BEAMER

v4.6.6 Manual

BEAMER Manual

Advancing the Standard in E-Beam Lithography

GenISys GmbH

3/28/2014 3:10 PM

BEAMER v4.6.6 Manual

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: March 2014 in Munich, Germany

Table of Contents

1 Introduction

This manual is intended as a reference for all users of BEAMER. The reader should have some general background with layout data processing, e-beam proximity effect correction and e-beam modeling. It is also assumed that the user is familiar with a window based graphical user interface (GUI) on modern computer platforms.

The baseline for this document is BEAMER 4.6.6 running on a Microsoft Windows operating system; however this manual applies for BEAMER running on a Linux operating system. BEAMER is available in various package and license configurations. This document will discuss the full functionality available in the BEAMER full advanced package.

Basic functionality:

- VisualFLOW^{TM} graphical user interface for drag and drop design of process flows
- Layout Viewer
- Import of layouts in various generic formats
- Export to various generic formats
- Extraction of cells, layers or regions of interest
- Transformation (Scale, Shift, Rotate, Mirror)
- Grid Controls (modify the database grid)
- Controls for splitting and merging of layouts
- Scripting and Loops for automation

Advanced functionality:

- Layout operations (e.g. healing, overlap removal, tone-reversal, bias, pattern XOR)
- Boolean operations (AND, OR, XOR, MINUS)
- E-Beam Proximity Effect Correction
- E-Beam Feature Dose Assignment by layer or data type
- E-Beam modeling (simulation of energy density)
- Resist modeling (simulation of resist contours)
- Metrology (measure a CD at a defined position)
- Export to various E-Beam machines
- Import of layouts in various E-Beam machine formats

1.1 System Requirements

BEAMER can be installed on either a Windows or Linux based platform. As the processing of layout data is typically CPU, memory and hard drive intensive, we suggest using a modern computer with a quad core CPU, adequate RAM and a hard drive with high read and write speeds.

We suggest that your computer has at least
- CPU : 2.5 GHz with 4 cores
- 4 GB RAM
- 500GB HDD

Supported platforms are:
- Microsoft Windows XP, Vista, 7, 8
- Linux : CentOS 4 or higher, RedHat 4 or higher both using kernel version 2.6 or above.

Please inquire about the support of other operating system environments.

1.2 License Overview

GenISys uses the CodeMeter USB hardware key and software rights management solution from WIBU Systems AG (http://wibu.com). It protects the software from unauthorized use. CodeMeter hardware keys are hardware tokens which, when attached to a computer, monitor and enforce the licensing of protected applications. The licensee may only use the programs for which he has paid a license fee (or in the case of an evaluation copy, those programs Licensee is authorized to evaluate) and for which Licensee has received a valid Software Certificate.

The CodeMeter software will be installed on the PC during the first installation of BEAMER. After installation the CodeMeter USB license stick needs to be connected to run BEAMER. The CodeMeter stick is pre-programmed by GenISys. It contains all information: license end date, features which are licensed, number of users, maintenance end date.

Important: Do not try to modify the data on the license stick. Any modification will void the license!

The license stick number and the state can be viewed by opening the "CodeMeter Control Center". You can open the Control Center by double clicking on the CodeMeter icon in the status bar, or by “Start -> Program Files -> CodeMeter -> CodeMeter Control Center”.

The Control Center is also used for updating the license (e.g. extending an expiry date, maintenance date, adding features or additional users) by generating a *. WibuCmRaC file and sending it to GenISys support. GenISys Support will send back a *.WibuCmRaU file, which can be loaded using the CodeMeter Control Center. Opening the WebAdmin (button at lower right of CodeMeter Control Center) you can get detailed information on the content of your license and control settings for the environment (eg. network). Please contact support@genisys-gmbh.com for more information.

The BEAMER software license offers the user the following:

- Multiple PC's by moving the USB license stick
  BEAMER can be installed on multiple PC's and run by connecting the USB license stick
- Multi-Platform – run BEAMER on both Windows and Linux
- Multi-User
  The number of parallel users depends on the purchased license. The standard is one (1). Please inquire about Multi-User license configurations.
- Workstation or Network Access
  The license can be used in a network by having one computer in the network act as license server, other computers in the network act as clients. Please contact GenISys Support for detailed information on setting up network (floating) licenses.

2 Installation

To install BEAMER, an install package is provided that contains runtime binaries, documentation and examples. In this install package are also third party software install packages that are needed to run BEAMER.

The BEAMER install package consists of:

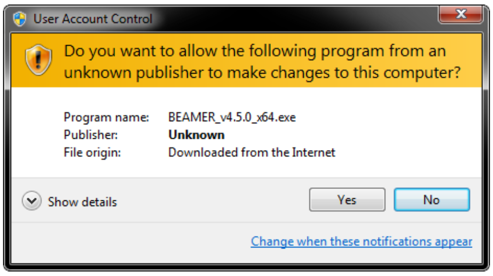
- Documentation (Manual, Release Notes)
- Training videos
- Sample layout and PSF files
- BEAMER runtime files
- CodeMeter license manager

Documentation, sample files and training videos are installed in the GenISys program folder by default. Under Windows-based systems this documentation can be accessed via the Start menu.

2.1 Windows

Download the BEAMER installation executable package and run the installer.

1. Depending on your version of Windows you may encounter the following dialog. Click "Yes" to continue.

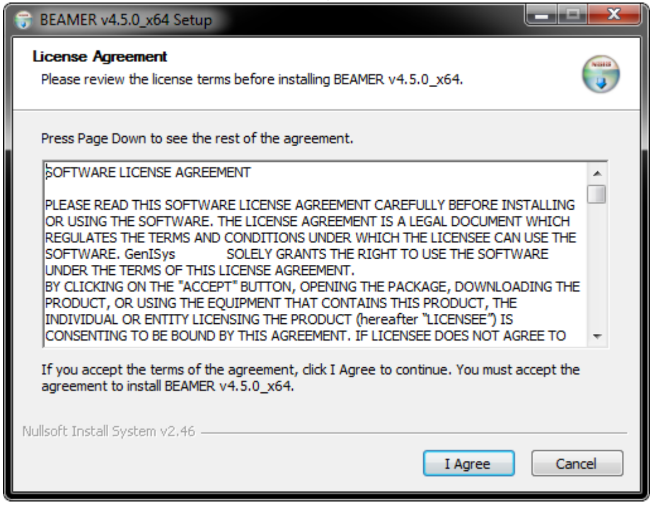


User Account Control

Do you want to allow the following program from an unknown publisher to make changes to this computer?

Program name: BEAMER_v4.5.0_x64.exe
Publisher: **Unknown**
File origin: Downloaded from the Internet

2. Click "Agree" to begin installation if you agree to the terms and conditions stated in the license agreement.Follow the installer instructions using the Back, Next or Cancel buttons.



BEAMER v4.5.0_x64 Setup

License Agreement

Please review the license terms before installing BEAMER v4.5.0_x64.

Press Page Down to see the rest of the agreement.

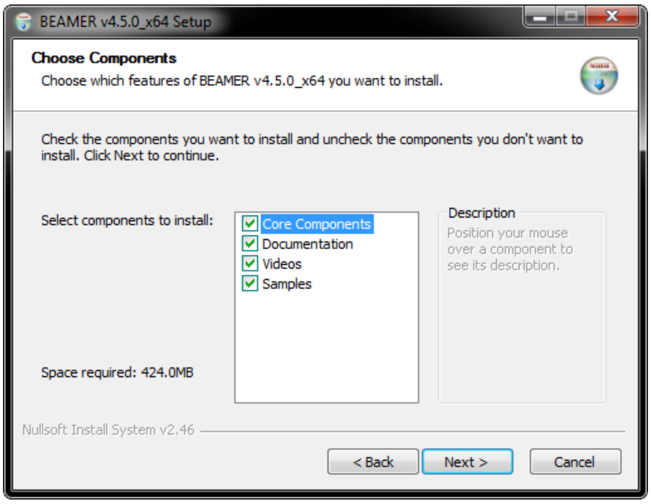
SOFTWARE LICENSE AGREEMENT

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE SOFTWARE. THE LICENSE AGREEMENT IS A LEGAL DOCUMENT WHICH REGULATES THE TERMS AND CONDITIONS UNDER WHICH THE LICENSEE CAN USE THE SOFTWARE. GenISys SOLELY GRANTS THE RIGHT TO USE THE SOFTWARE UNDER THE TERMS OF THIS LICENSE AGREEMENT.
BY CLICKING ON THE "ACCEPT" BUTTON, OPENING THE PACKAGE, DOWNLOADING THE PRODUCT, OR USING THE EQUIPMENT THAT CONTAINS THIS PRODUCT, THE INDIVIDUAL OR ENTITY LICENSING THE PRODUCT (hereafter "LICENSEE") IS CONSENTING TO BE BOUND BY THIS AGREEMENT. IF LICENSEE DOES NOT AGREE TO

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install BEAMER v4.5.0_x64.

Nullsoft Install System v2.46

3. By default, install all the components.



BEAMER v4.5.0_x64 Setup

Choose Components

Choose which features of BEAMER v4.5.0_x64 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

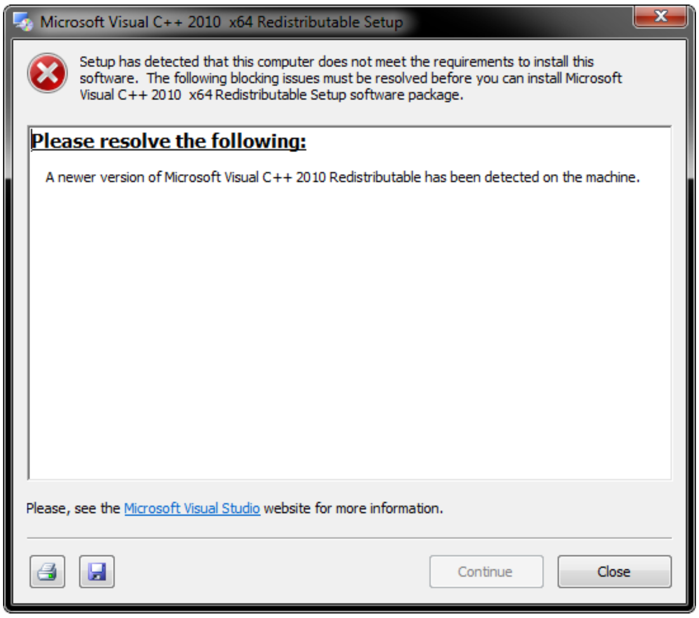
☑ Core Components
☑ Documentation
☑ Videos
☑ Samples

Description

Position your mouse over a component to see its description.

Space required: 424.0MB

4. Follow through the installation prompts for any of the 3rd party software like CodeMeter. To install the license management software program CodeMeter, you must have administration rights.
5. When installing the Microsoft redistributable library, you may encounter the following prompt. Click "Close" to continue.



Microsoft Visual C++ 2010 x64 Redistributable Setup

Setup has detected that this computer does not meet the requirements to install this software. The following blocking issues must be resolved before you can install Microsoft Visual C++ 2010 x64 Redistributable Setup software package.

Please resolve the following:

A newer version of Microsoft Visual C++ 2010 Redistributable has been detected on the machine.

Please, see the Microsoft Visual Studio website for more information.

6. The installation will tell you when it is finished. Click "Close" to exitout of the installation wizard.
7. After successful installation, connect your USB license key to the PC and start BEAMER using the BEAMER icon on your desktop or going through “Start -> Program Files -> BEAMER”

You will find the installation folder of BEAMER under “C:\Program Files\GenISys\BEAMER v4.6.6” (default at installation)

In this directory you will find the sub folders:

- bin : contains the binary files for BEAMER
  - Flow_Library : storage for Central Flow library entries
  - LayoutEditor : contains the binary files for LayoutEditor
  - rc : contains resource files for BEAMER

- share
  - 3rd_party : third party software runtime files for Visual C and CodeMeter
  - doc: this Manual, Training Videos, Release Notes, License agreement
  - examples : Sample layouts (used in the Manual/Tutorial), Sceleton PSF files (some MC simulation files for typical stacks)

For the first time user, it is highly recommended that you watch the training videos.

2.2 Linux

Be sure that you have downloaded the latest version of BEAMER from the GenISys website. If this is your first time installing BEAMER, you will also need to install CodeMeter. CodeMeter can be obtained from the GenISys website through the support page. The CodeMeter binary comes as an RPM package that can be installed by double-clicking the package and allowing the Linux package manager to handle the installation. You should have root access to install the package. Once CodeMeter is installed, follow the instructions below to install BEAMER.

In the instructions below, '$' signifies a command prompt. Anything following is the command you must type and hit the ENTER key. To begin, open a new terminal.

To install Layout BEAMER onto your Linux box, you will need to login as root. Type the following command; it will ask you for a password:

```
$ su
```

We are going install BEAMER in a directory called beamer under */usr/local*:

```
$ cd /usr/local/
$ mkdir beamer
$ cd beamer
```

In the beamer directory, unzip the entire BEAMER installation package. The * indicates the characters that would complete the file name; be sure to type the actual file name in its entirety. The period at the end of the command string is part of the command.

```
$ tar -xzf PATH_TO_LAYOUT_BEAMER_INSTALL_PACKAGE/BEAMER-*.tar.gz .
```

Create a soft link to the folder with the most current version of BEAMER. The * indicates the characters that would complete the directory name; be sure to type the actual directory name in its entirety:

```
$ ln -s BEAMER-*/ current
```

Copy the script to the present working directory. You should still be inside /usr/local/beamer. The period at the end of the command string is part of the command.

```
$ cp current/set_BEAMER_environment.sh .
```

Using your favorite editor, edit the file you just copied. In the command below, we are using the text editor called gedit:

```
$ gedit set_BEAMER_environment.sh
```

In the editor, look for the string `pwd`. Delete this string, including the tick marks and replace it with the following:

```
/usr/local/beamer/current
```

In the same editor, go the PATH variable and add in between the '=' and '$', the following string, including the colon:

```
/usr/local/beamer/current/bin:
```

Log out of root. Depending on how your system is administrated, you will need to add the following line at the end of every .bashrc file for each user:

```
source /usr/local/beamer/set_BEAMER_environment.sh
```

Your installation should be complete. Open a new terminal and run BEAMER:

```
$ BEAMER
```

2.3 Cluster Version

The following are required to run the cluster version of BEAMER:

1. The cluster version should be installed on a location that each node can see
2. The MPICH1 framework must be installed and must be accessible from each node
3. SSH must be installed

On the main node install the cluster version of BEAMER and set the environment. Make sure that the installation can be accessed from all other nodes.
Set the environment on all nodes to include these variables. Use paths that a computational node can see the installation.

```
export GENISYS_BEAMER_HOME=<path of BEAMER installation>
export GENISYS_BEAMER_PATH=${GENISYS_BEAMER_HOME}/bin
export LD_LIBRARY_PATH=${GENISYS_BEAMER_HOME}/lib:${LD_LIBRARY_PATH}
export PATH=${GENISYS_BEAMER_PATH}:${PATH}
export LANG=en_US.UTF-8
```

Launch 'SetupBEAMER' and configure/define the computational nodes.

