Report Collector

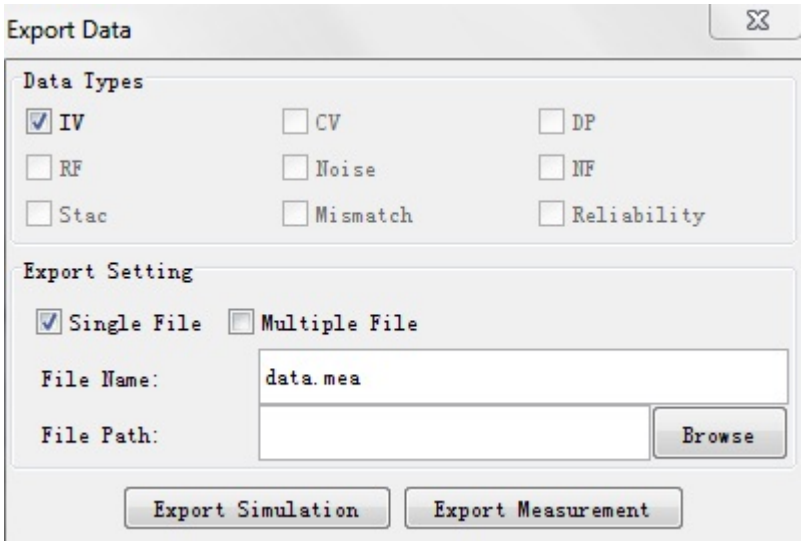


NOTE For **advanced** options, refer to the application note of Advanced Graph Export.

Export Sim

MBP also supports simulation results export functionality. Choose Export > Export Sim from the main menu and the pop-up window is shown as in following figure. The simulations are saved with the format of measurement data (*.mea). Therefore, you can load simulations the same way as data loading. You can choose to export either simulation data or measurement data by clicking Export Simulation or Export Measurement. Combined with the function Generate Virtual Data, it is quite flexible to run and check simulation with MBP. It also provides an easy way of transforming other data to that of MBP compatible.

Export Simulation

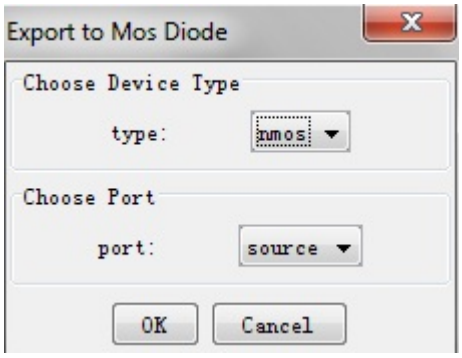


Export to MosDiode Data

CJGATE (CJSWG), PBSWG and MJSWG are used to model MOS gate-edge to bulk junction capacitance. This junction CV is often characterized in diode mode. However, the standard diode model is not capable of modeling it. Therefore, in MBP, if the drain or source periphery capacitance along the gate edge is measured in diode mode, it is required to convert it to MOS diode data first.

To activate this function, select one diode model first. After clicking Export > Export to MosDiode Data from the main menu, the window is displayed as shown in following figure.

Export to MosDiode



It is necessary to specify the device type (either nmos or pmos) and the port (either the junction is drain or source related). Click OK to choose the path to save. MBP assigns the file name automatically, mainly by instance parameters.

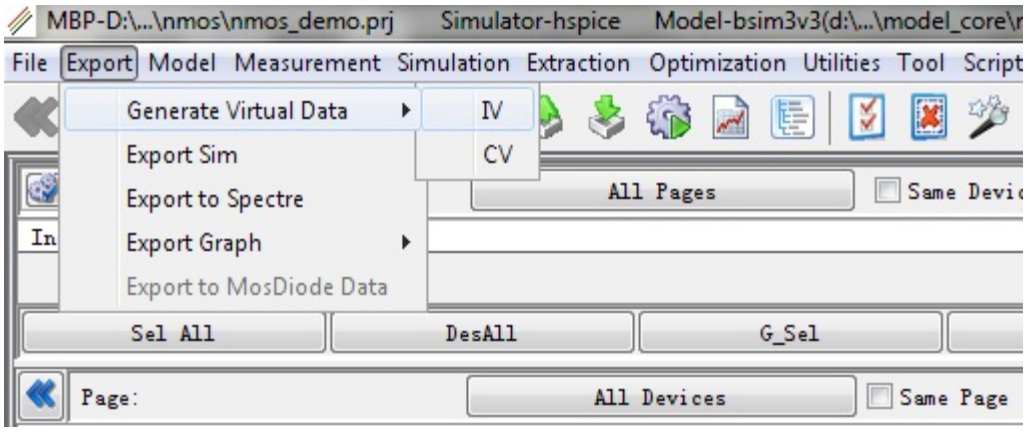
Export to Spectre

You can obtain Spectre compatible model card (or library) from HSPICE compatible model card (or library) by choosing Export > Export to Spectre from the main menu. A nice feature of this module is that it supports the conversion of model library directly. For such purpose, open the Lib Parser first. More details about Lib Parser are in Lib Parser.

Export Virtual Data

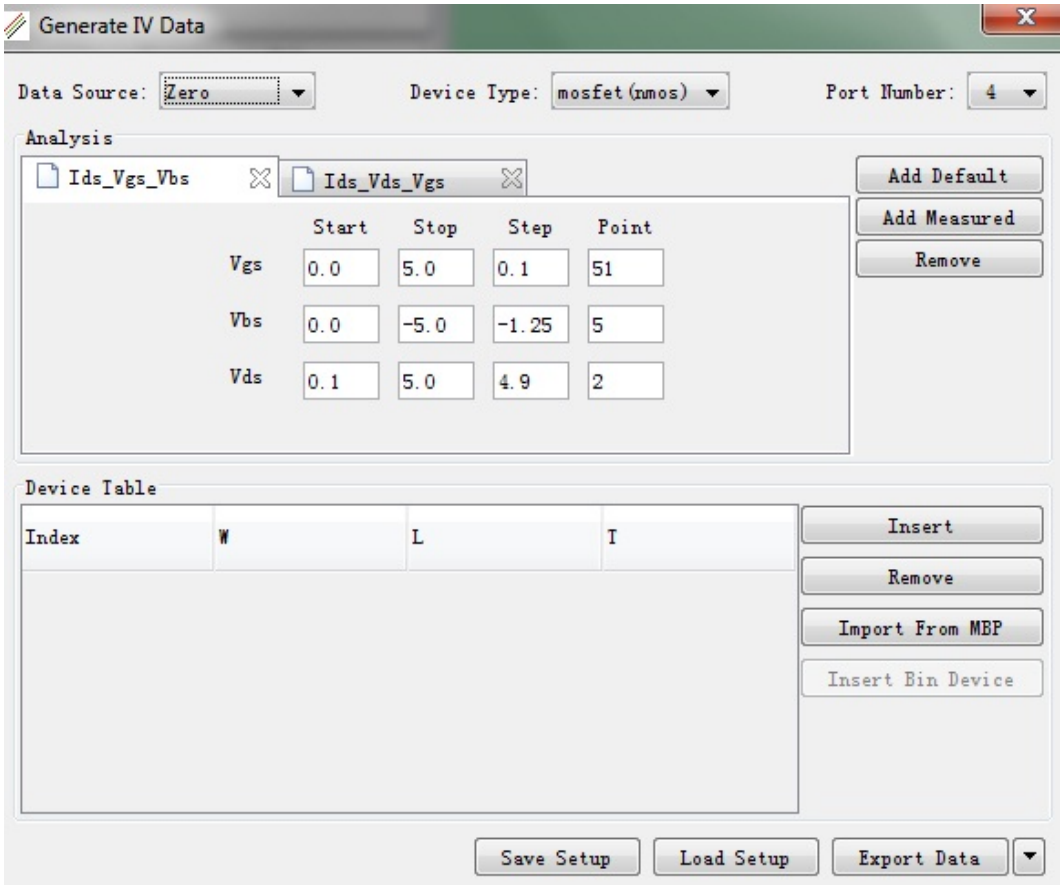
To view the I-V characteristics of a model without real measurement data, choose Export > Generate Virtual Data > IV (as show in following figure) to specify the sweeping conditions and instance parameters. See MBP Files for the generated data format.

Start Virtual Data module



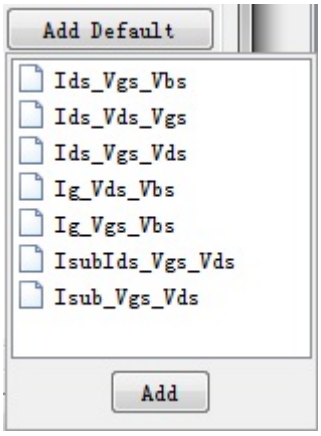
The procedure of virtual data generation partially borrows from the measurement module. The resulting Generate IV Data widow is shown in following figure. To generate a set of data, follow the steps as listed:

Generate Virtual Data

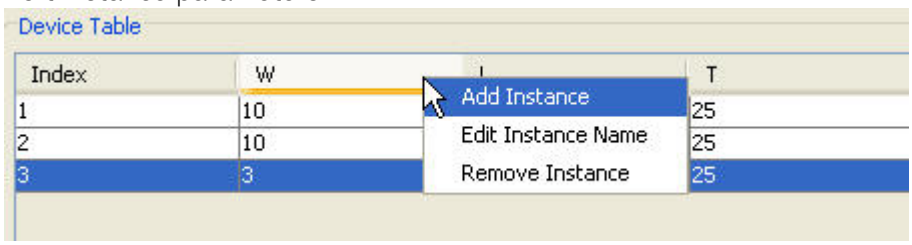


1. Select Data Source. In the drop-down menu, simulation means Y values of virtual data are identical to the simulations from current model in the background and zero means that all Y values are set to zero.
2. Select Device Type. Presently, mosfet (both bulk and SOI), jfet, bjt, diode, resistor and capacitor are available to choose.
3. Choose Port Number of the selected device.