Execute then the BEAMER cluster version using *Execute BEAMER cluster*.

Visit FILE/Properties and configure the temp directory. It must be visible for all computational nodes!

Design and execute your flows.

2.3.1 Debugging

In case you run into issues executing your cluster version of BEAMER, follow these steps to identify the issue:

1. Install BEAMER.
2. Check if BEAMER runs by executing 'BEAMER' (if this is not working the paths are not set properly).
3. Check if BEAMER runs from another node (if this is not working the paths are not set properly).
4. Install mpi.
5. Check the mpi installation by 'which mpirun'

3 BEAMER Overview

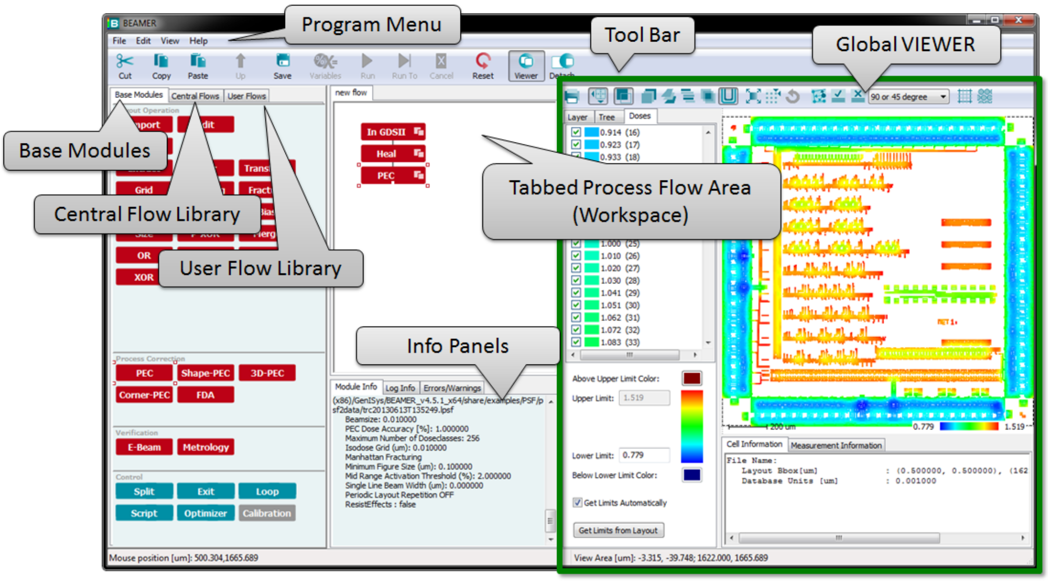
BEAMER is the most comprehensive and fully integrated e-beam lithography software package in use today. It includes proximity effect correction (PEC), e-beam and process modeling, and fracturing to e-beam machine formats. This unique GenISys GmbH solution offers the market's most advanced, easy-to-use, high performance, flexible and fully featured package for achieving the best results on the wafer.

- Comprehensive library with all needed layout processing functions
- Superior process correction including proximity effect, writing strategy and resist effects
- Integrated e-beam modeling including machine shot placement
- Fast and flexible viewing for inspection and verification with the integrated Layout Viewer
- Ease of use with "drag and drop" GUI (VisualFLOW$^{©}$) design, and process flow database management
- Supports all major layout and machine formats (JEOL, Vistec, GDS, TxL, etc.)
- Responsive support and regular updates, with new functionality and enhancements

As a starting point we want to recommend to you to watching the training videos you will find in the documentation folder and on Windows in the Start / Program Files / GenISys / BEAMER / Documentation.

GenISys has developed a unique yet simple approach to dealing with the highly complex issues related to e-beam data preparation, proximity effect correction and simulation. The GenISys VisualFLOW$^{©}$ platform allows the user to visually design work flows in a building block manner, inspect results, make necessary changes on the fly and quickly export results.

The BEAMER interface has the following main components:


Program Menu
Tool Bar
Global VIEWER
Base Modules
Central Flow Library
User Flow Library
Tabbed Process Flow Area (Workspace)
Info Panels
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Import Edit Transf Fract Grid Bias Size P-XOR Merg OR XOR
Process Correction PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification E-Beam Metrology
Control Split Exit Loop Script Optimizer Calibration
new flow
In GDSII Heal PEC
Layer Tree Doses
0.914 (16) 0.923 (17) 0.933 (18) 1.000 (25) 1.010 (26) 1.020 (27) 1.030 (28) 1.041 (29) 1.051 (30) 1.062 (31) 1.072 (32) 1.083 (33)
Above Upper Limit Color: Upper Limit: 1.519 Lower Limit: 0.779 Below Lower Limit Color:
Get Limits Automatically
Get Limits from Layout
90 or 45 degree
200 um
0.779 1.519
Module Info Log Info Errors/Warnings
Cell Information Measurement Information

```
(x86)/GenISys/BEAMER_v4.5.1_x64/share/examples/PSF/p
sf2data/trc20130613T135249.lpsf
    Beamsize: 0.010000
    PEC Dose Accuracy [%]: 1.000000
    Maximum Number of Doseclasses: 256
    Isodose Grid (um): 0.010000
    Manhattan Fracturing
    Minimum Figure Size (um): 0.100000
    Mid Range Activation Threshold (%): 2.000000
    Single Line Beam Width (um): 0.000000
    Periodic Layout Repetition OFF
    ResistEffects : false
```

```
File Name:
   Layout Bbox[um]              : (0.500000, 0.500000), (162
   Database Units [um]          : 0.001000
```

Mouse position [um]: 500.304,1665.689

View Area [um]: -3.315, -39.748; 1622.000, 1665.689

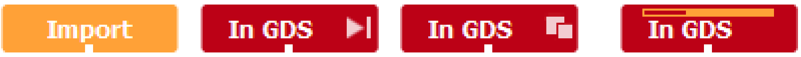
3.1 VisualFLOW

Introduction

BEAMER provides an innovative GUI for the modular design of complex process flows. This new VisualFLOW© platform has been enhanced with features and use concepts that will provide greater process flexibility. Users can quickly create process flows using modules from the Base Modules library or from the user defined Central Flows or User Flows libraries. Building a process flow is done simply by *drag-and-drop* (using the left mouse button) of the desired module from the library into the Process Flow Design Area window, or by double clicking on the desired module to automatically connect that module to a selected module in the Process Flow Design Workspace.

Module Status

The status of a module is shown by the module color and the icon marker located at the right side of the module. A module which has not had its parameters set is identified as an orange colored module, whereas a red color and a triangular-shaped Run To button signifies that parameters have been set but the module has not been run yet. A module which has been run and the results are ready, is shown with an overlapping-box View icon. A progress bar is shown while the process is being computed . If a module has been selected with the mouse or with the Ctrl-A command, it will have four red dots surrounding the module. Moving your mouse over a module will display the module name and any comments that have been assigned to that module.

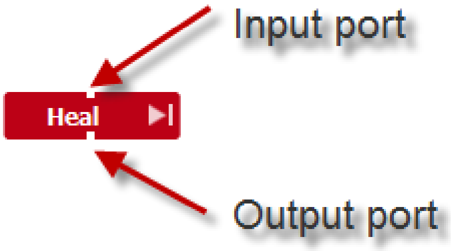


Single Port Modules

A port is used to connect modules. There are input and output ports. A single port module has only one port at the bottom center of the module signifying that it has only output connectivity capability. It can be connected to another module, but has no input capability within the flow. A module's ports are identified as a small, white-colored box.

Dual Port Modules

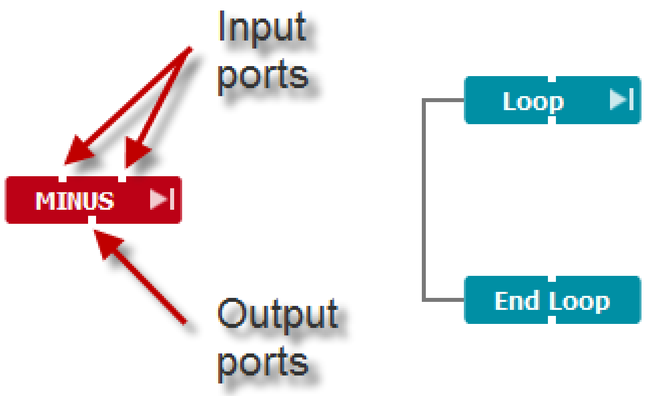
A dual port module has a connectivity port at both the top and bottom center of the module, signifying that it has both input and output connection capability in a flow.

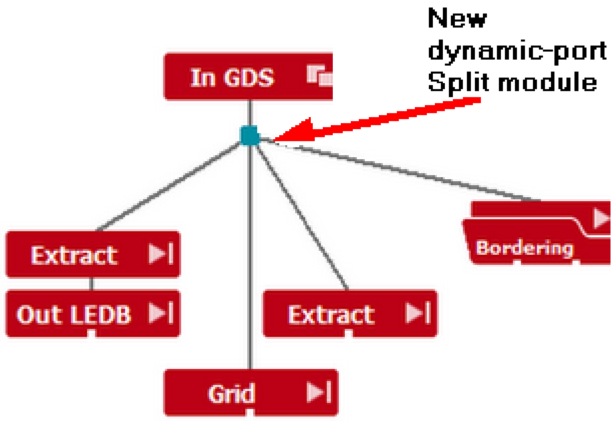


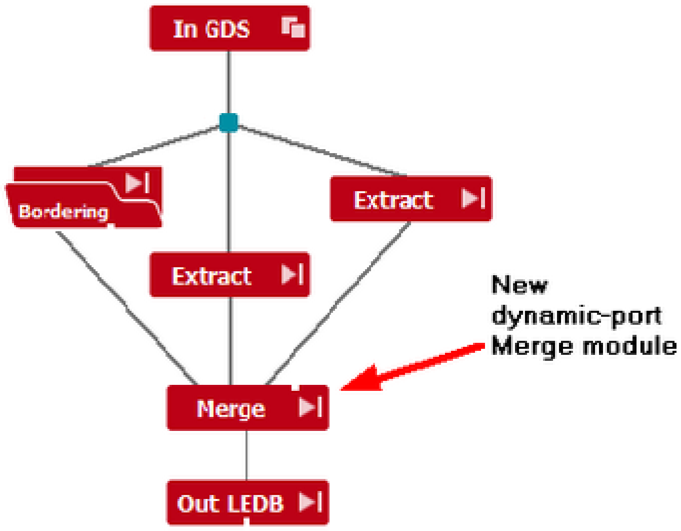
Dual port modules include: Extract, Filter, Transform, Grid, Heal, NOT, Bias, P-XOR, PEC, Shape-PEC, 3D PEC, FDA, Export, E-Beam and Metrology.

Multi Port Modules

A multi port module usually has three ports with a few exceptions. The multi port modules have two input ports at the top on the module and one output port on the bottom. Some specialty modules will have one input and two output modules while others, such as Loop and Optimizer, will combine two modules as a single process, each with input and output connectivity capability. The other exception is the new Merge and Split modules which transform into a hub with a dynamic number of ports.

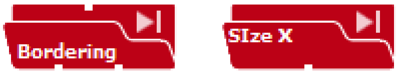






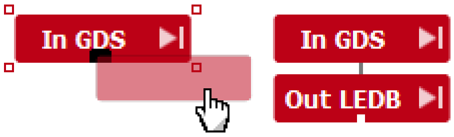
Multi port modules include: OR, MINUS, AND, XOR, Split, Merge, Loop and Optimizer.

Database Modules
A database module is a module that has predefined parameters set by the user and saved into the Central Flows or User Flows database. Database modules are identified by a folder-shaped icon, and share the same characteristics as the Base Modules described above.



How To Connect Modules

There are three ways to connect modules. Drag and drop a module from the library onto the target module in the flow until the small white connection port turns black, then release the mouse and the modules will be connected. Modules can also be manually connected by using the right mouse button, held down, to draw a line between open ports. Finally, double clicking a module in any library will automatically send it to the Process Flow Design window, and connect it to a *selected* module in the flow.

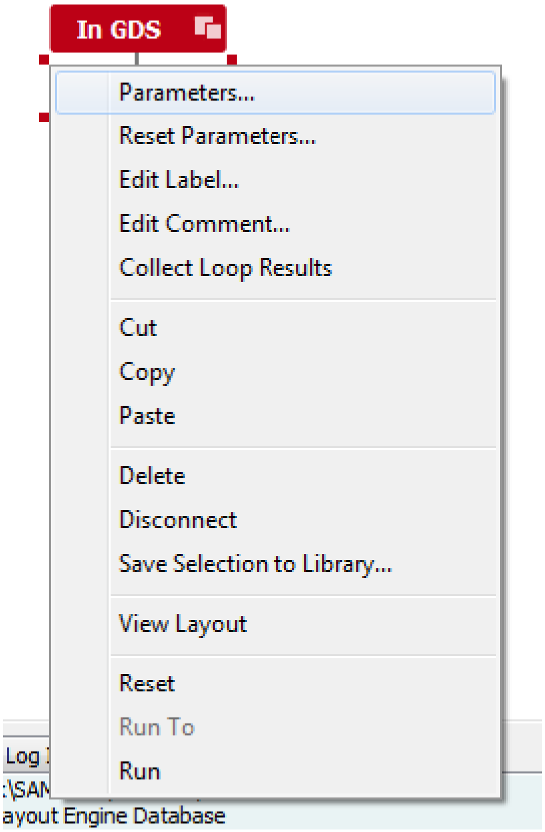


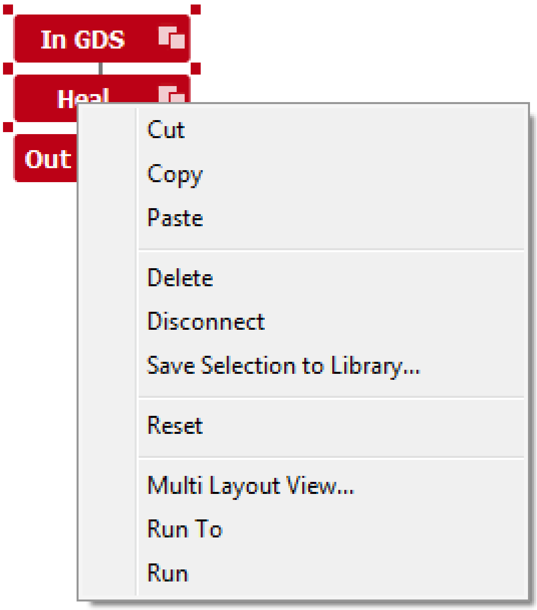
Module Parameter Settings

Each module has a parameter window associated with it. When building a process flow, individual module parameter windows will become available when they have been dropped into the Process Flow Design window. Some parameter windows do not open automatically but double clicking a module will open its parameter window.

3.2 Context Menus

Each module has a popup context menu where additional options and operations can be selected. To open the context menu, right click on the module. The context menu options will differ between the selection of a single module and selection of multiple modules.





Parameters...
Opens the parameter dialog box of the module should one exist.

Reset Parameters...
Resets the parameters to the default settings for that module.

Edit Label...
Opens a dialog box to rename the module to a unique name.

Edit Comment...
Opens a dialog box where the user can input comments. The comment is made visible with a mouse-over.

Collect Loop Results
Enabling this option will save Loop results of parameterized runs.

Cut, Copy, Paste
The action will be applied to the selected module(s). The keyboard short cuts (Ctrl-X, Ctrl-C and Ctrl-V) are also available.

Delete
The selected module(s) will be removed from the flow.

Disconnect
The selected module(s) will be disconnected from the flow but not deleted from the Process Flow Design window. It can then be reconnected by the methods described above.

Save Selection to Library...

Any selected modules and their settings will saved to the Central Flows or User Flows database library. A dialog box pops up giving the option to provide a unique name and library location with a relative or absolute path.

View Layout

Opens the Layout VIEWER program to view the results of the selected module.

Reset

The module status will be reset from Complete (green check) to Standby (yellow dot). This will reset subsequent modules in the process flow. Modules prior to the selected module will not be reset.

Run To

Executes up to the selected module.

Run

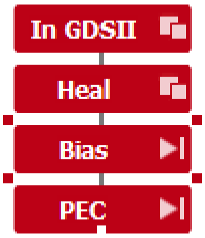
Executes all modules in the current flow.

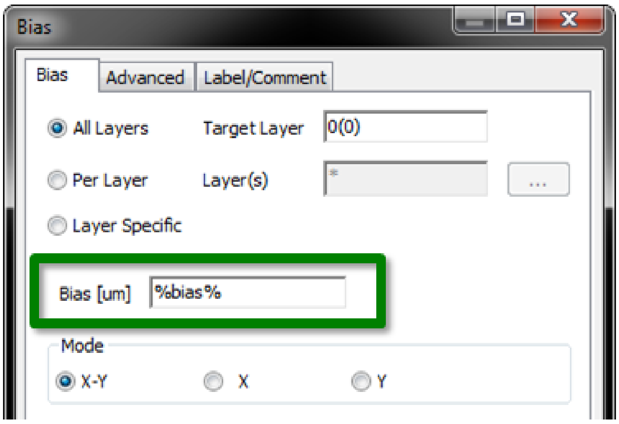
Multi Layout View...

To view the results of two or more modules in a single overlapping view, select the target modules using the mouse while holding down the Ctrl-key, then right click.

3.3 Variables

Variables are a convenient way of changing module parameters to understand the impact of changing values. Most fields in the base modules can take a variable. Variables are handled at a global and local scope and are defined by an alpha-numeric string beginning and ending in with percent (%) symbols. The scope of a variable depends on its usage.



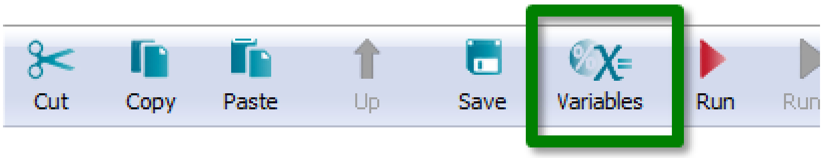


● All Layers Target Layer 0(0)
○ Per Layer Layer(s) *
○ Layer Specific

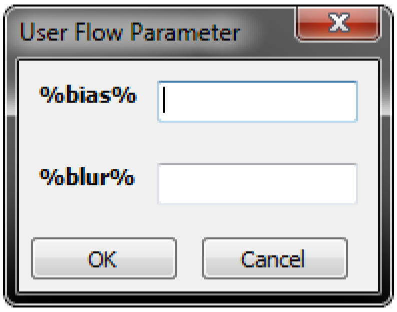
Bias [um] %bias%

Mode: ● X-Y ○ X ○ Y

For example, in the Bias dialog above right, the variable %bias% in used in the **Bias [um]** field. When this variable is written in this field and the module dialog is closed, the variable is recognized as a global variable. As a result, the **Variables** button in the tool bar is enabled. This button is for global variables in BEAMER.



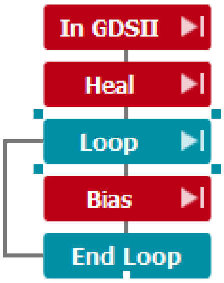
When clicking this button, a dialog will appear allowing the input of the %bias% variable value. Any other global variables will be displayed in the dialog for editing. Global variables are extremely helpful especially if there are multiple modules in a flow that require the same parameter value but instead of editing every module needing this value, a global variable in the field of interest will take care of this task.

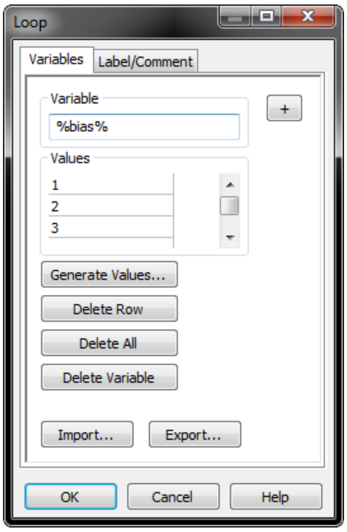


%bias%
%blur%

Variables can have local scope. Local scope is managed automatically when using loops. For instance, a simple loop applying different bias values on a pattern can be done like so:

1. Generate a flow similar to the one below. Double-click the Loop module and add the %bias% variable as shown to the right.

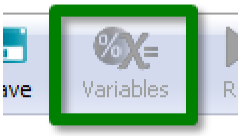




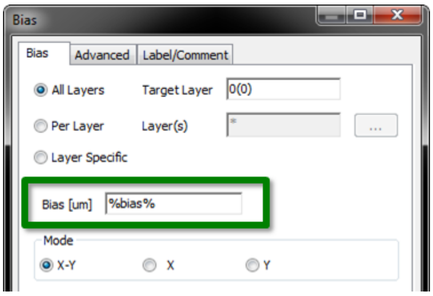
Variable: %bias%

Values:
1
2
3

You should notice at this point the Variables button remains disabled since the variable is considered to have local scope within the loop:



2. Add the %bias% variable to the Bias module



● All Layers Target Layer 0(0)

○ Per Layer Layer(s) *

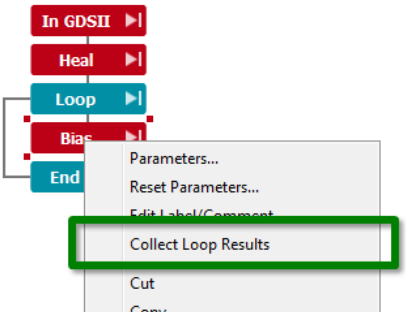
○ Layer Specific

Bias [um] %bias%

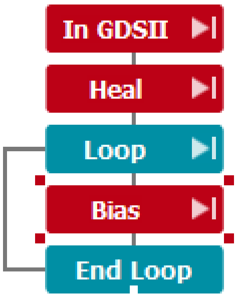
Mode

● X-Y ○ X ○ Y

3. Right-click on the Bias module and select **Collect Loop Results...** .

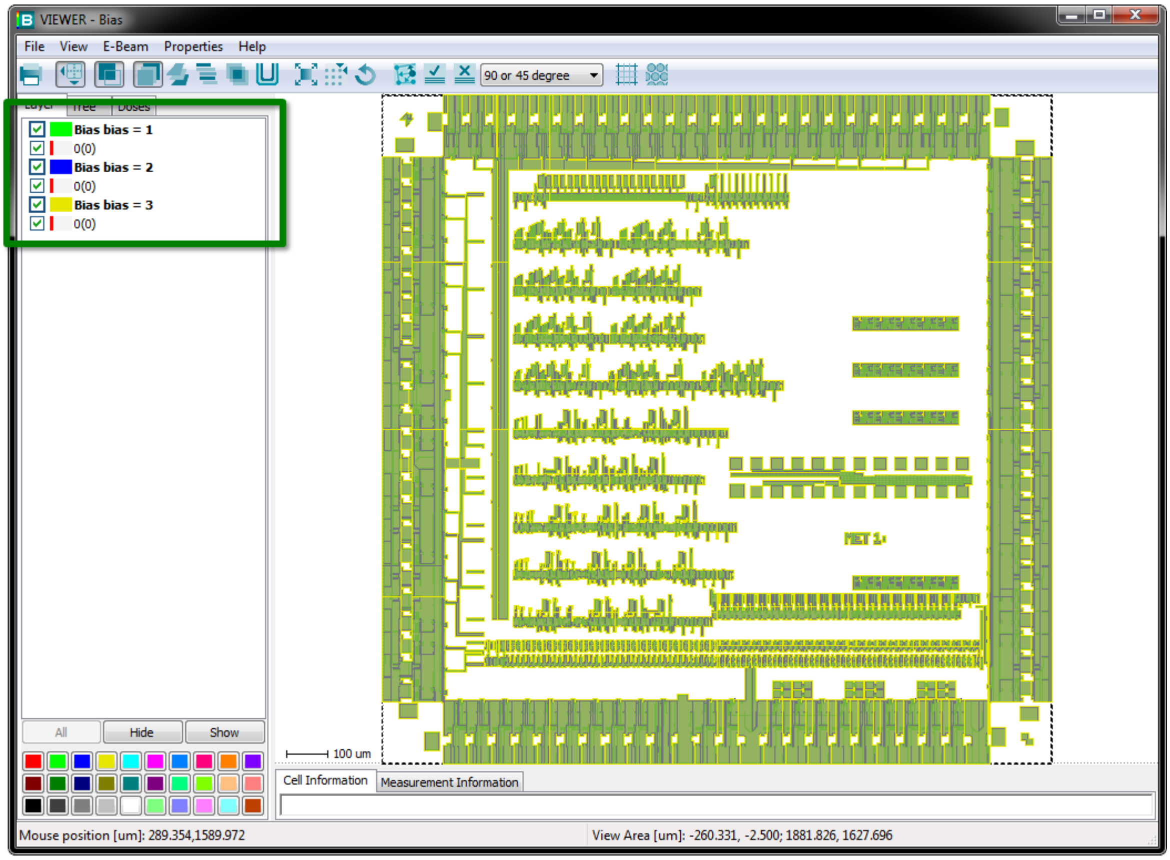


4. Now with one of the modules selected, click the Run button in the tool bar.



5. When completed click on the icon on the Bias module.

6. In the VIEWER, you should find that three different layouts exists for every bias applied in the loop. You can zoom in for inspection.



☑ 🟩 Bias bias = 1
☑ 0(0)
☑ 🟦 Bias bias = 2
☑ 0(0)
☑ 🟨 Bias bias = 3
☑ 0(0)

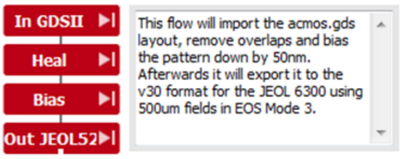
Mouse position [um]: 289.354,1589.972

View Area [um]: -260.331, -2.500; 1881.826, 1627.696

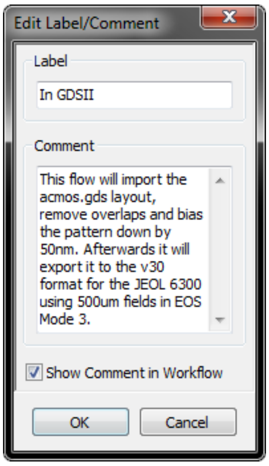
Variables are also used in Database modules. Please refer to the Database Modules section on how to use variables with Database modules.

3.4 Comments

Comment windows are inline/resizable text boxes that can be used to provide clarity especially for complex flows. Comments can be enabled/ disabled at the module and global level.



This flow will import the acmos.gds layout, remove overlaps and bias the pattern down by 50nm. Afterwards it will export it to the v30 format for the JEOL 6300 using 500um fields in EOS Mode 3.

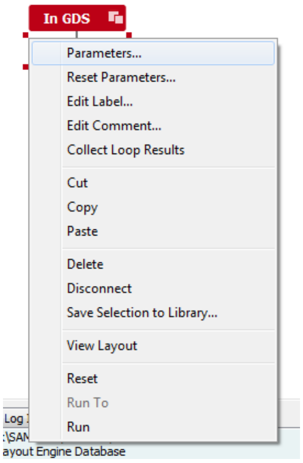


Label: In GDSII

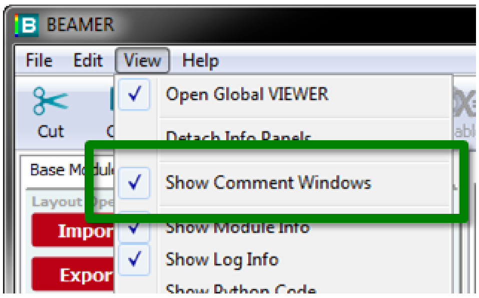
Comment: This flow will import the acmos.gds layout, remove overlaps and bias the pattern down by 50nm. Afterwards it will export it to the v30 format for the JEOL 6300 using 500um fields in EOS Mode 3.

☑ Show Comment in Workflow

At the module level, comments are shown with by checking the Show Comment in Workflow checkbox in the Label/Comment tab in the module dialog. The Label/Comment dialog (seen left) can be accessed directly via the Context Menu by right-clicking on the module and selecting either **Edit Label...** or **Edit Comment...** as illustrated to the right.