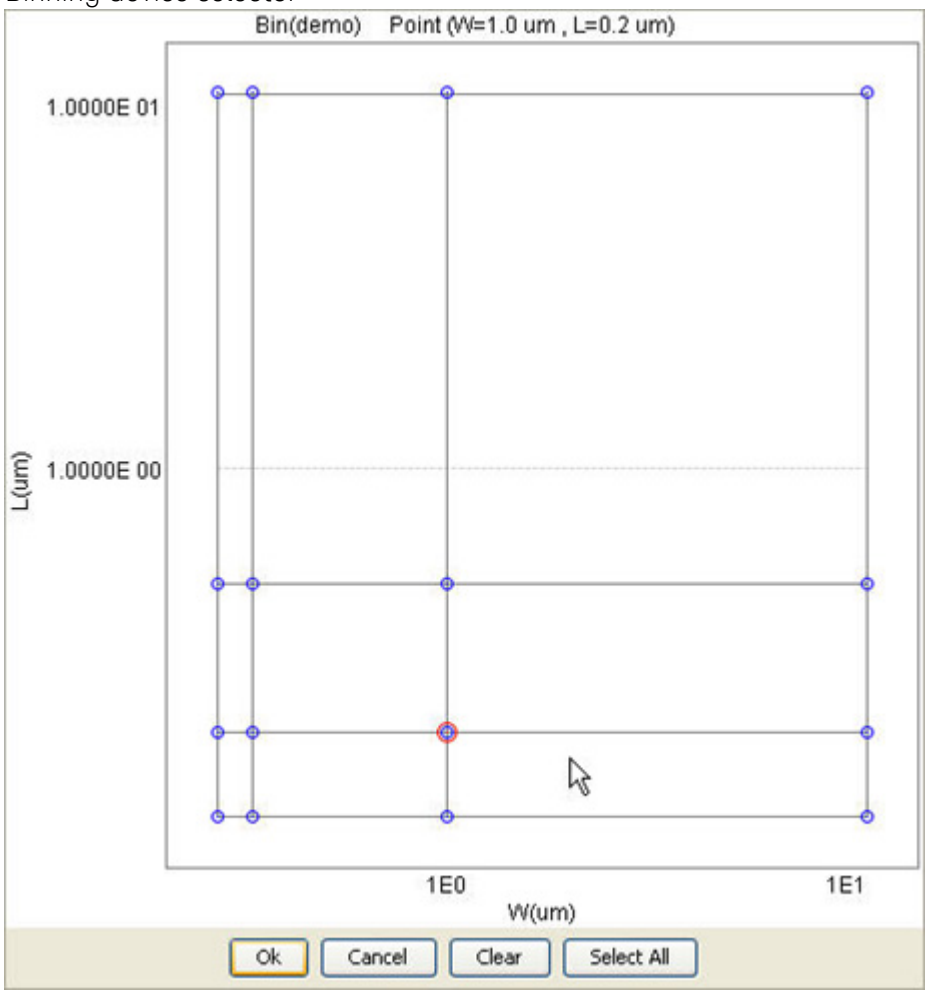
4. Choose the Analysis. An analysis is equivalent to one characteristic of the device. As in measurement module, MBP provides various built-in analyses to select from, as shown in following figure. Also, MBP parses the analysis of current data in the background in the Add Measured drop-down list. If no data is loaded, the list remains empty. Add the analysis



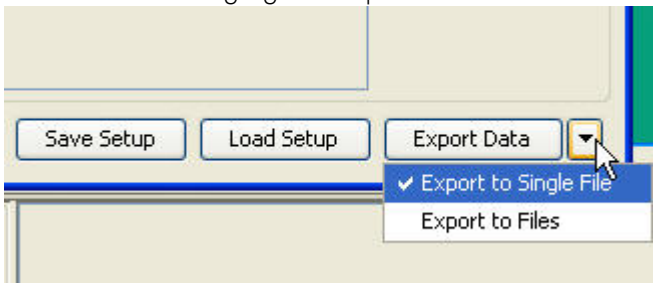
5. Set the bias/sweeping conditions for each analysis. For the items Step and Point, specify one of them. The other gets calculated by MBP accordingly.
Edit instance parameters



Binning device selector



6. The data can be output by clicking Export Data. If the data contains multiple devices, you can choose to save the data to one file or several files where each file corresponds to the measurement of a single device. The window is shown in following figure. Export the virtual data



Similar with IV data, you can also generate virtual CV data by choosing Export > Generate Virtual Data > CV from the main menu.

IMV Pages and DP Implementation

This section describes the features and settings of IMV Pages.

Introduction to IMV and DP

IMV stands for intermediate variable. For circuit designer, the physical quantities like Vth, Idsat, Gm, and Gds, and so on could reflect device characteristics more clearly, especially compared with model parameters. In MBP, these physical quantities are defined as IMV. Correspondingly, Modeling engineers often place more emphasis on fitting IMV. Also, the trend of IMV plots could effectively keep extracted model away from bad scalability. MBP enables definition of ones own IMV variables and output plots with high flexibility.

DP stands for device parameter and comes from Device Process characterization. Similar to IMV Pages in MBP, DP also outputs intermediate variables like Vth and Idsat on plots. However, the difference is that IMV gathers intermediate variables from general IV data while DP deals with intermediate variables directly.

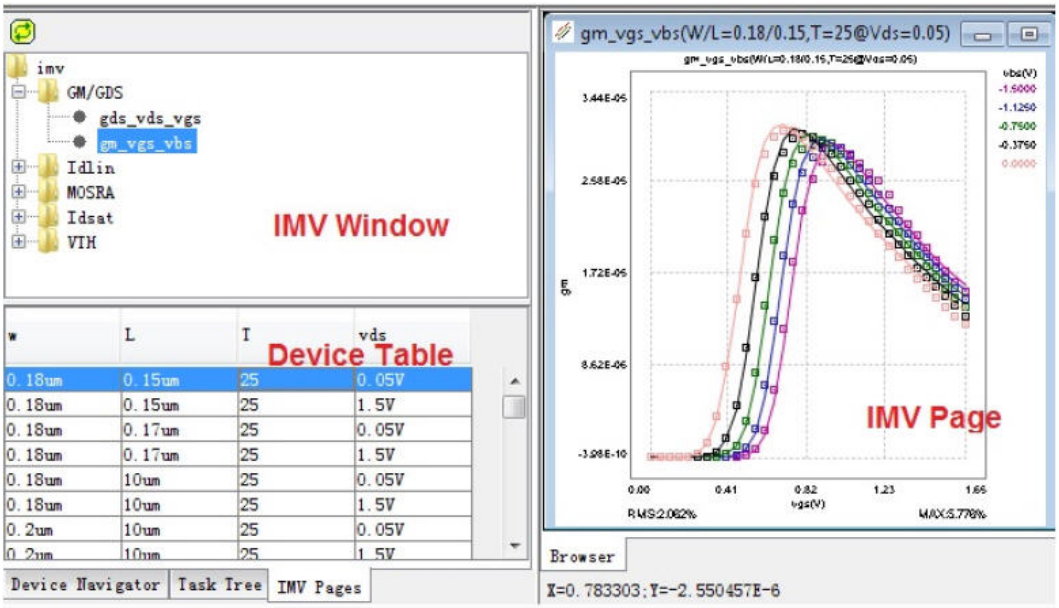
In summary, there are two main differences between IMV and DP:

- IMV calculates intermediate variables from the general IV curves. While DP deals with the intermediate variables directly because generally the value of intermediate variables is included in the DP data file.
- IMV plot calculates the intermediate variables according to the bias conditions and the sweeping conditions from the measurement data. Differently, DP plot gathers the simulation targets from the configuration files.

View IMV Pages

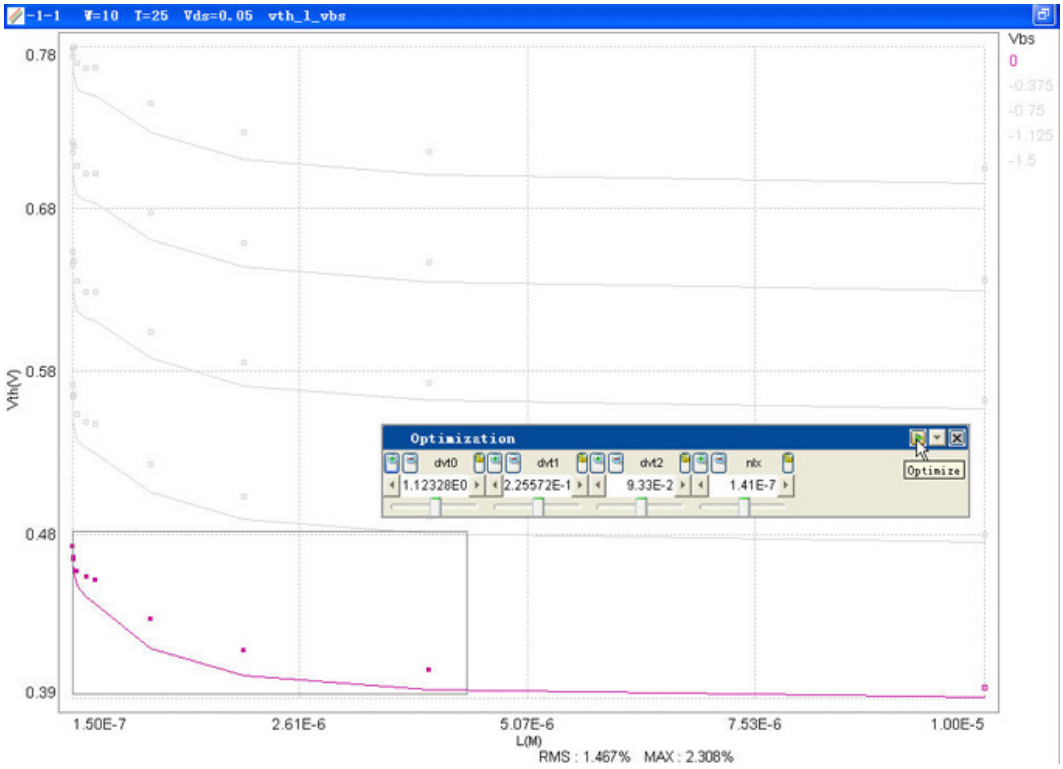
To open IMV pages in MBP, click Extraction > IMV > IMV Pages from the main menu or click the IMV Pages icon  in the tool bar. MBP displays all IMV pages by tree view in hierarchy, as shown in following figure. Select the type of the IMV page and the device to view, the corresponding IMV page gets displayed in the active window.

IMV page window



You can either optimize the IMV pages automatically (by clicking the Optimize icon  shown in following figure) or manually tune the corresponding model parameters.