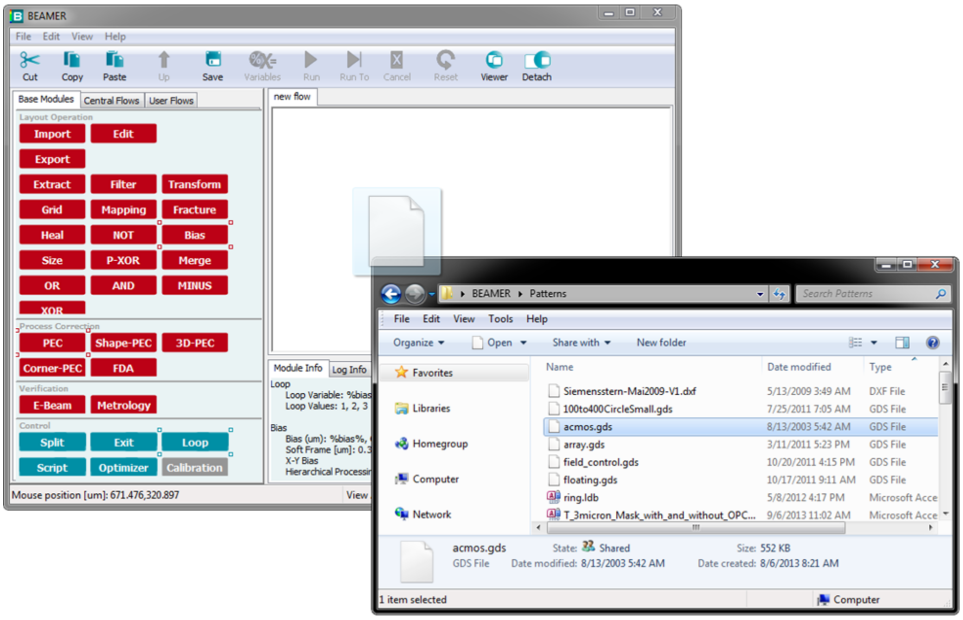


At the global level, all the comments that are visible at the module level can be toggled on or off by selecting **Show Comment Windows** in the View menu. This allows users to turn of all visible comment windows in the workspace to optimize their view.



3.5 Drag and Drop

Pattern Files and Flows can be opened by directly dropping them onto the BEAMER GUI. Dragging and dropping a flow or multiple flows will open a new tabbed workspace or tabbed workspaces. Pattern files in a recognized format will automatically add an Import module(s) to the workspace.



Loop
Loop Variable: %bias
Loop Values: 1, 2, 3

Bias
Bias (um): %bias%,
Soft Frame [um]: 0.3
X-Y Bias
Hierarchical Processi

Mouse position [um]: 671.476,320.897

Name	Date modified	Type
Siemensstern-Mai2009-V1.dxf	5/13/2009 3:49 AM	DXF File
100to400CircleSmall.gds	7/25/2011 7:05 AM	GDS File
acmos.gds	8/13/2003 5:42 AM	GDS File
array.gds	3/11/2011 5:23 PM	GDS File
field_control.gds	10/20/2011 4:15 PM	GDS File
floating.gds	10/17/2011 9:11 AM	GDS File
ring.ldb	5/8/2012 4:17 PM	Microsoft Acce
T_3micron_Mask_with_and_without_OPC...	9/6/2013 11:02 AM	Microsoft Acce

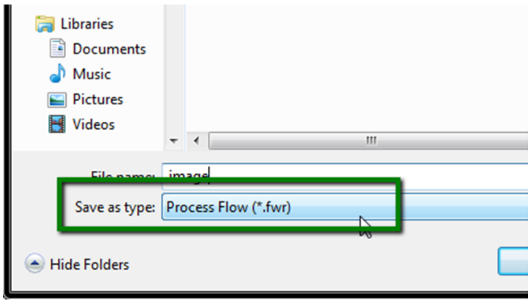
acmos.gds
GDS File
State: Shared
Date modified: 8/13/2003 5:42 AM
Size: 552 KB
Date created: 8/6/2013 8:21 AM

This funtionality works on both the Windows and Linux platforms.

3.6 Saving Executed Flows with Results

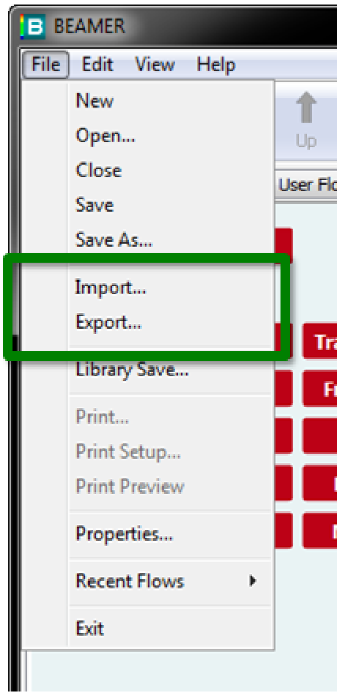
BEAMER allows the users to export the flow and the results of the executed flow into the *.fwr format. The *.fwr format can help you save time by running computationally intensively flows and saving the results for later.

To save a flow with results (fwr), click **Export...** in the File menu and select the *.fwr format in the file dialog as shown below.



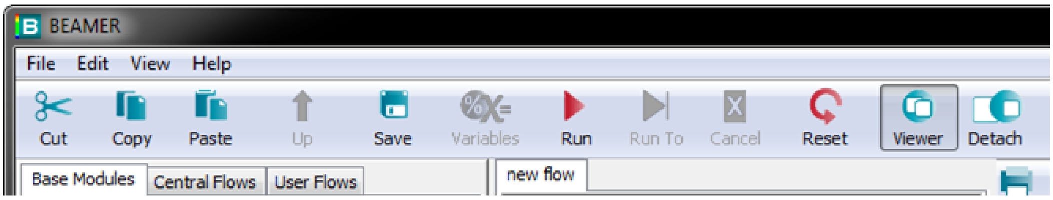
File name: image

Save as type: Process Flow (*.fwr)



3.7 Program Menu and Tool Bar

The program menu and toolbar features are described in this section. Immediate access to the most commonly used functions in the Program Menu are provided in the Tool Bar, whose buttons are defined here. The Program Menus are defined in their respective sub-sections.



Cut, Copy, Paste

Cut, copy and paste commands can be found in the Edit menu, the tool bar and the individual module context menus. You may manipulate the currently selected module(s) or entire flows. Keyboard shortcuts, Ctrl-X, Ctrl-C, and Ctrl-V are available on the Windows platform.

Up

Up is used to go up to the root process flow from within a Database module.

Variables

Quickly access global variables present in the current flow to inspect/edit their values. Please refer to the section on Variables to learn more about using global or local variables in BEAMER.

Save

Saves the current process flow in the Process Flow Design workspace to file. This feature can be quickly accessed via the toolbar.

Run

If a module in a process flow is selected, clicking Run will execute the entire flow to completion.

Run To

If any module in a process flow is selected, clicking Run To will execute only up to the selected module.

Cancel

This button cancels a process flow in execution.

Reset

Reset will simply reset a selected module. To reset an entire flow, be sure the topmost module is selected and then press Reset.

Viewer

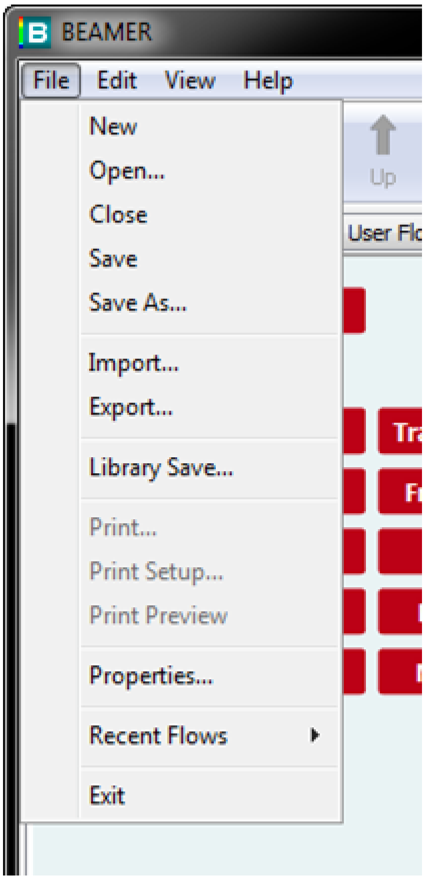
This button will launch the Global Layout Viewer. If Detach is disabled (default), the viewer will appear inline to the main GUI. Otherwise, if enabled, the Global Viewer will appear in a separate window.

Detach

If Detach is disabled (default), the viewer will appear inline to the main GUI. Otherwise, if enabled, the Global Viewer will appear in a separate window.

3.7.1 File Menu

The **File Menu** has the following options:



New

New creates a new workspace tab to build a new BEAMER process flow.

Open...

Opens a saved BEAMER *.ftxt* file, a process flow or set of process flows with defined parameters for data preparation and/or e-beam simulation. BEAMER can also open CATS© CINC files. See CINC to FTXT conversion for more details.

Close

Closes the current process flow.

Save

Saves the current process flow in the Process Flow Design workspace to file. This feature can be quickly accessed via the toolbar.

Save As...

Saves the current process flow and its defined parameters, with a new name. When a file is saved, and a file name given by the user, the new file name is displayed in the blue BEAMER application window, just above the Tool bar. The saved file can be loaded back into BEAMER by selecting Open from the File Menu.

Import...

Imports a process flow with results in the *.fwr format. This format when imported will restore a previously ran flow that was export with the results intact, hence the name "flow with results" or fwr.

Export...

Exports a process flow with results in the *.fwr format. This format save a previously ran flow with the results intact per module, hence the name "flow with results" or fwr.

Library Save...

Saves the current process flow and its defined parameters as a module in the BEAMER database. The flow can be either a single module or a complicated design flow consisting of many modules. The saved flow of module(s) is stored in the database as a Central Flow, User Flow or any other created database tab. When saving into the library, the handling of paths can be set to relative or absolute paths to enhance compatibility. Storing predefined modules or flows in the database is useful for the later building of new process flows, or for their reuse in the current session or later sessions.

Print.../Print Setup.../Print Preview

The Print functionality is disabled in the current version.

Properties...

Provides access to the dialog for BEAMER default settings. See the next section on Properties for more details.

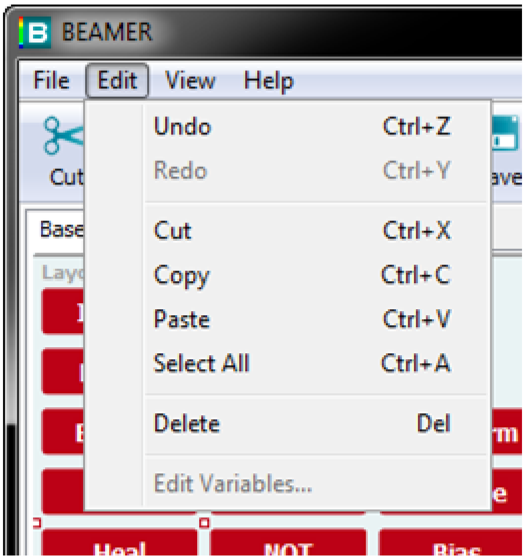
Recent Flows

Provides instant access to recently used BEAMER *.ftxt* flows.

Exit

Closes the current BEAMER session. A dialog window will pop up and ask if you would like to save the current flow before exiting the program.

3.7.2 Edit Menu



Undo, Redo

Allows the undo/redo (Ctrl-Z/Ctrl-Y) of module placement and module parameter modifications. With undo/redo, one can edit flows effectively with the ability to restore recently deleted modules in a session and to undo modified module parameters.

Cut, Copy, Paste

Cut, copy and paste commands can be found in the Edit menu, the tool bar and the individual module context menus. You may manipulate the currently selected module(s) or entire flows. Keyboard shortcuts, Ctrl-X, Ctrl-C, and Ctrl-V are available on the Windows platform.

Select All

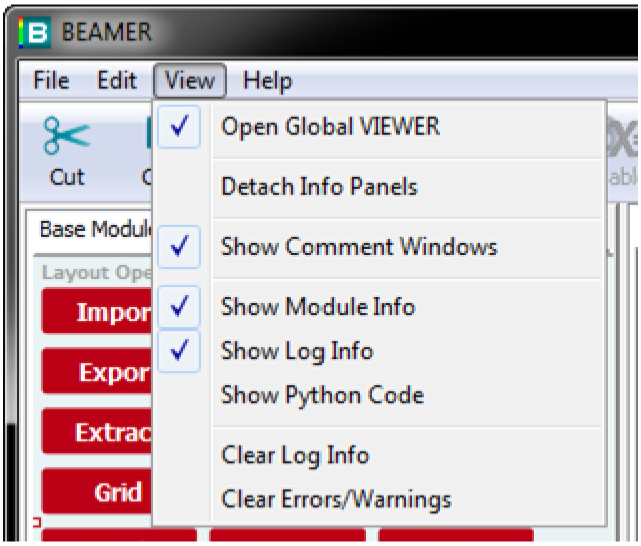
The Select All feature under the Edit menu or with the standard keyboard shortcut (Ctrl-A) will select all modules in the design flow window. The mouse can also be used to make a selection of all modules by drawing a box around the intended modules.

Delete

The Delete feature under the Edit menu has the same functionality as the Tool bar icon Cut which will remove any currently selected module(s) from the design flow window. This functionality is also available from each module's context menu.

3.7.3 View Menu

The View menu allows the user to control the BEAMER interface environment from turning on the inline viewer to disabling comments from view to detaching the info panels for more workspace real estate. In short, the GUI is highly configurable to optimize productivity.

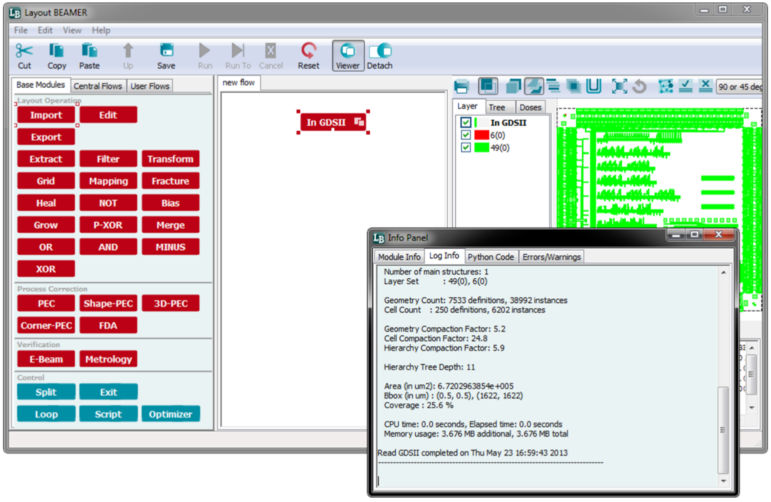


Open Global VIEWER

When toggled on, the Global VIEWER is shown inline with the BEAMER GUI. By toggling off may help optimize the workspace area for larger flows if you have limited resolution on your monitor.