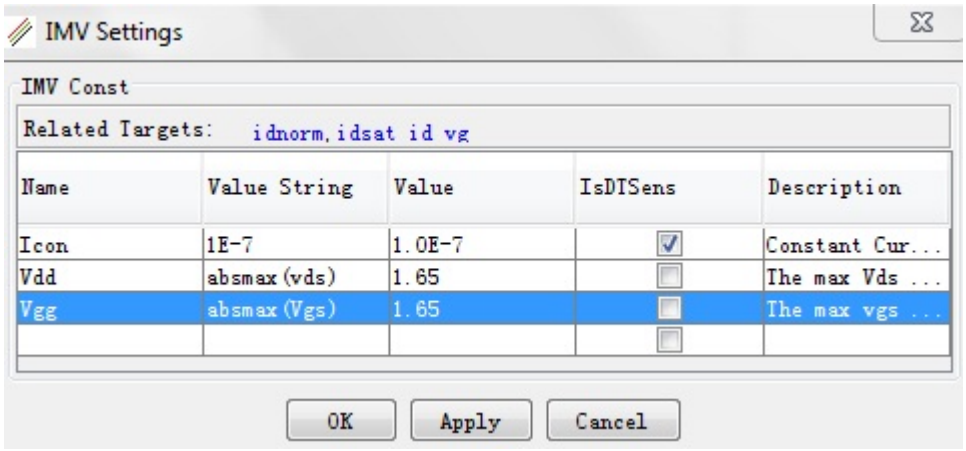
Optimization on the IMV page



IMV Page Settings

You can partially modify the IMV Setting on GUI. Click Extraction > IMV > IMV Setting from the main menu, the IMV Settings window pops up as shown in Figure: IMV Setting. In this window, you can customize IMV constants. When one row of the IMV constants is highlighted ("Vgg" in this example), the related targets get displayed at the top of the table (like "idnorm, idsat id vg"). The value of IMV constant could be modified by double-clicking the corresponding data in column Value. You can also set the value as an expression in the column Value String. In this example, "absmax(Vgs)" mean to grasp the absolute maximum Vgs value from the data file. IsDTSens means is device type sensitive. If the option IsDTSens is checked, the corresponding value is automatically multiplied by "-1" for the p-type mosfet device.

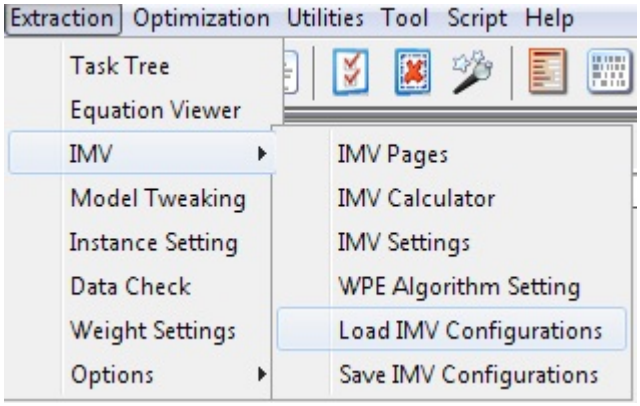
IMV Setting



In case of you have customized IMV configurations files, you can load the settings directly by choosing Extraction > IMV > Load IMV Configurations from the main menu (as shown in following figure). Settings are enabled automatically once loaded successfully. Also, to save customized configuration file, click Extraction > IMV > Save IMV Configurations from the main menu.

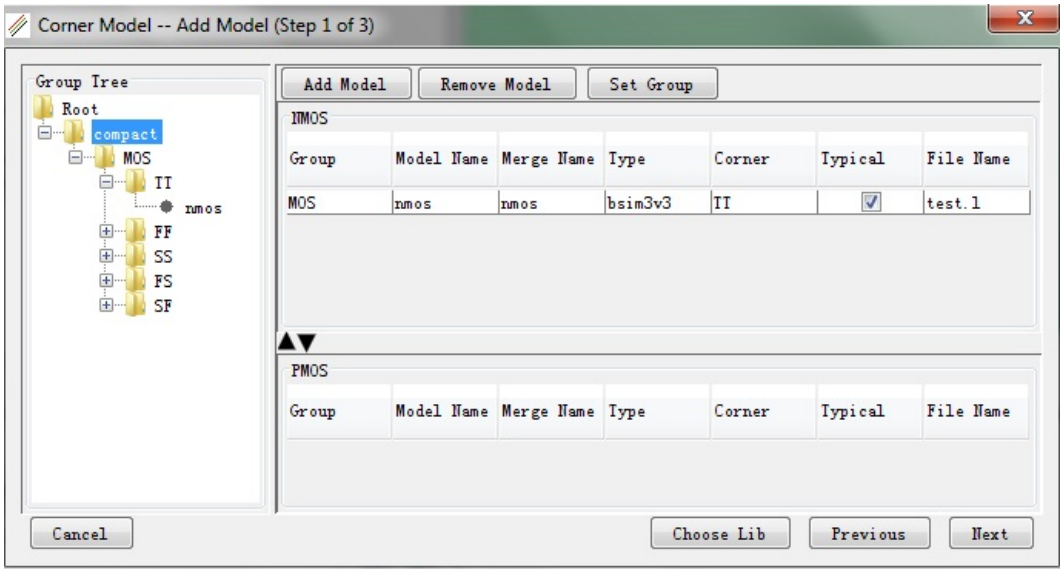
NOTE **While saving the configurations, all the old formats are converted to new script format automatically.**

Load IMV configurations



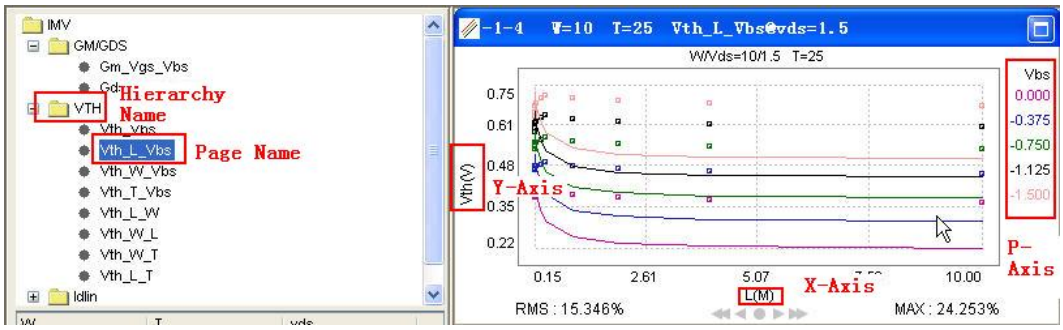
For the further configurations, you can customize IMV definition (vth, idsat, gm, gds, plot type (vth vs. L/W/T) and plot layout (group the plots). Lets use MOSFET as an example. All the configuration files regarding IMV are located in `\MBPHOME\jef\usr\imv\device\mosfet`,the file structure is explained in following figure.

File Structure for IMV pages



IMV values are derived from IV or CV data, for example, Linear Vth is derived from Ids vs. Vgs curve at Vds=0.1/0.05V. You can load IV or CV data in order to view IMV pages. These definitions can be seen from the final IMV plots directly. Their meanings are shown in following figure.

IMV Graphic Definition

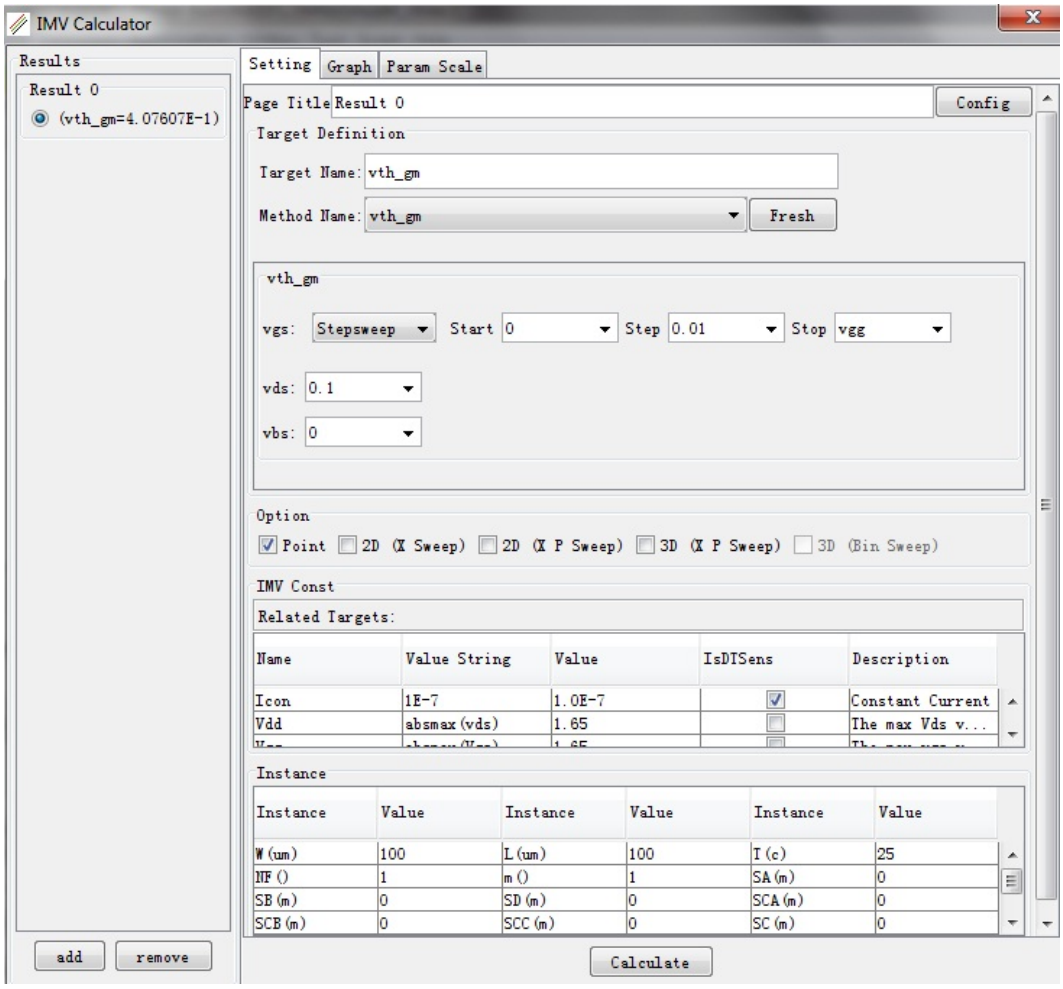


See MBP Files for more details about the files regarding IMV.

IMV Calculator

In addition to IMV pages, MBP also provides IMV calculator for convenience. Choose Extraction > IMV -> IMV calculator from the main menu, the main interface pops up as shown in following figure.

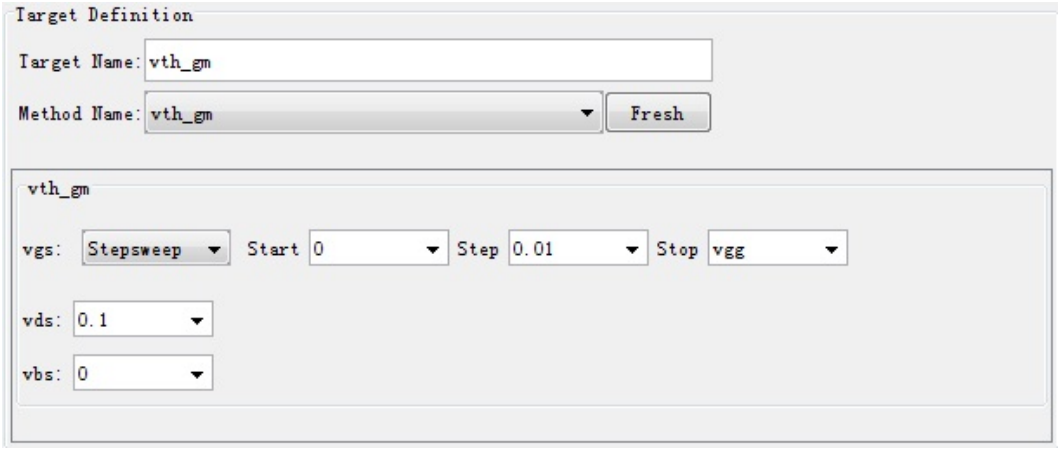
IMV calculator



In the Setting tag, enter all the configuration of the calculator. It contains the following parts:

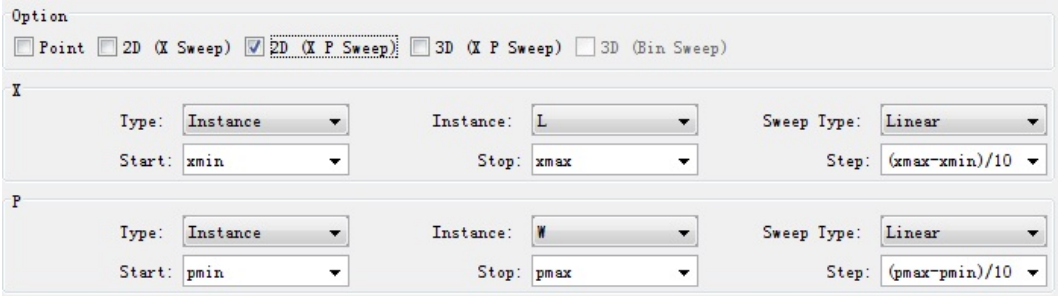
- Target Definition: IMV Calculator provides several IMV targets to choose, including vth_gm, vth_con, gm, gds, vth_gm_is, ids and igswl. In this part, Target Name, Method Name, and the sweep is required. You have an option to choose either Stepsweep or Pointsweep from the drop-down menu. For example, as shown in following figure, the method vth_gm is used to calculate Vth. As for the bias condition: Vgs is swept from 0 to Vgmax with the step 0.01V, Vds=0.1V and Vbs=0V.

Target definition



- Option: Here defines the output method of IMV calculator including point calculation, 2D plot with X sweep, 2D plot with X,P sweep, 3D plot with X, P sweep, and 3D bin sweep. When choosing any option other than Point, the calculator shows the X (and P) sweep options. Both instance parameters and model parameters are supported to sweep, and both Linear and List sweep type are available as well. The example is shown in following figure. It's a 2D plot with X,P sweep. X is the instance of L in Linear sweep. And P is the instance of W in also Linear sweep method.

Output Option



- IMV Const: This part defines the IMV constants which are the same of constants definition in IMV settings (by clicking Extraction > IMV-> IMV Settings from the main menu).
- Instance: Instance parameters (as shown in Figure 8.10) are defined in this part, including W, L, T, SA, SB, etc.

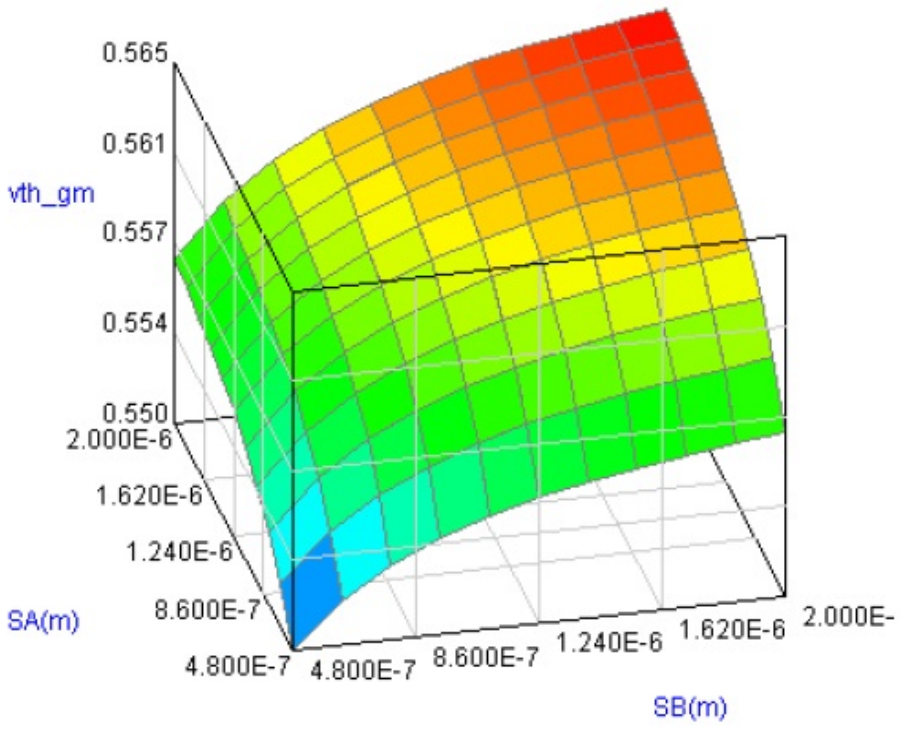
Instance parameters

Instance

Instance	Value	Instance	Value	Instance	Value
W (um)	100	L (um)	100	T (c)	25
NF ()	1	m ()	1	SA (m)	0
SB (m)	0	SD (m)	0	SCA (m)	0
SCB (m)	0	SCC (m)	0	SC (m)	0

After all the settings mentioned are done, click the Calculate button to proceed. For 2D or 3D plot output, switch to Graph tag to view the results. One 3D graph is shown in following figure.

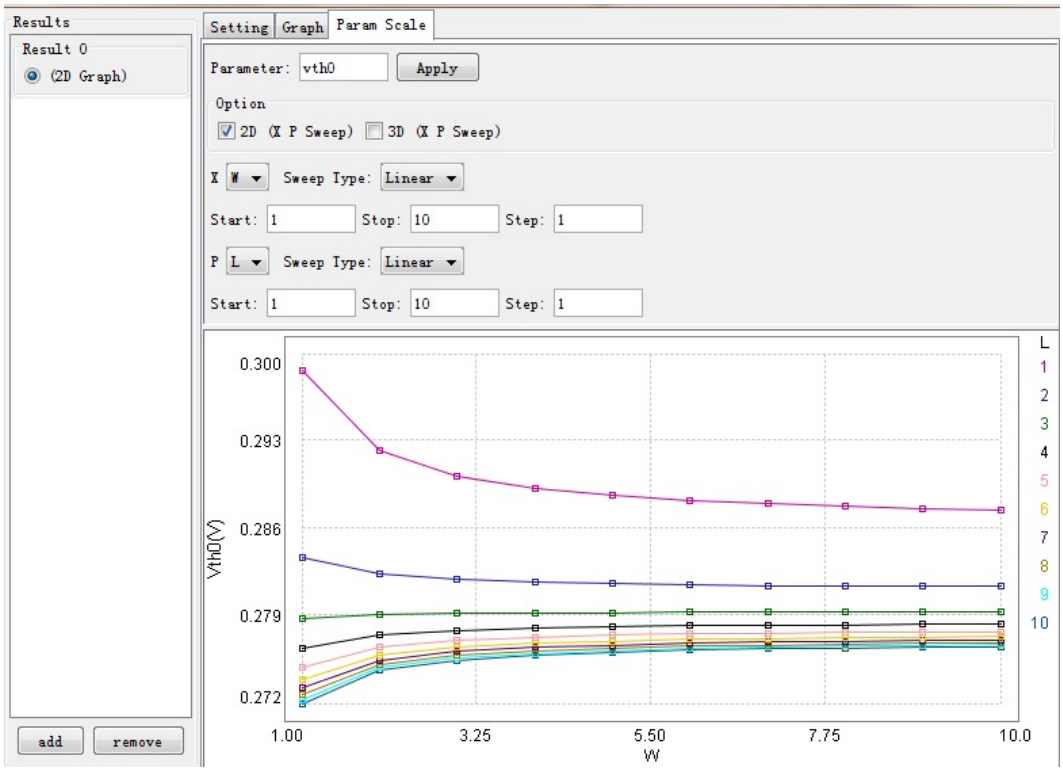
IMV calculator results (3D plot)



Right-click on the graph for more functions. Use View Data as Table to view the data in the format of table. Export to save the graph as a file with either graph (.gif) or text (.txt) format. And Linear/Log to choose one from the various combination of X-axis, Y-axis and P-axis. Besides, for 3D plot, IMV calculator supports zoom in/out with mouse wheel and rotate the plot in any directions by dragging the cursor.

In the Param Scale tag, plot the change of model parameter with the X-axis and P-axis. This function is especially useful for the binning models. One example of vth0 vs. W, L is shown in following figure.

Parameter scale



NOTE

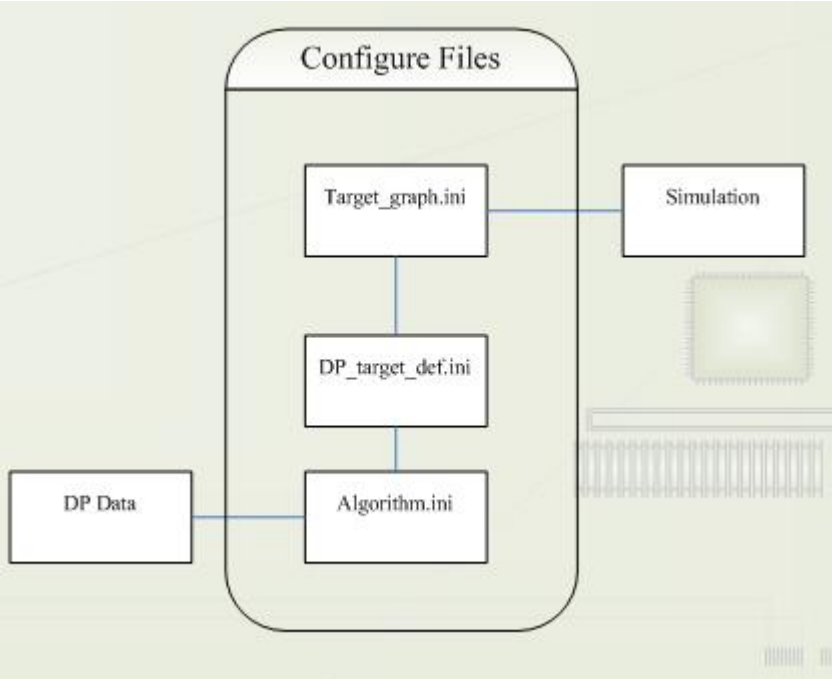
- The bias condition is required to be set properly to generate correct result. Especially, pay attention to the bias voltage of PMOS. It must be negative.
- There are no limits related to measurement data on the results or the sweep conditions of IMV calculator.

DP Support

DP plots output DP target vs instance parameters, in which DP target refers to intermediate variables, including Vth, Idsat, etc. MBP now supports loading ET data and comparing with simulation in DP module. You can also tune the model directly on the trend plots, such as Vth vs. SA/SB.