Detach Info Panels

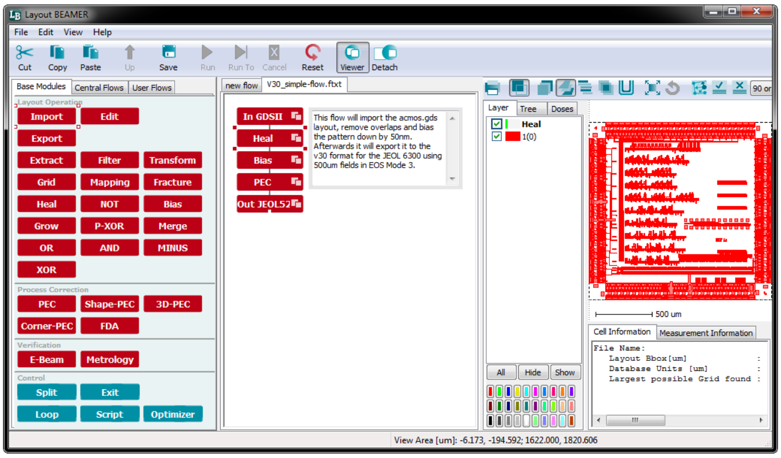
Detaches the Info Panels from the main GUI, placing them in a separate window. By detaching the Info Panels the workspace size increases vertically.



```
 Number of main structures: 1
 Layer Set         : 49(0), 6(0)

 Geometry Count: 7533 definitions, 38992 instances
 Cell Count    : 250 definitions, 6202 instances

 Geometry Compaction Factor: 5.2
 Cell Compaction Factor: 24.8
 Hierarchy Compaction Factor: 5.9

 Hierarchy Tree Depth: 11

 Area (in um2): 6.7202963854e+005
 Bbox (in um) : (0.5, 0.5), (1622, 1622)
 Coverage : 25.6 %

 CPU time: 0.0 seconds, Elapsed time: 0.0 seconds
 Memory usage: 3.676 MB additional, 3.676 MB total

Read GDSII completed on Thu May 23 16:59:43 2013
--------------------------------------------------------------------------------
```

Show Comment Windows

Globally toggles on or off the comment windows next to each module whose comments have been enabled.


Layout BEAMER
File Edit View Help
Cut Copy Paste Up Save Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Grow P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer
new flow V30_simple-flow.ftxt
In GDSII Heal Bias PEC Out JEOL52
Layer Tree Doses
Heal
1(0)
All Hide Show
500 um
Cell Information Measurement Information

This flow will import the acmos.gds layout, remove overlaps and bias the pattern down by 50nm. Afterwards it will export it to the v30 format for the JEOL 6300 using 500um fields in EOS Mode 3.

```
File Name:
   Layout Bbox[um]            :
   Database Units [um]        :
   Largest possible Grid found :
```

View Area [um]: -6.173, -194.592; 1622.000, 1820.606

Show Module Info

Toggles the Module Info tab in the Info Panels.

Show Log Info

Toggles the Log Info tab in the Info Panels.

Show Python Code

Toggles the Python Code tab in the Info Panels.

Clear Log Info

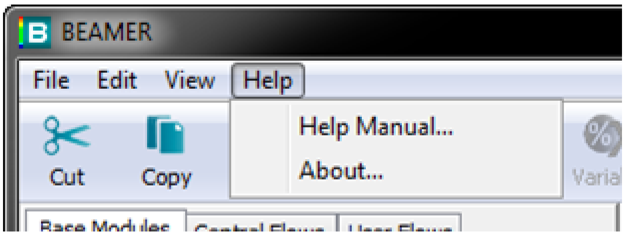
Clears the Log Info panel on command.

Clear Errors/Warnings

Clears the Errors/Warnings panel on command.

3.7.4 Help Menu

The Help menu provides access to this manual and the version information of BEAMER.

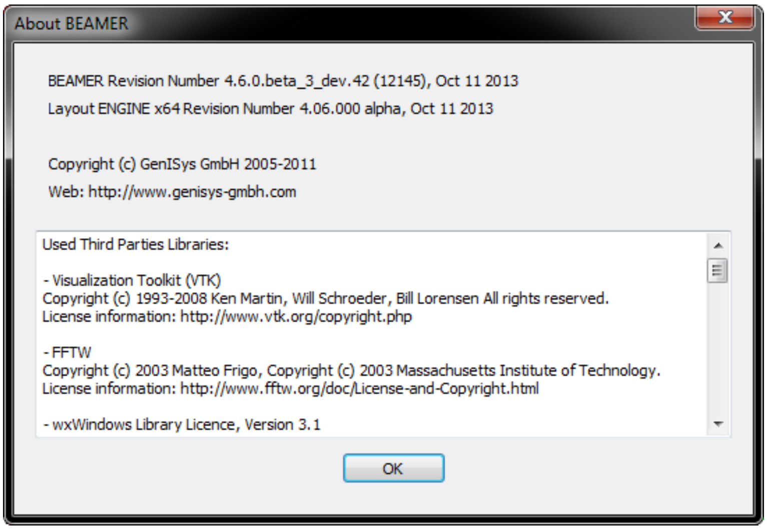


Help Manual...

Opens this Help Manual.

About..

Opens a dialog that shows the version of BEAMER you are running. An example is shown below.



BEAMER Revision Number 4.6.0.beta_3_dev.42 (12145), Oct 11 2013
Layout ENGINE x64 Revision Number 4.06.000 alpha, Oct 11 2013

Copyright (c) GenISys GmbH 2005-2011
Web: http://www.genisys-gmbh.com

Used Third Parties Libraries:

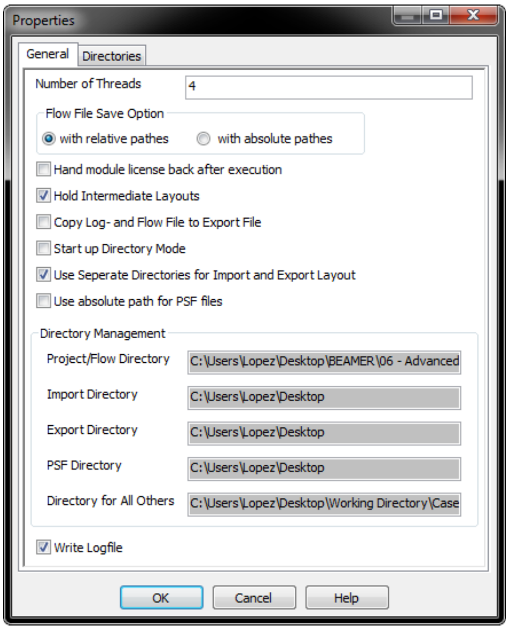
- Visualization Toolkit (VTK)
Copyright (c) 1993-2008 Ken Martin, Will Schroeder, Bill Lorensen All rights reserved.
License information: http://www.vtk.org/copyright.php

- FFTW
Copyright (c) 2003 Matteo Frigo, Copyright (c) 2003 Massachusetts Institute of Technology.
License information: http://www.fftw.org/doc/License-and-Copyright.html

- wxWindows Library Licence, Version 3.1

3.8 Properties

The properties of BEAMER can be accessed via the File Menu.



Number of Threads: 4

Flow File Save Option

● with relative pathes ○ with absolute pathes

□ Hand module license back after execution

☑ Hold Intermediate Layouts

□ Copy Log- and Flow File to Export File

□ Start up Directory Mode

☑ Use Seperate Directories for Import and Export Layout

□ Use absolute path for PSF files

Directory Management

Project/Flow Directory	C:\Users\Lopez\Desktop\BEAMER\06 - Advanced
Import Directory	C:\Users\Lopez\Desktop
Export Directory	C:\Users\Lopez\Desktop
PSF Directory	C:\Users\Lopez\Desktop
Directory for All Others	C:\Users\Lopez\Desktop\Working Directory\Case

☑ Write Logfile

Number of Threads

BEAMER can work with multiple threads in a growing number of modules. This parameter controls how many threads should be used.

Flow File Save Option

When saving a process flow, the default option (selected in the image) will save the reference to any pattern files or PSFs using a relative path. For example, a flow in the directory "*C:\BEAMER\flows\*" imports a GDSII file, *acmos.gds*, from the same directory as the flow. The relative path used to reference the pattern is "*./ acmos.gds*" and is placed in the *.ftxt* file. If an absolute path is used, the path used to reference the pattern will be "*C: \BEAMER\flows\acmos.gds*".

Hand module license back after execution

This option allows to control the check-in and checkout of licenses for each module. This is advantageous for installations with multiple users, where some features (e.g. PEC) are limited to certain number of users at a given time. Enabling this feature will allow BEAMER to return the license to the license server after executing a module so that it can be free to use by another user. Disabling this option (default) allows to keep the license for automated scripts or loops.

Hold Intermediate Layouts

Controls whether intermediate module results are kept in memory. By default, the setting is enabled allowing all modules to store their results in memory to be available to the user. The results of all modules in a user flow can be viewed and parameters can be changed for re-calculation. If this option is disabled, only the result of the last executed module will be available. When disabled, it will not be possible to change parameters of a module for re-calculation. When using large files, **disabling** this option will free up memory.

Copy Log- and Flow File to Export File

Enabled, this feature will always place a copy of the flow and the log file in the same directory of the exported file. Ultimately, it allows a user to always keep the same flow file and output log that was used to generate the exported file for future reference.

Start up Directory Mode

By default the BEAMER working directory is the previous working directory, and the directories listed in the Directory Management section. This option forces BEAMER to use the directory it was started from. This feature is mainly intended for Linux users who launch BEAMER from the command line.

Use Separate Directories for Import and Export Layout

This feature manages the initial directories for the Import and Export modules. With this feature disabled, the directory of the imported file will be used as the starting directory of a newly attached Export module in the initial file dialog. With this feature enabled. the starting directory of the Export module will be different than the directory for the Import module. Depending on your work style, you may always want to export the file in the same directory as the imported pattern. If so, disable this feature. If you routine just export to one directory but import from a different directory, enable this feature.

Use absolute path for PSF files

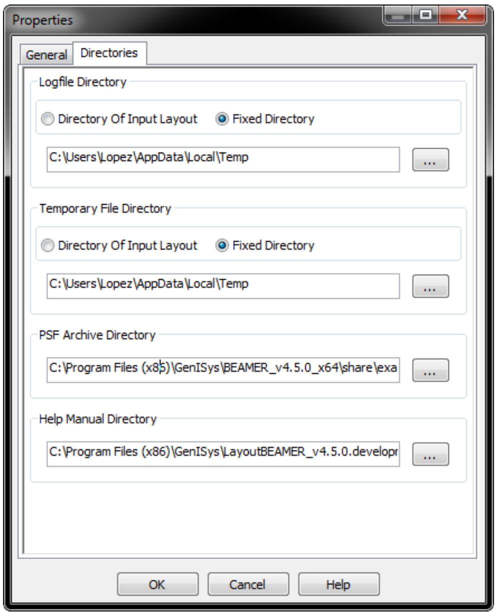
In the PEC module, PSF files can be referenced using a relative path or and absolute path. A relative path maintains the relative location between the flow file and the PSF file. An absolute path maintains the absolute location of the PSF. This is helpful especially when the flow file is moved around in the file system, which means the relative location between the flow and the PSF file is changed as a result.

Directory Management

BEAMER tracks different directories when users interact with the software. This table shows what directories are the starting directories for specific file functions in BEAMER. For example, when the option **Use Separate Directories for Import and Export Layout** is enabled, the starting directory of an Import module can be different than the starting directory of the Export module. The Project/Flow Directory is the location where the flows were recently saved and will be saved when a subsequent flow is saved unless the user specifies a new directory. Import and Export directories indicate the source and target directories for the layouts, respectively. These are changed when the user selects a new directory from/to which to import/export, respectively. The PSF directory is the source where point spread functions (PSFs) were last found and will opened the next time a user looks for a PSF. This will change if the user loads a PSF from another directory. This action results in the new directory to be the starting directory for PSFs in the PEC and E-beam modules in a subsequent file search from these modules. Modules that have a file selection capability will use as source the *Directory for All Others*.

Write Logfile

Allows to save the log-file of each run if switched on.



Logfile Directory
○ Directory Of Input Layout ● Fixed Directory
C:\Users\Lopez\AppData\Local\Temp

Temporary File Directory
○ Directory Of Input Layout ● Fixed Directory
C:\Users\Lopez\AppData\Local\Temp

PSF Archive Directory
C:\Program Files (x86)\GenISys\BEAMER_v4.5.0_x64\share\exa

Help Manual Directory
C:\Program Files (x86)\GenISys\LayoutBEAMER_v4.5.0.developr

Logfile Directory

This option controls the directory for the log file when the option **Write Logfile** is checked on the General Tab.

Temporary File Directory

This option controls the location for the temporary files generated when running a flow. The temporary files are removed if BEAMER closes properly. If the BEAMER crashes, the temporary files will remain in the specified directory.

PSF Archive Directory

This option controls the location for the PSF archive generated by TRACER. When newer versions of BEAMER are installed, the location of the PSF archive is maintained so that the TRACER archive is always used.

Help Manual Directory

The location of the BEAMER help manual is referenced here. This option is used to maintain the fixed location of the user help manual.

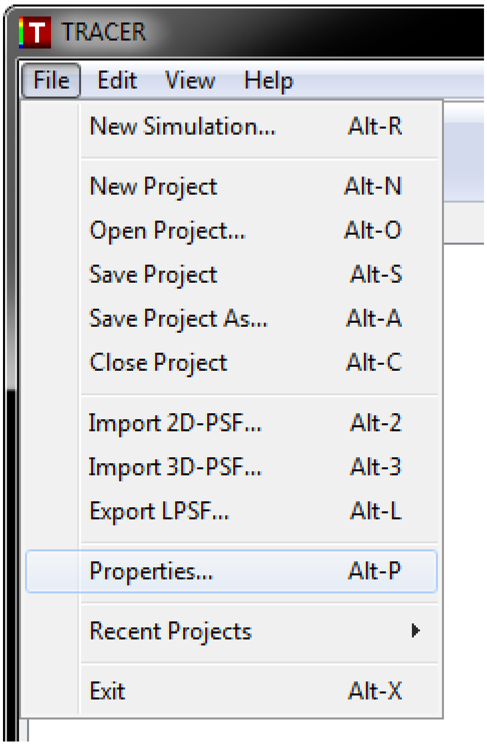
Default Browser [Linux only]

The location of the browser to use by default to load the help manual.