The DP configuration files structure is shown in following figure.

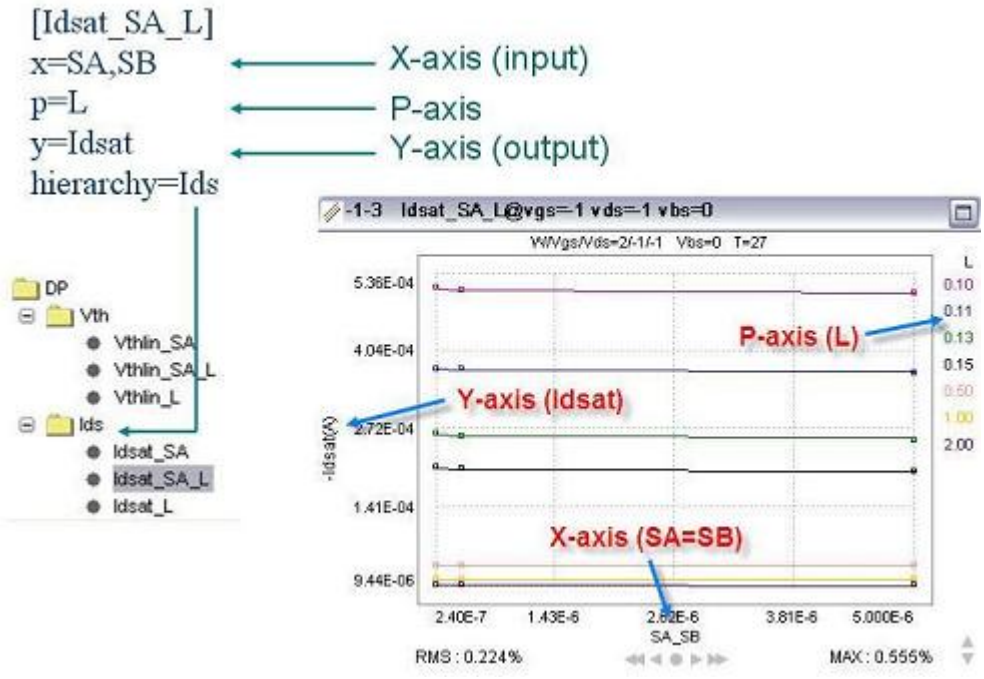
DP Configure files structure



After configuring the initial files, you can load DP datato MBP (same as loading IV /CV data). MBP displays the DP data and the corresponding simulations in MBP. The operations are the same as IMV. You can also optimize directly on DP plots.

File `target_graph.ini` defines what to plot on GUI. The DP plot definition is shown in following figure

DP plot definition

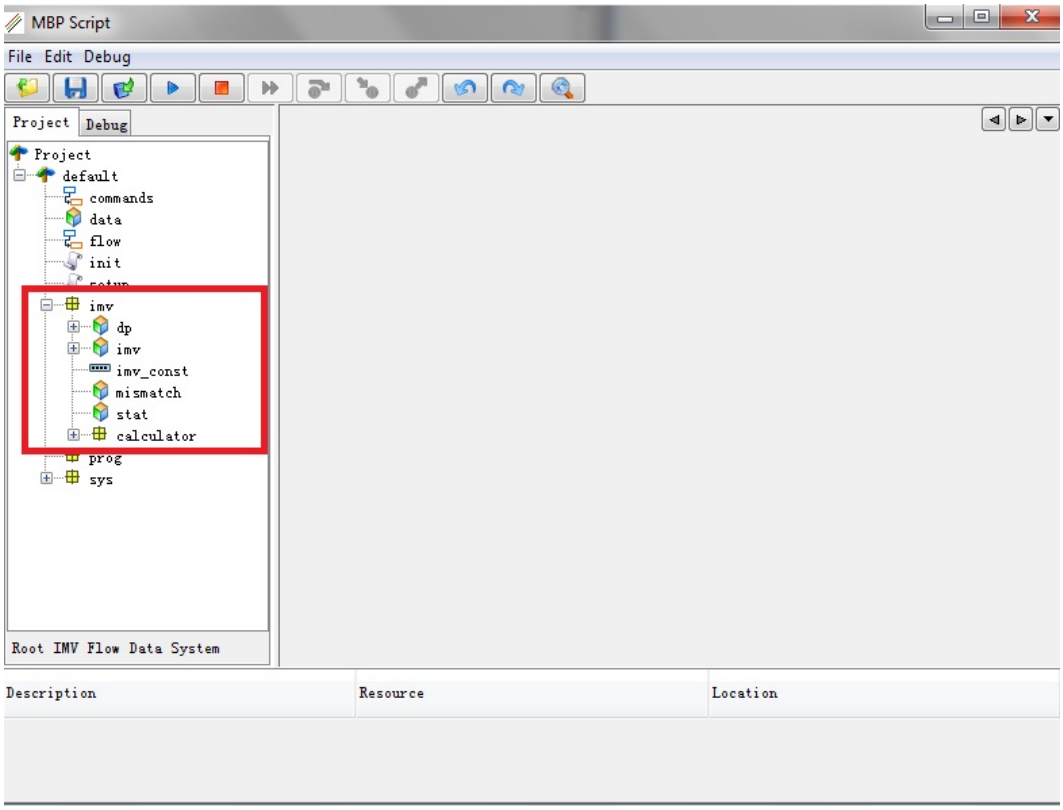


See MBP Files for more details about the files regarding DP.

Script Implementation

All the configurations regarding IMV are automated via scripts. To realize it, choose Script > Script Project from the main menu, as shown in following figure. IMV includes four catalogs of configurations: DP, IMV, stat, and mismatch. The old configurations of IMV and DP are still compatible with new version. However, when loading the old settings into MBP, the tool converts it automatically to new script format.

MBP script



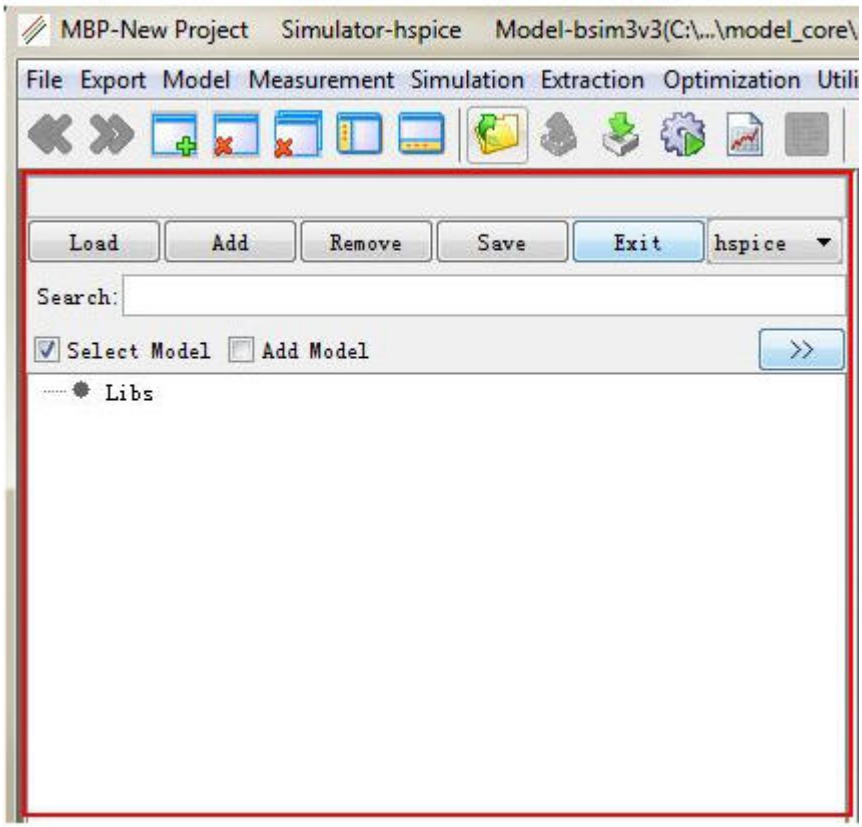
For more information about how to configure IMV by script, refer to *MBP Script Application Manual*.

Lib Parser

Lib Parser enables you to analyze and tune model library directly in MBP. You can access every corner library and model contained, adjust any parameters especially corner skew parameters, with viewing simulated target values (Idsat and Vth) at designated device size and bias condition instantly. It provides both modeling engineer and IC designer an easy and convenient way of skewing corner models to reflect process variations.

By clicking Utilities > Lib Parser from the main menu, the window shown in following figure is displayed.

Main view of Lib Parser

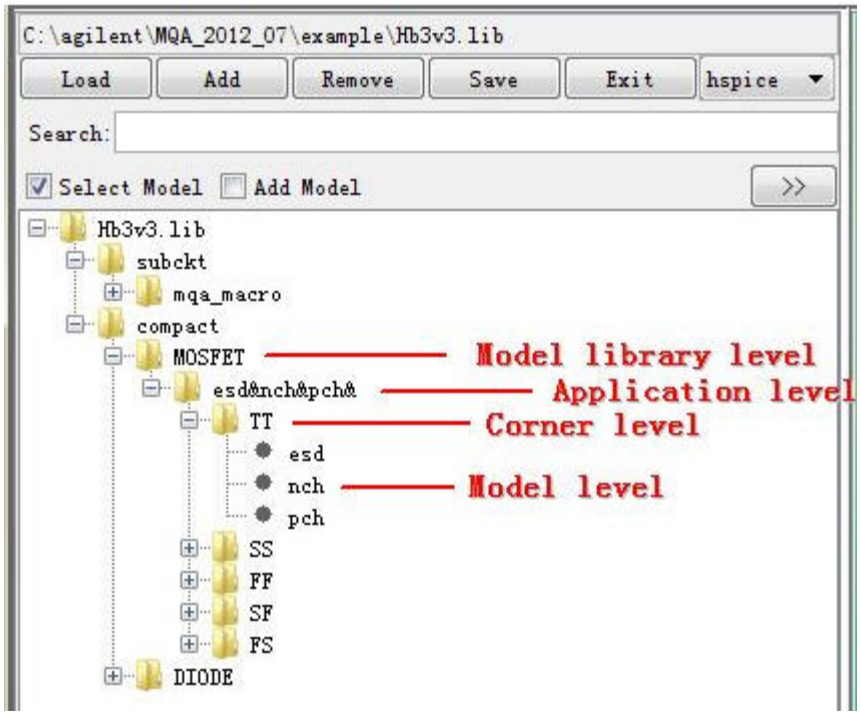


Model Library Window

Read the model library by clicking Load button at the top of model library window. As illustrated in Figure: Model Library Window, model library is displayed in tree view. This hierarchical structure has four levels, from top to bottom (also from high level to low level), they are:

- Model library level
- Application level
- Corner level
- Model level

Model Library Window



A model library contains device models for various process variations (different applications). For example, some of models are for low voltage usage and some are for high-voltage usage. MBP can recognize different application levels and separate them in the window.

MBP also supports to load multiple models from different libraries to compare or tweak. Click Add button to load multiple model libraries. You can also use Remove button to remove the unnecessary model libraries from the list.

NOTE

- You can load only one model library at one time. Use **Add** to load multiple model libraries.
- Lib Parser can read in all MBP supported models including diode, GP-BJT, BSIM3, BSIM4, PSP etc. It also supports subcircuit models.

Output the Model Library

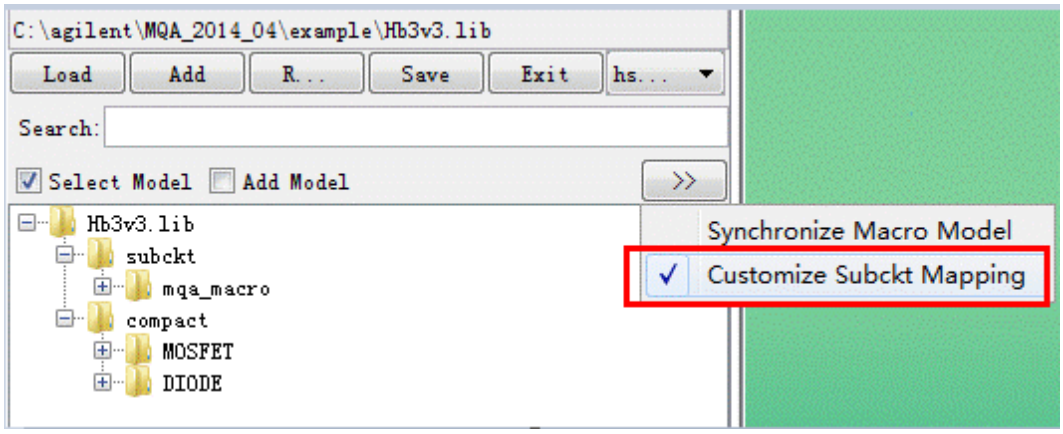
Click the Save button on the top of model library window to output the model library.

NOTE

Clicking **Save** directly overwrites the original model library. However, two files with extension .ver and .0001 will be generated at the same time. The one with .ver is a log file to record the revision information. And the other file with .0001 is actually the original model library file.

Other Feature

If Customize Subckt Mapping is disabled, the subckt mapping window won't pop up when selecting model. (Default is enabled)

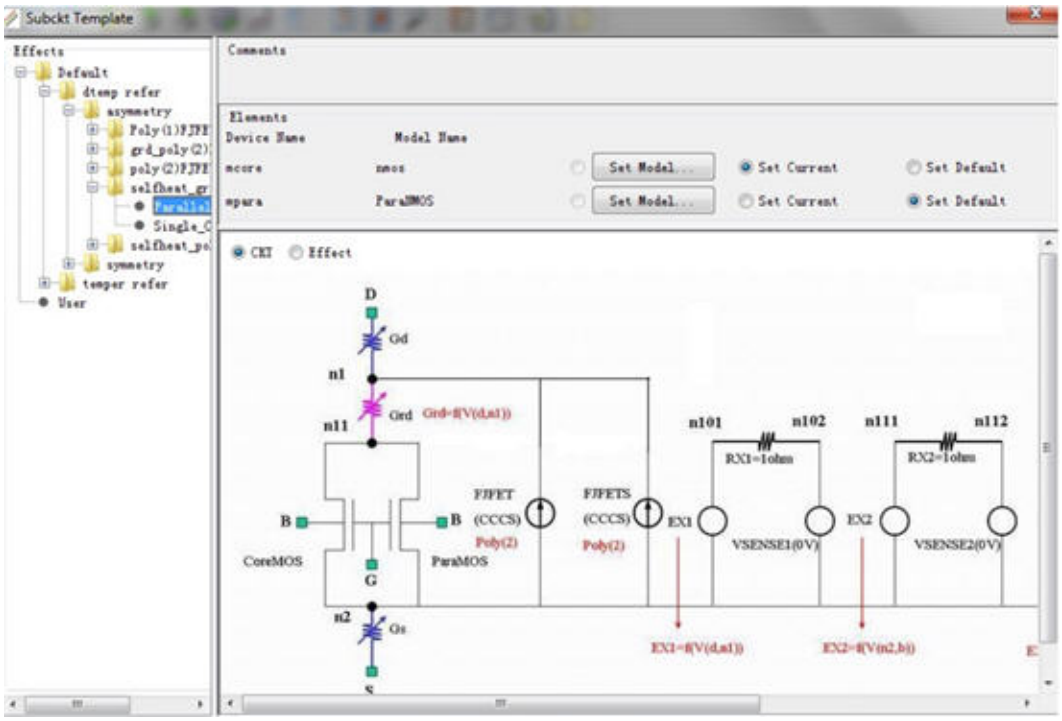


Subckt Template

With subcircuit template, you can generate subcircuit models very easily, with no worry about any typo error or standardization. It holds subcircuit model definitions. You can fill in core elements (like MOSFET, BJT) with desired model cards.

Look at our built-in subcircuit templates:

Main view of the HV Template



The built-in HV templates of MBP are divided into asymmetry and symmetry structure. The next level shows the different physical effect corresponding models to conquer. For each subcircuit model, you can either view its equivalent circuit by checking CKT or see which kinds of effects the model to describe by selecting Effect . And after confirming which subcircuit models to use, to specify models for core elements, there are 3 options:

- Set Model: Load a model file from hard disk
- Set Current: The model in MBP right before starting HV Template
- Set Default: The default model defined in MBP

After setting models for core elements, you can output the subcircuit model by clicking Save Model.

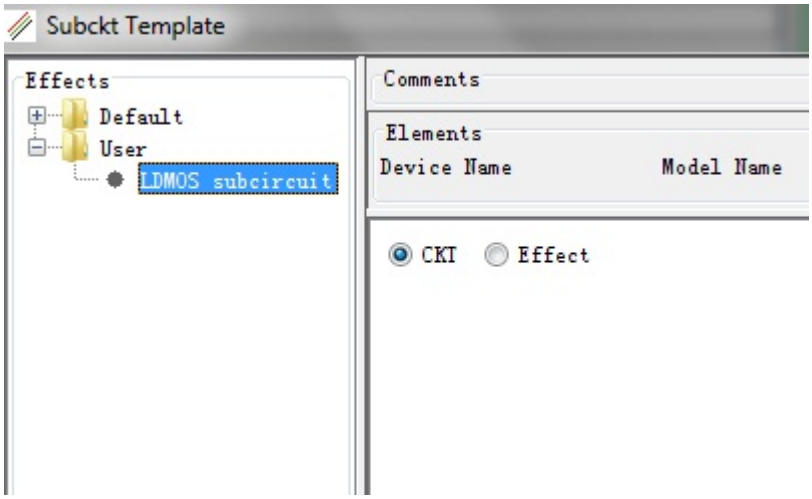
It is easy to append your own templates to the HV Template. For example, you have an n-type subcircuit model to be made as a template. Then do the following:

1. Create a new folder to `<MBP HOME>\JEF\usr\hvtemplate\MOSFET`, as `LDMOS subcircuit`. If needed, you can also create subfolders under it.

2. Rename this n-type subcircuit model to **`model_nmos.l`**, put it to the folder created. In case of p-type subcircuit model, name it as **`model_pmos.l`**. The naming conventions must be followed.
3. This step is optional. To show equivalent circuit and effect figure in the HV Template, name them as **`subckt.jpg`** and **`effect.jpg`**, respectively. Put them together with model_nmos.l/model_pmos.l.

Following above steps, the next time HV Template is started, you can see the defined subcircuit templates under expanded **`User`** set, as shown in following figure.

User-configured subcircuit model template

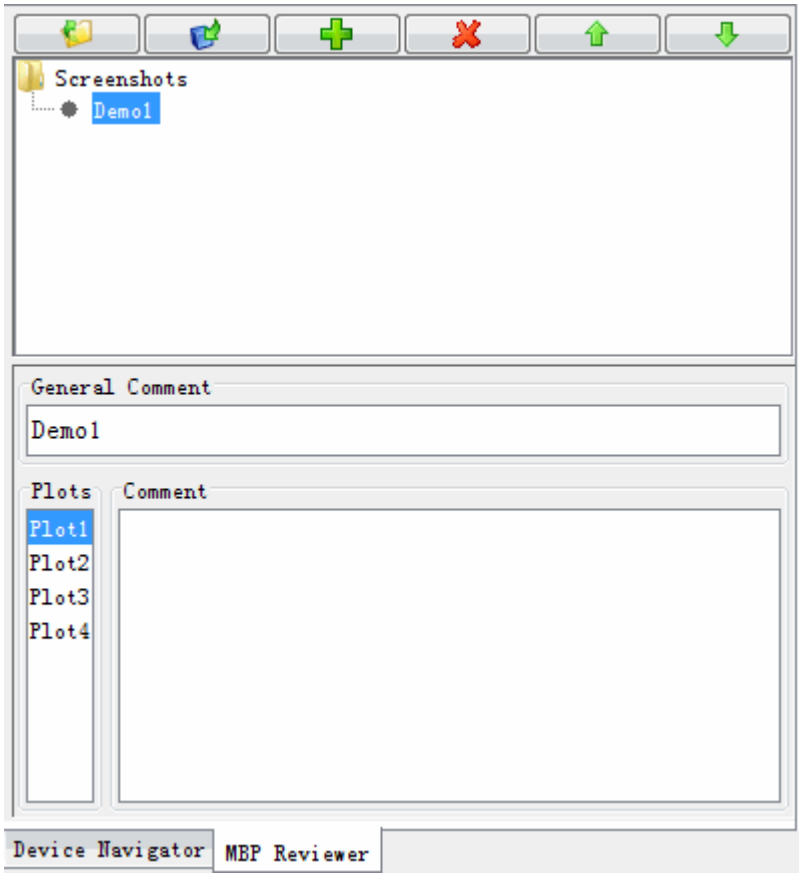


MBP Reviewer

Overview

MBP Reviewer is a result sharing solution to solve the painful communication problem between modeling engineers. We used to write documents, screen shots, etc to describe all the issues need to be discussed between modeling engineers, and the engineers from both sides need to reproduce the issues and then work on the replies. This iteration can bring in a lot of additional problems by mistake, such as wrong model card, wrong devices, etc. MBP Reviewer provides a feature combining model card, data, setups, graphic and comments together to keep everyone on the same stage.

Start the feature from shortcut button  . A new tab will pop up as below.



All the operation buttons are located on the top of MBP Reviewer tab.