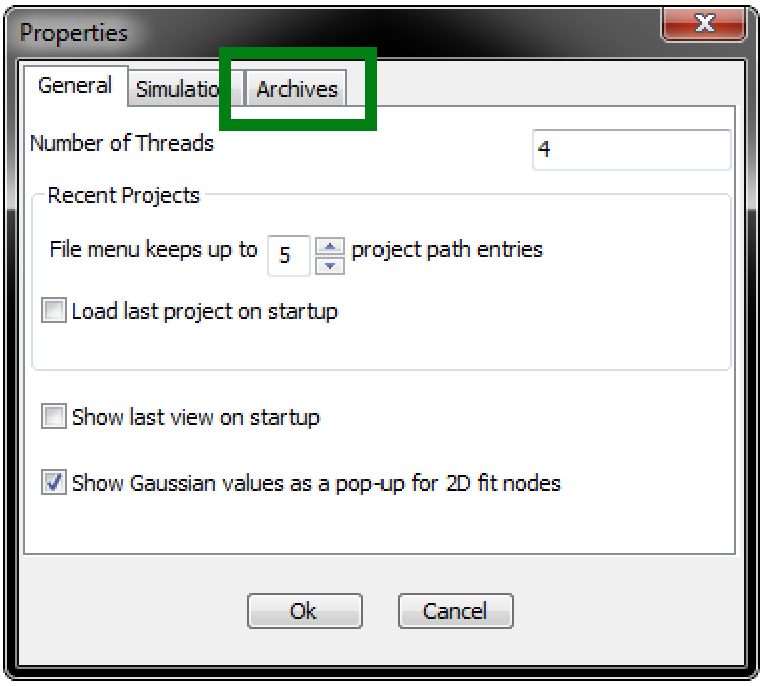
3.9 Synchronizing to the TRACER PSF Archive

If you have acquired TRACER, it is important to synchronize BEAMER to the active TRACER 2D Archive. To do this follow the instructions below.

By default TRACER should have a default directory for the PSF archive. This can be found in **Properties...** under the File menu.



From there you should get a dialog box that looks like the following. Click on the Archives Tab which is highlighted:



Number of Threads 4

Recent Projects

File menu keeps up to 5 project path entries

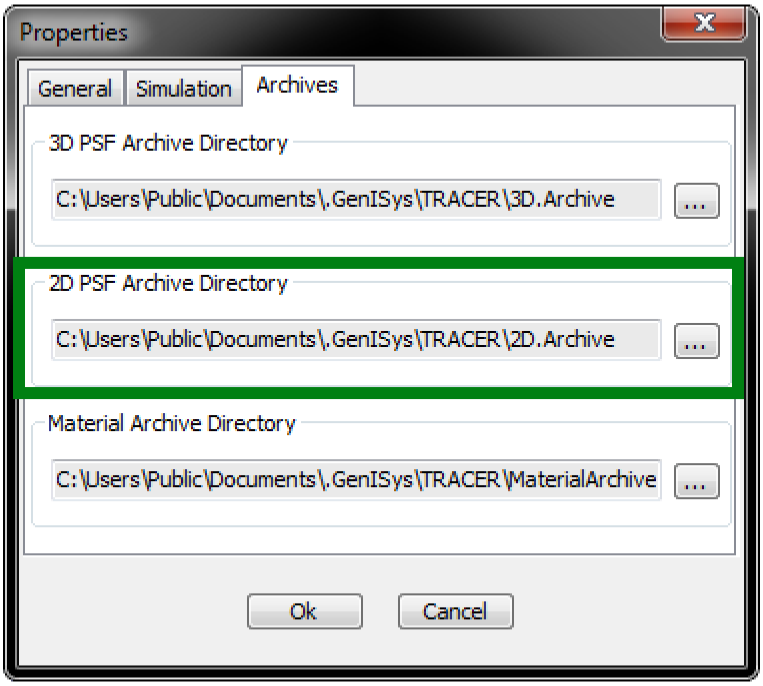
☐ Load last project on startup

☐ Show last view on startup

☑ Show Gaussian values as a pop-up for 2D fit nodes

Continue to the next page ...

On the Archives tab you should see the 2D PSF Archive Directory as highlighted below. This directory should be the same as in BEAMER.


Properties

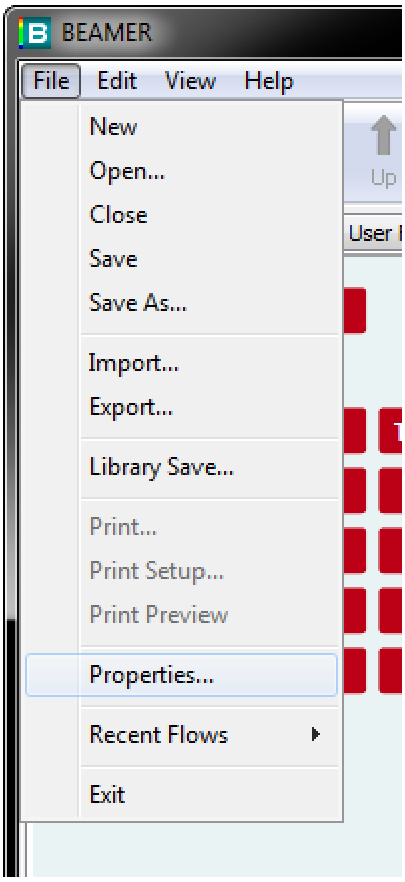
General | Simulation | Archives

3D PSF Archive Directory
C:\Users\Public\Documents\.GenISys\TRACER\3D.Archive

2D PSF Archive Directory
C:\Users\Public\Documents\.GenISys\TRACER\2D.Archive

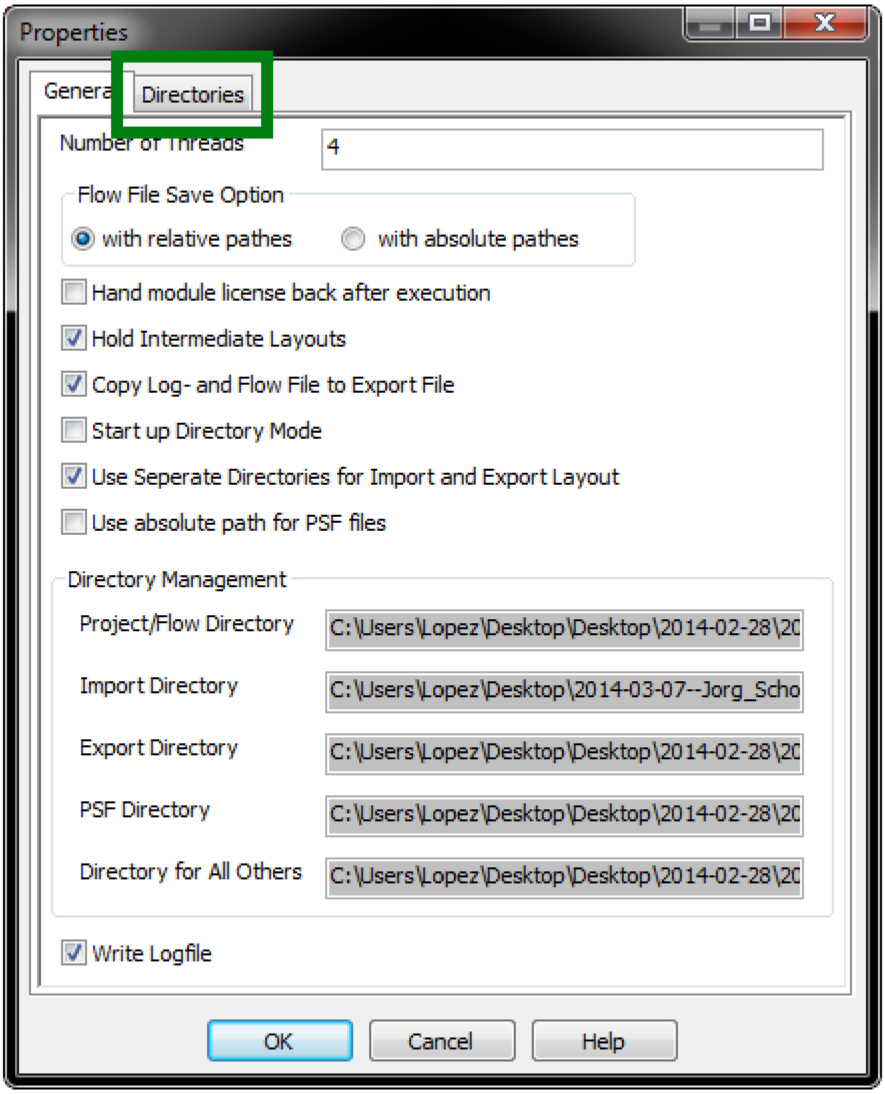
Material Archive Directory
C:\Users\Public\Documents\.GenISys\TRACER\MaterialArchive

In BEAMER, go to **Properties...** under the File menu like below:



Continue to the next page ...

You should get a dialog that looks like this:



Number of Threads: 4

Flow File Save Option
● with relative pathes ○ with absolute pathes

□ Hand module license back after execution

☑ Hold Intermediate Layouts

☑ Copy Log- and Flow File to Export File

□ Start up Directory Mode

☑ Use Seperate Directories for Import and Export Layout

□ Use absolute path for PSF files

Directory Management

Project/Flow Directory: C:\Users\Lopez\Desktop\Desktop\2014-02-28\20

Import Directory: C:\Users\Lopez\Desktop\2014-03-07--Jorg_Scho

Export Directory: C:\Users\Lopez\Desktop\Desktop\2014-02-28\20

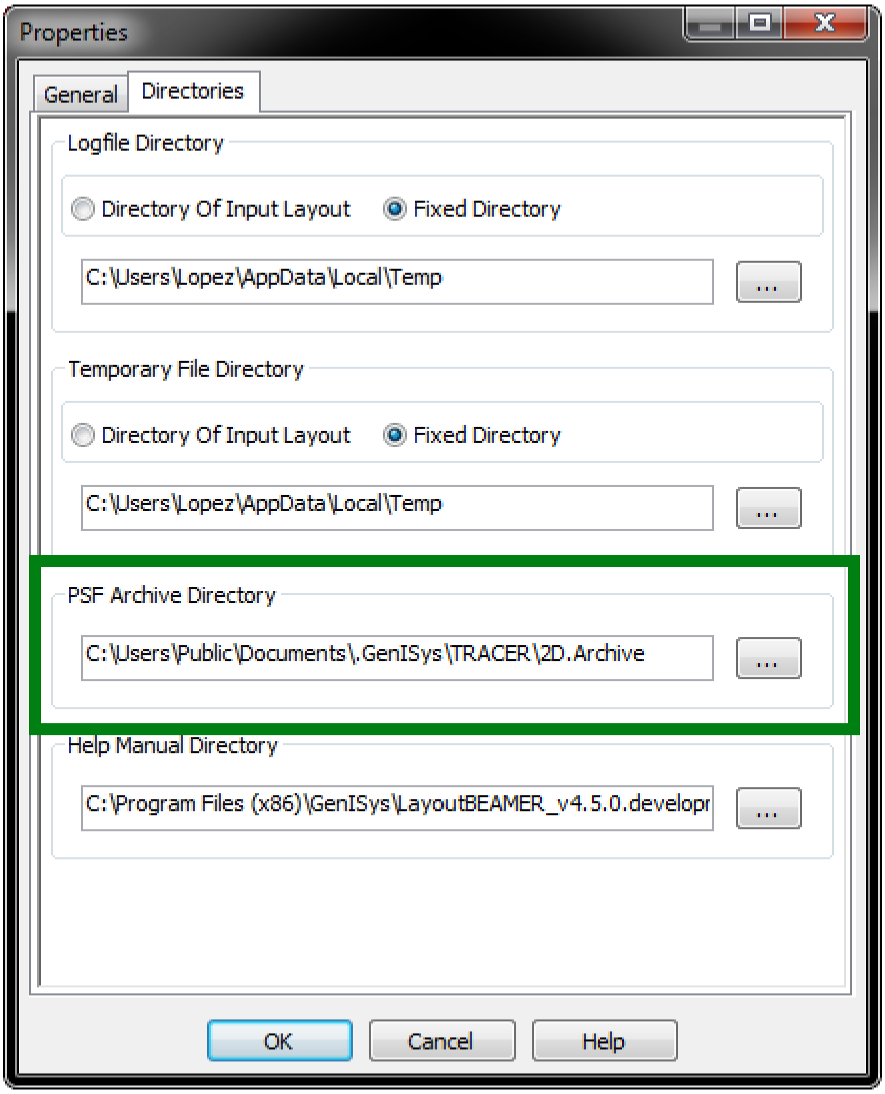
PSF Directory: C:\Users\Lopez\Desktop\Desktop\2014-02-28\20

Directory for All Others: C:\Users\Lopez\Desktop\Desktop\2014-02-28\20

☑ Write Logfile

Continue to the next page ...

Click on the Directories tab highlighted in Green to get the follow GUI:



Logfile Directory

○ Directory Of Input Layout ● Fixed Directory

C:\Users\Lopez\AppData\Local\Temp

Temporary File Directory

○ Directory Of Input Layout ● Fixed Directory

C:\Users\Lopez\AppData\Local\Temp

PSF Archive Directory

C:\Users\Public\Documents\.GenISys\TRACER\2D.Archive

Help Manual Directory

C:\Program Files (x86)\GenISys\LayoutBEAMER_v4.5.0.developr

Notice that the PSF Archive Directory is the same as the TRACER 2D PSF Archive directory. If it is not, make sure the two Archive Directories match so that BEAMER and TRACER are synchronized properly.