Save/Load

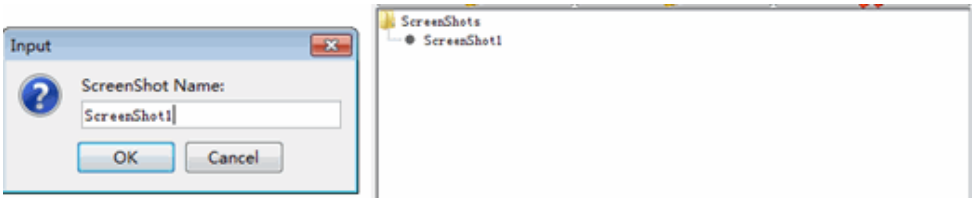
- Save MBP Reviewer: MBP reviewer will save a single zip file compressed model card, data (optional), model type setup, script setup, reviewer screen setup, etc.
- Load MBP Reviewer: Load zip file to show the exactly same screen /comments as saved.

Zip file contains:

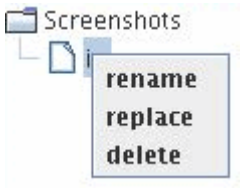
- Project file (.prj)
- Data: Data will be copied and zipped directly from original data file.
- Models: Model/Model library will be copied and zipped directly from original model file.
- Script: all the script setup in MBP including IMV setup, data organization, extraction flow, etc.
- Config: MBP model type configuration including analysis.xml, default instance setup, etc.

Screenshot operation

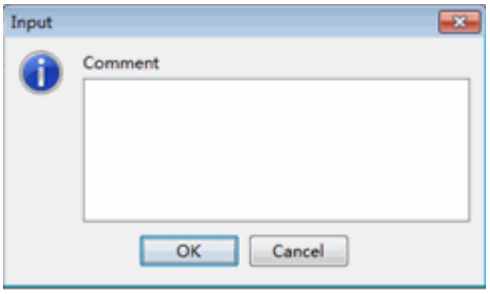
- Add screenshot
  Set layout and select plots on MBP main browser, click Add button in MBP reviewer tab. Input name. The screenshot will be shown in the tree list.



- Show screenshot
  Click on the tree node ScreenShot1, a new browser Reviewer will appear. All the screen status will show exactly the same with the Browser added.
- Remove , up and down
  Click on the tree node and click the delete, up or down button.
- Rename/Save/Delete screenshot
  - Right click on a tree node and choose Rename, user could rename the tree node.
  - Choose Replace, current screen status will replace the previous one.
  - Choose Delete to remove selected screenshot.

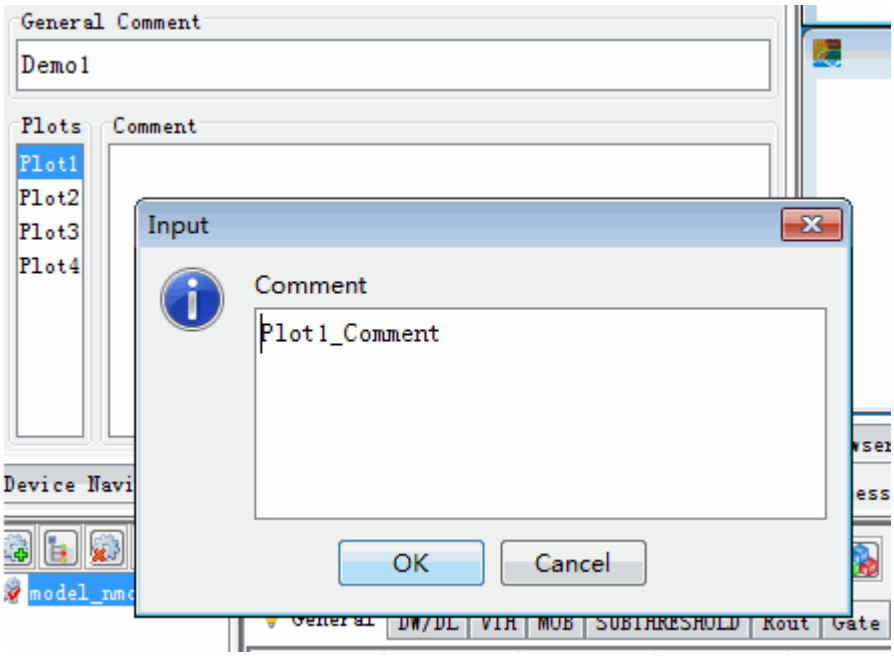


- Edit screenshot comment
  Double click on the Total comment blank field, an input pane will pop up. Input the comment and then click OK.



NOTE When you click **Replace**, the previous comments are lost.

- Edit screen comment
  Select screen index on the screen list. Double click on the comment blank field, input the comment and then OK.

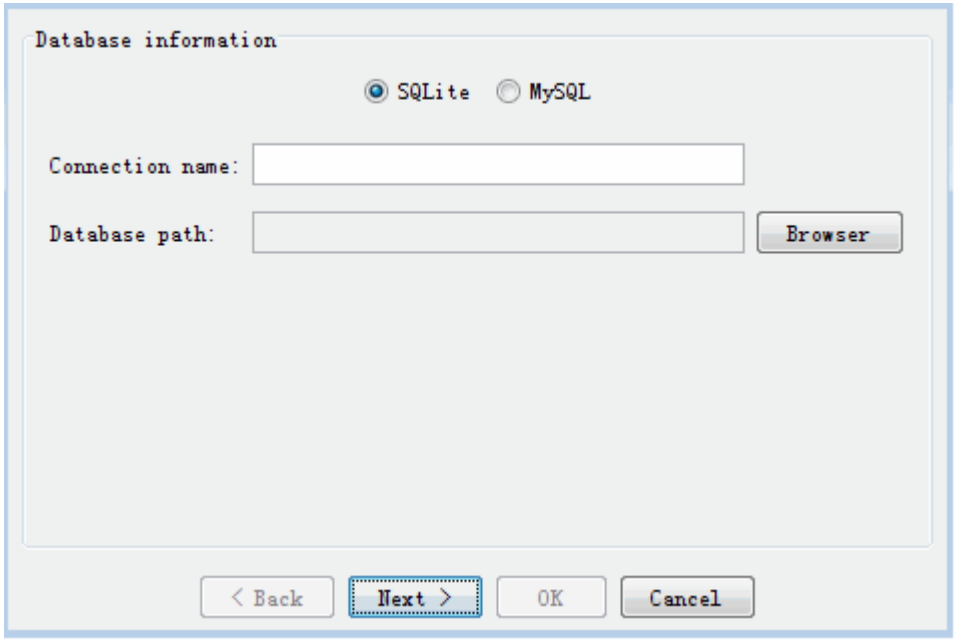


DB Data Loader

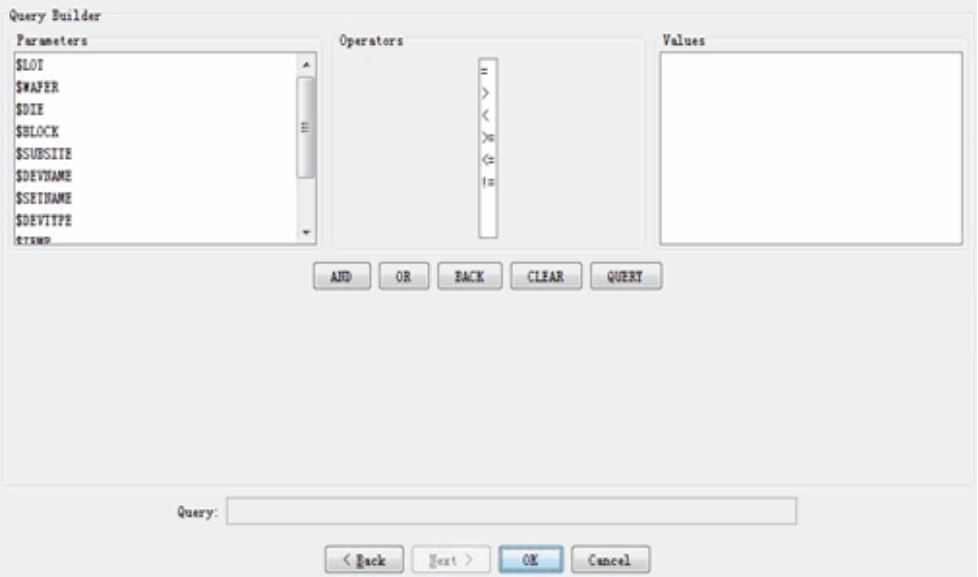
DB Data Loader

From menu Utilities > DB data loader, user could load and convert ICCAP database to MBP data format.

1. Select Data from DB and then choose data base information is SQLite or MySQL
   For MySQL, user needs to fill in the connect name, database name, host, username, password, and port, respectively on GUI.



2. Click next and the data base query will show on GUI.
   Select the parameters that we are interested in and the query string will be shown in the query box at the bottom.
   'Die''DEVNAME''$SETNAME' is obligated.



3. Click query button, then browser will fetch the data and generate a MEA file. Once it finished, MBP will pop up the path of MEA file generated.

MBP supports DC and CV data only in MBP version 2013.01.

Configuring Database with Database Connection Manager

Before beginning with Database Connection Manager, you must configure and create a link to MySQL databases.

The Database Connection Manager works with SQLite as out of the box solution and do not require any additional configuration.

This section provides information on how you can configure MySQL on Windows, Linux, and Solaris operating systems following the links below:

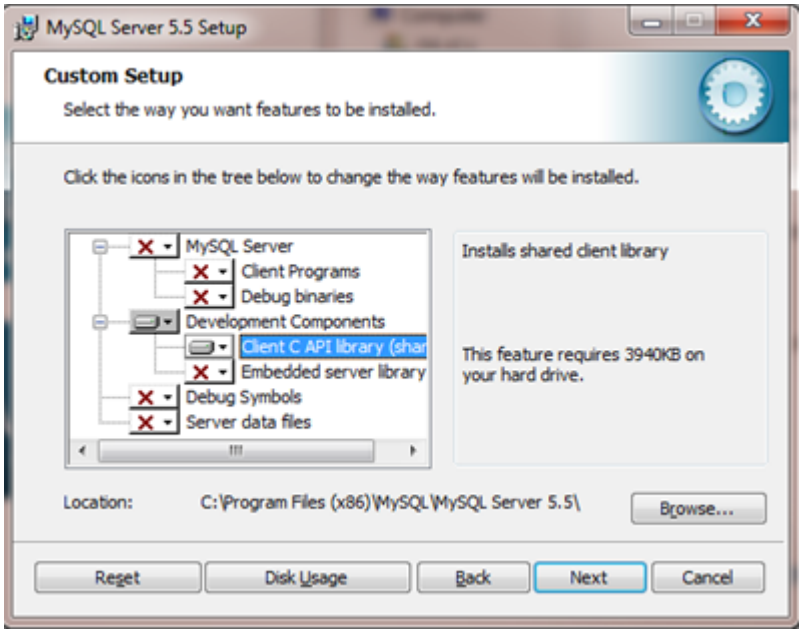
- Configuring MySQL on Windows
- Configuring MySQL on Linux

Configuring MySQL on Windows

The following sections provide information on how you can create and configure the plugin to link MySQL on Windows.