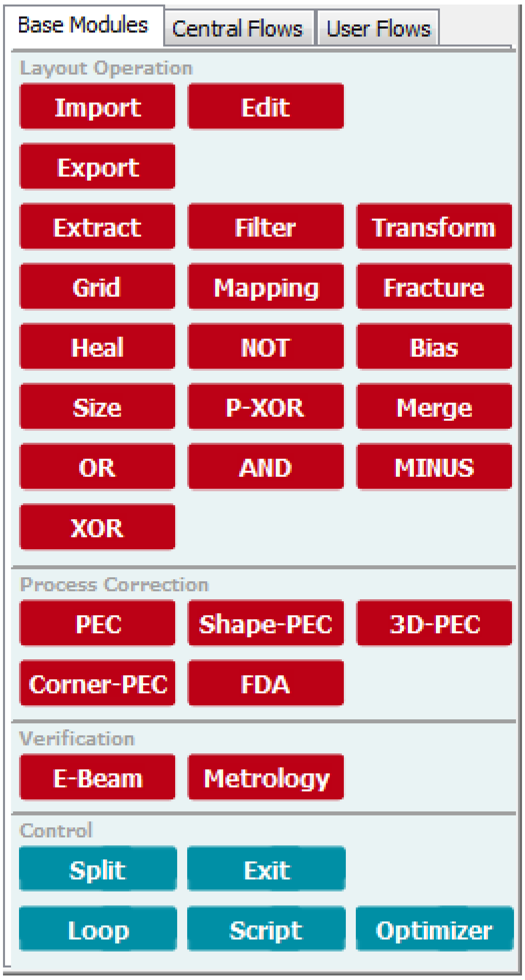
3.10 Base Module Library and User Libraries Overview

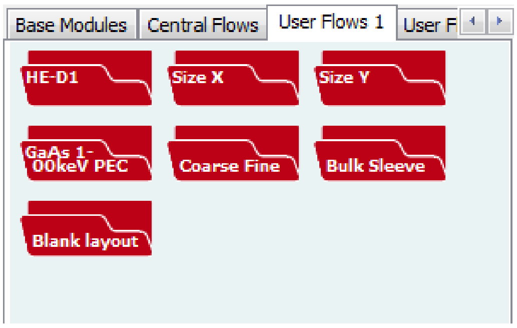
The *Base Modules* is the primary library for BEAMER. It provides a feature rich and comprehensive set of tools that can be used for preparing your data for exposure. All the functional modules needed for data preparation, proximity effect correction, simulation and modeling, fracturing and automation are provided.

A detailed explanation of individual module functionalities and settings can be found in the Base Modules section of this document.

BEAMER utilizes a user defined library/ database where users can save customized modules, complete flows, and all of their settings encapsulated into a single user defined database module. They are distinguished by a folder-shaped icon. Database modules behave like the Base Library modules and can be connected to the base modules, other database modules or be used individually.

There are two locations for storing these customized database modules: Central Flows and User Flows. The location does not change its behavior, but rather defines the availability to other users. A database module saved in the Central Flows library can be accessed by any user on the system, or on the network in cases where the system is configured as a server. Any module saved in the User Flows library is only available to the current user.





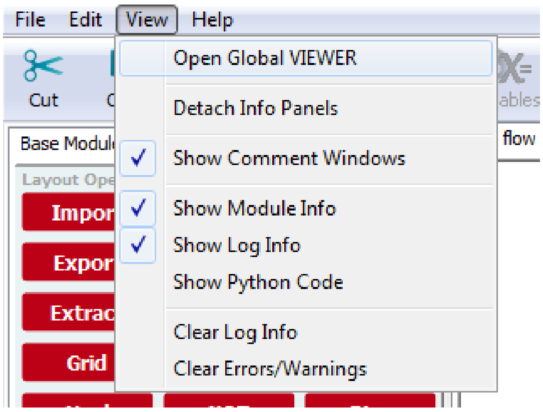
The library modules are saved by default in a BEAMER Flow library folder located in My Documents. The Database section of this document describes this functionality in more detail, including customization and database management.

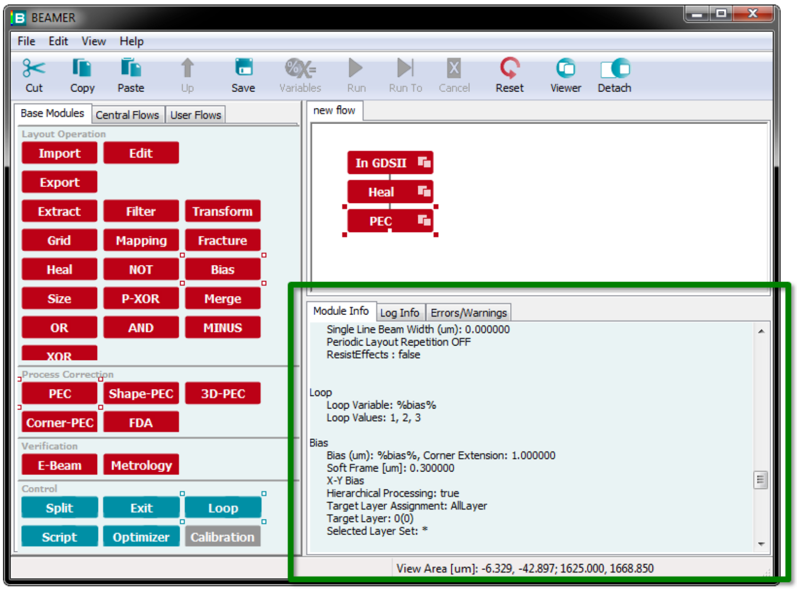
3.11 Info Panels

The Info Panels is located just below the workspace as highlighted in green below and is resizable for optimal viewing. It consist of up to four tabs:

1. Module Info
2. Log Info Tab
3. Python Code
4. Errors/Warnings

The visibility of these tabs are controlled via the View menu. Moreover, the clearing of the Log Info and Errors/Warnings tabs is available in the View menu.

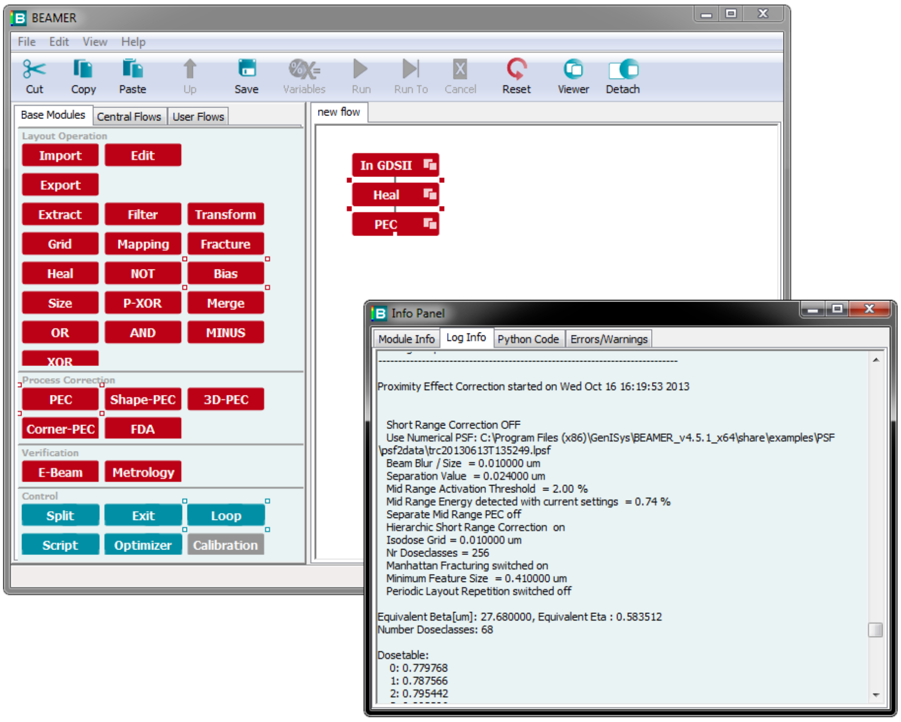




```
    Single Line Beam Width (um): 0.000000
    Periodic Layout Repetition OFF
    ResistEffects : false


Loop
    Loop Variable: %bias%
    Loop Values: 1, 2, 3

Bias
    Bias (um): %bias%, Corner Extension: 1.000000
    Soft Frame [um]: 0.300000
    X-Y Bias
    Hierarchical Processing: true
    Target Layer Assignment: AllLayer
    Target Layer: 0(0)
    Selected Layer Set: *
```

View Area [um]: -6.329, -42.897; 1625.000, 1668.850

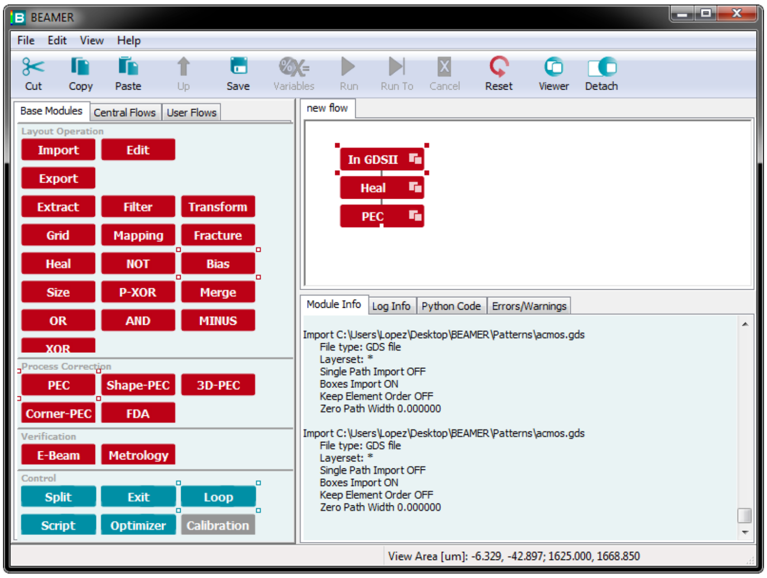
Additionally, the Info Panels can be quickly detached into a separate window for optimized viewing by selecting **Detach Info Panels** in the View menu. Illustrated below is an example of the Info Panels detached.


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
new flow
In GDSII Heal PEC
Info Panel
Module Info Log Info Python Code Errors/Warnings

```
------------------------------------------------------------------------------------

Proximity Effect Correction started on Wed Oct 16 16:19:53 2013


  Short Range Correction OFF
  Use Numerical PSF: C:\Program Files (x86)\GenISys\BEAMER_v4.5.1_x64\share\examples\PSF
\psf2data\trc20130613T135249.lpsf
  Beam Blur / Size  = 0.010000 um
  Separation Value  = 0.024000 um
  Mid Range Activation Threshold  = 2.00 %
  Mid Range Energy detected with current settings  = 0.74 %
  Separate Mid Range PEC off
  Hierarchic Short Range Correction  on
  Isodose Grid = 0.010000 um
  Nr Doseclasses = 256
  Manhattan Fracturing switched on
  Minimum Feature Size  = 0.410000 um
  Periodic Layout Repetition switched off

Equivalent Beta[um]: 27.680000, Equivalent Eta : 0.583512
Number Doseclasses: 68

Dosetable:
   0: 0.779768
   1: 0.787566
   2: 0.795442
```

Module Info

This tab reveals the information of a selected module without having to open the module dialog. Use it for quickly reviewing a module's setting. This tab can be toggled on or off via the View menu by selecting **Show Module Info**. In the example below, the Import module is selected and the Module Info tab tells us the parameters of the highlighted module.

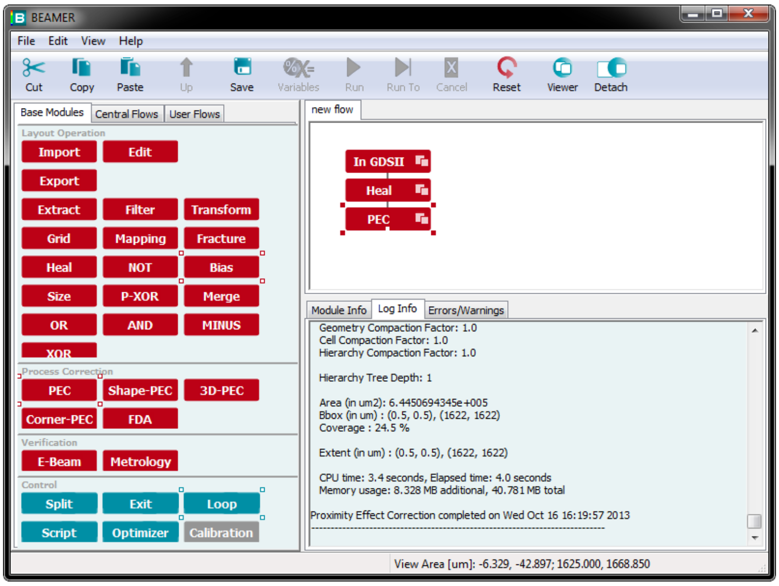

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
new flow
In GDSII Heal PEC
Module Info Log Info Python Code Errors/Warnings

```
Import C:\Users\Lopez\Desktop\BEAMER\Patterns\acmos.gds
    File type: GDS file
    Layerset: *
    Single Path Import OFF
    Boxes Import ON
    Keep Element Order OFF
    Zero Path Width 0.000000

Import C:\Users\Lopez\Desktop\BEAMER\Patterns\acmos.gds
    File type: GDS file
    Layerset: *
    Single Path Import OFF
    Boxes Import ON
    Keep Element Order OFF
    Zero Path Width 0.000000
```

View Area [um]: -6.329, -42.897; 1625.000, 1668.850

Log Info

This tab logs the run of every module in a flow. Subsequent runs are appended to the log. This tab can be toggled on or off via the View menu by selecting **Log Info**. Also, this tab can be cleared by going to the View menu and selecting **Clear Log Info**. In the example below, the log of the executed flow is shown in the Log Info tab revealing information module by module.