Download and install MySQL on Windows

Download and install the MySQL installer for 32-bit Windows available at http://dev.mysql.com/downloads/mysql/. If MySQL is already installed on your system, or if you are installing the 64-bit version of MySQL, you need to install the Client C API Library only. The 32-bit version of these libraries are required to create the plugin. For the 32-bit version of MySQL also, ensure that you include the Client C API Library in your installation, as shown in the following screenshot.

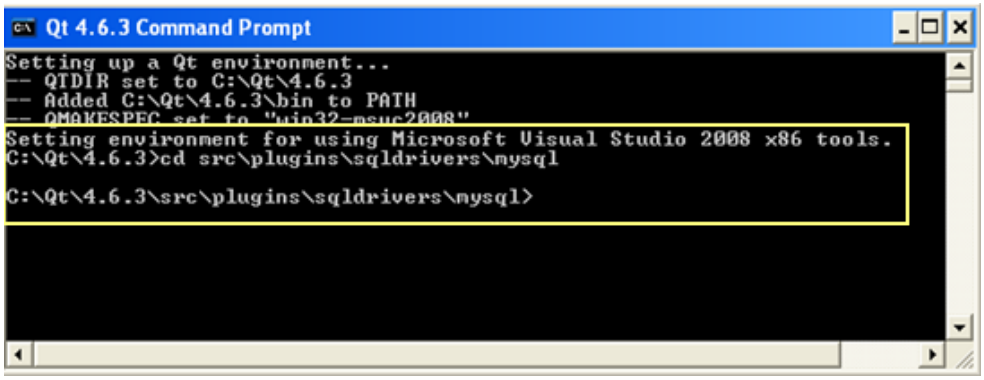


To install *only* the development library, ensure to deselect the Launch MySQL Configuration option in the last screen of the installer.

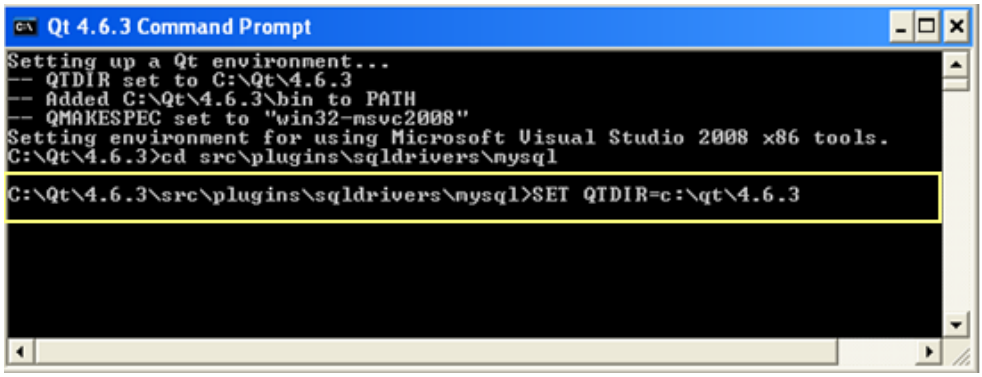
Configuring Qt and Building the Plugin

After the MySQL installation is complete, perform the following steps to configure Qt:

1. Download the *qt-win-opensource-4.6.3-vs2008.exe* file for Visual Studio 2008 from QT FTP Archive.
2. Install the downloaded *qt-win-opensource-4.6.3-vs2008.exe* file. By default, the Qt 4.6.3 is installed at *C:\Qt\4.6.3* filepath.
3. Navigate to Start > Programs > Qt by Nokia v4.6.3 > QT 4.6.3 Command Prompt to open the Qt 4.6.3 command prompt.
4. In the Qt 4.6.3 Command Prompt, change the directory to *%QTDIR% \src\plugins\sqldrivers\mysql*.



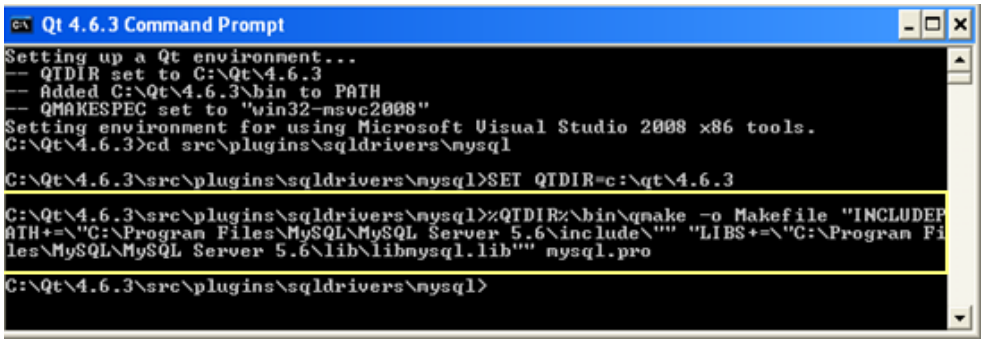
5. Set the **QTDIR** variable to the QT installation directory, for example, *c:\qt\4.6.3*.



6. Execute the following command:

```
%QTDIR%\bin\qmake -o Makefile "INCLUDEPATH+=\"C:
\Program Files\MySQL\MySQL Server 5.6\include\"" "LI
BS+=\"C:\Program Files\MySQL\MySQL Server 5.6
\lib\libmysql.lib"" mysql.pro
```

The following screenshot displays the above command in the Qt 4.6.3 Command Prompt window.



7. Then, execute the `nmake` command to build the *qsqlmysql4.dll* plugin in the *%QTDIR%\plugins\sqldrivers* directory.
8. Copy the *qsqlmysql4.dll* file and paste it in the *%ICCAP_ROOT%\bin\plugins\qt\sqldrivers* directory.
9. Copy the *libmysql.lib* file which is in the *lib* directory of the MySQL installed path and paste it in the *%ICCAP_ROOT%\bin* directory.

Now, launch Database Connection Manager from Tools menu in IC-CAP Main window.

You can access the MySQL option from the Select Database Connection Type drop-down list.

Configuring MySQL on Linux

To build the plugin for linking MySQL on Linux, you require development libraries package instead of full MySQL installation. There are various packages available for Linux, however, in this example, we have used the Redhat 4 package for supported IC-CAP Linux platforms.

NOTE **You may require a different package depending on your platform and versions of MySQL.**

Download and Install MySQL Development Libraries

Download and install the appropriate Development Libraries from http://dev.mysql.com/downloads/mysql/. In this example, we have downloaded and installed Development libraries under the Oracle Linux 4 & 5 section, specifically, the `MySQL-devel-5.5.18-1.rhel4.i386.rpm` distribution. You must install this using rpm. If it conflicts with an existing MySQL installation, you may find the equivalent development package for your installation (if it is not already installed), or uninstall the conflicting version (and any applications depending on it).

The development package will create a */usr/include/mysql* directory and a */usr/lib/mysql* directory with required libraries to link against. Other distributions may install these to different locations.

NOTE **An alternative is to download the source to MYSQL and compile it completely to avoid having an install conflicting version.**

Configuring QT

NOTE **If you already have a 4.6.3 distribution of Qt on your Linux box, or you find a downloadable rpm for Qt 4.6.3, you can skip this section and build the plugin directly as shown in Building the plugin section.**

To configure Qt 4.6.3, perform the following steps:

1. Navigate to http://qt.nokia.com/downloads and click on *Visit our FTP archive* section at the bottom of the page.
2. Under the FTP Index, go to *qt/source* directory.
3. Download any qt-everywhere 4.6.3 package (either in zip or tar), for example, `qt-everywhere-opensource-src-4.6.3.tar.gz` or `qt-everywhere-opensource-src-4.6.3.zip`.
4. Create a clean working directory and unpackage the source code.

   NOTE – **This distribution does not include any binaries so you may have to build it. It is recommended to use the same gcc compiler as required for IC-CAP userc compilation.**

- These instructions compile the minimal distribution that can build the required MySQL plugin. It is applicable on a 64-bit machine as long as 32-bit versions of the X libraries exist and 32-bit versions of other required libraries exist. However, if possible, perform the compilation on a 32-bit machine. IC-CAP requires a 32-bit version of the MySQL plugin.

5. Change the working directory to the unpackaged `qt-everywhere-opensource-src-4.6.3` directory.
6. Execute the following command:

```
./configure -release -prefix-install -prefix ..
/qt_target -platform linux-g++-32 -opensource -no-
largefile -no-accessibility -no-qt3support -no-
xmlpatterns -no-multimedia -no-audio-backend -no-
phonon -no-phonon-backend -no-svg -no-webkit -no-
javascript-jit -no-script -no-scripttools -no-
declarative -no-gif -no-libtiff -no-libpng -no-
libmng -no-libjpeg -no-openssl -no-rpath -no-nis -
no-cups -no-iconv -no-pch -no-dbus -no-separate-
debug-info -no-gtkstyle -no-nas-sound -no-opengl -
no-openvg -no-sm -no-xshape -no-xsync -no-xinerama -
no-xcursor -no-xfixes -no-xrandr -no-xrender -no-
mitshm -no-fontconfig -no-xinput -no-xkb -no-glib
```

7. Accept the displayed `lgpl` license agreement.
8. Now, to make Qt, execute the `gmake` command.

NOTE It is not mandatory to make the install target.

Building the Plugin

After the Qt configuration is complete, you can build the `libqsqlmysql.so` plugin to link MySQl with Database Connection Manager.

To build the plugin, perform the following steps:

1. Set the `QTDIR` variable to the location you unpackaged the `qt-everywhere-opensource-src-4.6.3` package as shown in the following example.

```
export QTDIR=/tmp/ qt-everywhere-opensource-src-4.6.
3
```

2. Execute the following command:

```
$QTDIR/bin/qmake -o Makefile "INCLUDEPATH+=<path to
MYSQL installation>\include" "LIBS+=-L<path to
MYSQL devel libraries > -lmysqlclient_r" mysql.pro
```

If you have installed the MySQL development package, as described previously, then execute the following command:

```
$QTDIR/bin/qmake "INCLUDEPATH+=/usr/include/mysql" "
LIBS+=-L/usr/lib/mysql -lmysqlclient_r" mysql.pro
```

3. Execute the `gmake` command. This step builds the `libqsqlmysql.so` plugin in the *$QTDIR\plugins\sqldrivers* directory.
4. Now, copy the `libqsqlmysql.so` plugin to your *%ICCP_ROOT%\bin\plugins\qt\sqldrivers* directory to be able to use MySQL with Database Connection Manager on Linux.

Now, launch Database Connection Manager from Tools menu in IC-CAP Main window.

You can access the MySQL option from the Select Database Connection Type drop-down list.

This information is subject to change without notice.

www.keysight.com