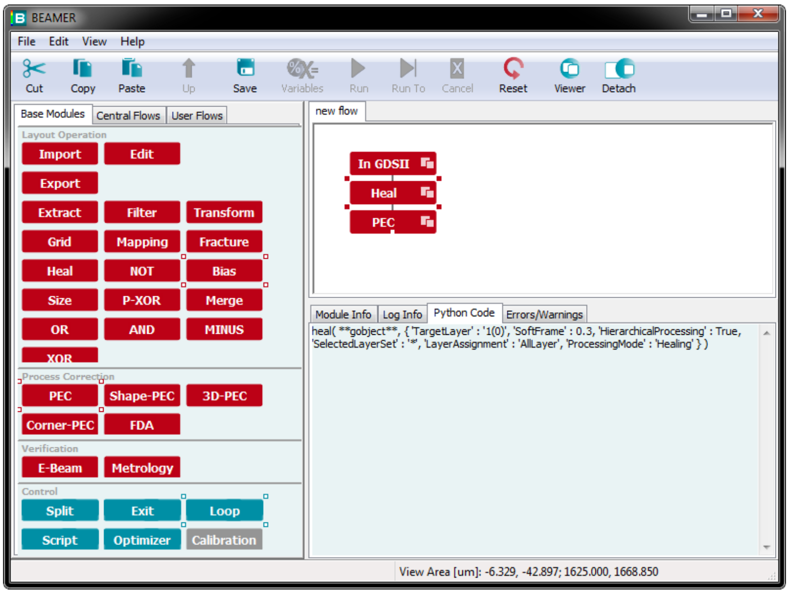


```
 Geometry Compaction Factor: 1.0
 Cell Compaction Factor: 1.0
 Hierarchy Compaction Factor: 1.0

 Hierarchy Tree Depth: 1

 Area (in um2): 6.4450694345e+005
 Bbox (in um) : (0.5, 0.5), (1622, 1622)
 Coverage : 24.5 %

 Extent (in um) : (0.5, 0.5), (1622, 1622)

 CPU time: 3.4 seconds, Elapsed time: 4.0 seconds
 Memory usage: 8.328 MB additional, 40.781 MB total

Proximity Effect Correction completed on Wed Oct 16 16:19:57 2013
--------------------------------------------------------------------------------
```

View Area [um]: -6.329, -42.897; 1625.000, 1668.850

Python Code

When toggled on, the Python Code tab reveals the basic Python syntax of a selected module that can be copied and pasted into your Python script. This tab can be toggled on or off via the View menu by selecting **Show Python Code**. In the image below, the PEC module is highlighted and the Python Code tab is toggled on revealing the Python code syntax of the module.



```
heal( **gobject**, { 'TargetLayer' : '1(0)', 'SoftFrame' : 0.3, 'HierarchicalProcessing' : True,
'SelectedLayerSet' : '*', 'LayerAssignment' : 'AllLayer', 'ProcessingMode' : 'Healing' } )
```

View Area [um]: -6.329, -42.897; 1625.000, 1668.850

Errors/Warnings

All errors and warnings are logged in this tab. This is the only tab that cannot be toggled off so that users are aware of any errors or warnings that occur during flow execution. This tab can be cleared by going to the View menu and selecting **Clear Errors/Warnings**.

3.12 Logging

BEAMER generates log files during layout processing, allowing the users to examine actions taken and the parameters set in the flow. Depending on the settings of the logging mechanism, the log files are stored at different locations and with different content. The logging method is available under File/Properties. The checkbox "copy log and flow file to the export file directory" managed the mode to be used.

With the checkbox turned off, the log file is written to the same directory the first Import module used for it's layout. The log file is given a name associating it to the imported layout with a time stamp attached.

When ever a reset is done on the flow and the flow is run again, the log file will be expanded with the logging information for the newly executed modules.

The second mode is when the checkbox is turned on. Then the log file is written to the same directory the Export module points to. This means that for each Export a new log file is generated. In complex flows, this makes the tracking of steps between multiple Export modules easier. This mode also writes a copy of the flow to the Export directory.

3.13 Layer/Datatype Handling

Layer/datatype handling is unified across all modules. The information presented here can be applied to IMPORT, EXPORT, FDA, EXTRACT or HEAL.

Import
For input formats having layer and datatype numbers, a name in the form layer(datatype) is generated and used for the GUI. For example, a GDS-II element in layer 5 / datatype 8 will get the layer name: ‘5(8)’. Other formats that support layer names (such as CIF and DXF), these layer names will be maintained throughout the entire data preparation flow.

Export
For output formats with layer and datatype numbers, an attempt is made to extract layer and datatype numbers from the layer name. If the layer name is consistent with the format 'layer(datatype)', and the layer and datatype numbers are positive, these numbers will be used. If the layer name is a simple positive number, it will be used as the layer number and the datatype is set to 0. If no layer number can be extracted, layer numbers will be generated, one for each layer name. If the number can be extracted from the name for some of the layer, these will be mapped to the identified numbers first, the remaining will be mapped to the remaining available numbers.

Layer name ‘0’ generates an element with layer 0 and datatype 0 on GDS-II output.
Layer name ‘1(4)’ generates an element with layer 1 and datatype 4 on GDS-II output.
Layer name ‘foobar’ generates an element with layer 2 and datatype 0.
Another layer name in the same layout would then generate an element with layer 3, datatype 0.

New Layers
Operations that generate new layers (Boolean operations, heal, …) a layer name can be specified.

Extract
The extract operation has only one entry field for layer selection. For this, a special syntax is supported to still allow the selection of layer and datatype ranges, consistent with the 'layer(datatype)' syntax. Ranges can be specified using a selection string in the form ls-le(ds-de) where ls, le, ds, and de are numerical values. This will select all layers which follow the layer(datatype) convention and where layer is in the range from ls to le and datatype is in the range from ds to de. The range specification ls-le or ds-de may be replaced by ‘*’ to select all layers or datatypes, or by a singe number to select only this layer or datatype. In addition, the specification of the datatype range can be omitted and implicitly all datatypes are selected. A simple ‘*’ selects all layer names. Several selection strings may be separated by a comma.

Example String	Function
*	selects all layer names
foo	selects only layer named 'foo'
foo, bar	selects layers named 'foo' and 'bar'
4(0)	selects layer named '4(0)'
1-3(7-9)	selects layers named ‘1(7)’, ‘1(8)’, ‘1(9)’, ‘2(7)’, ‘2(8)’, ‘2(9)’, ‘3(7)’, ‘3(8)’, ‘3(9)’ if they exist within the pattern
3(*)	selects layers following the layer(datatype) convention where layer is ‘3’
*(4)	selects layers following the layer(datatype) convention where datatype is ‘4’.
2	selects layer name ‘2’ and also all layers following the layer(datatype) convention where layer is ‘2’ (e.g.’2(6)’)

Merge Operation
The merge operation merges by layer names.

3.14 Environment Variable Handling

BEAMER supports the use of an environment variable to controls its principle behavior.

GENISYS_BEAMER_USER
Controls where BEAMER looks for a profile, meaning which user generated libraries and settings are loaded. By default BEAMER looks in the .GenISys folder of the user’s home directory. Using this variable sets the folder for the default setting to .GenISys/< GENISYS_BEAMER_USER>/ .

GENISYS_BEAMER_DEFAULTS
Sets the directory path for global BEAMR settings. Settings in these folders are read in first and can be overwritten by settings in the user folder. The idea is to have a global setting recommendation which can be locally overwritten.

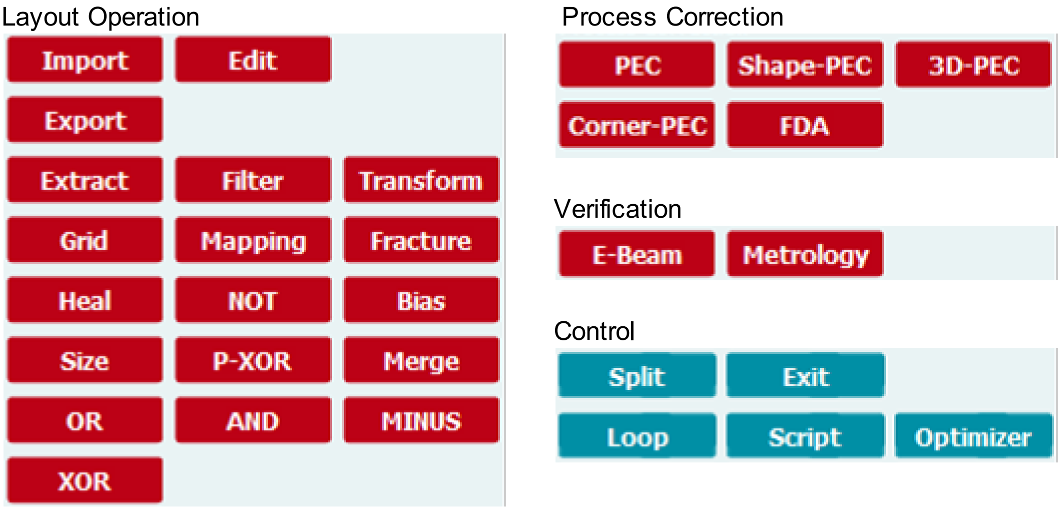
BEAMER_USER_LIBRARY_PATH
Controls the path where user libraries are being stored. If not set it uses the home directory of the user.

BEAMER_CENTRAL_LIBRARY_PATH
Controls the path where the centralized libraries are being stored. If not set, it store the libraries in the 'Flow_Library' folder of the installation directory.

GENISYS_BEAMER_EXPORT_CONFIGURATION_PATH
Controls the path for the storage of the configuration files (currently EBPG and HIMT only). The default of this path is the installation directory and there the configuration path.

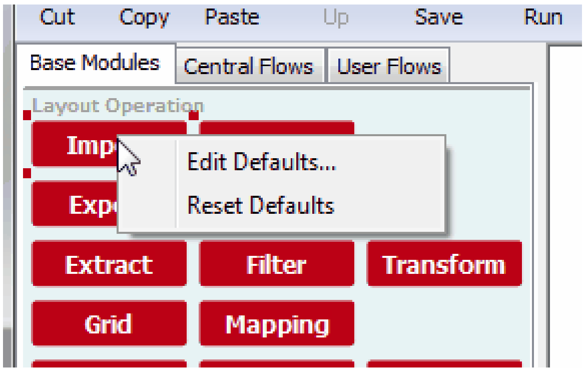
4 Base Modules Library

The BEAMER Base Modules library is rich in functionality and includes a comprehensive set of tools for layout operations. In this section, a detailed description for each module is provided. Note that not all modules and their full functionality will be available by default, as some modules, features or capabilities may be limited by the license being evaluated or purchased.



4.1 Setting Default Values

Each module can be assigned new default values. Right-clicking on a module will open up a parameter dialog box. There is also an option to reset to default settings.

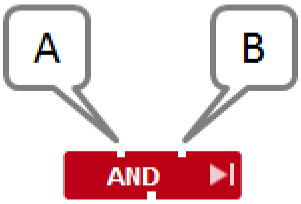


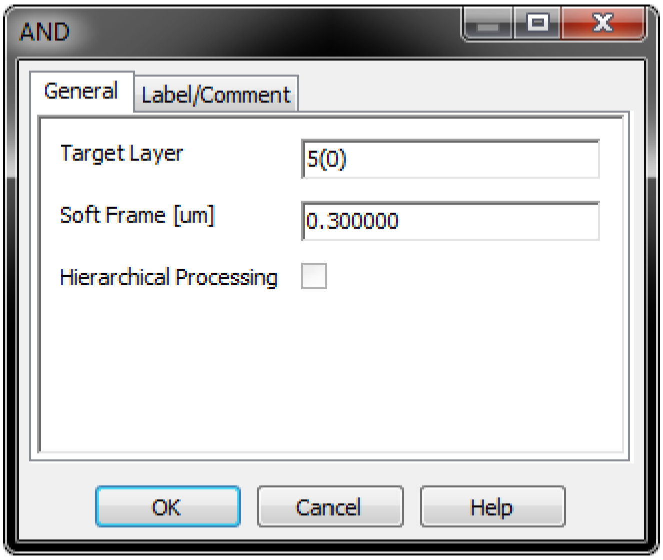
4.2 Layout Operation

Layout Operation modules provide a powerful set of base functionality for data preparation.

4.2.1 AND

The AND module takes two layouts as input and returns data where Layout A and Layout B data overlap.



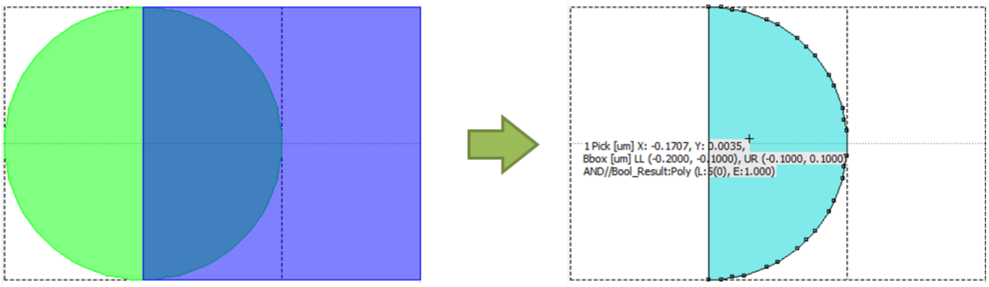


Target Layer 5(0)

Soft Frame [um] 0.300000

Hierarchical Processing ☐

The AND operation will keep the areas that are in common between layout A (green) and B (dark blue):



```
1 Pick [um] X: -0.1707, Y: 0.0035,
Bbox [um] LL (-0.2000, -0.1000), UR (-0.1000, 0.1000)
AND//Bool_Result:Poly (L:5(0), E:1.000)
```

Layout Logic:

AND		
A	**B**	**Result**
0	0	0
0	1	0
1	0	0
1	1	1

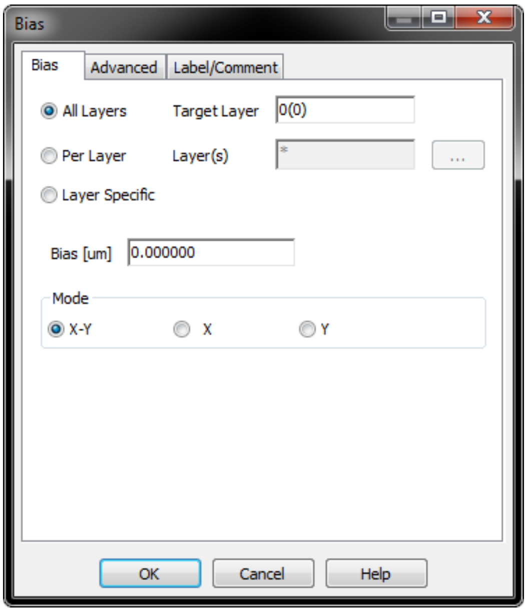
0 indicates a layout element is absent. 1 indicates a layout element is present.

The target layer and datatype for the result can be specified by opening the parameter window with a double click. By default, the target layer and datatype is 5(0). To avoid unnecessary fracturing of the original layout a soft frame can be defined in the parameter window.

This operation uses virtual tiles in memory to execute this operation quickly. The pattern as a whole is divided into virtual tiles, where each tile is kicked off to an available processing core for execution. Depending on the size of the pattern and available memory, few or many tiles are generated and the desired operation is executed on the tiles in parallel. Once all the tiles have been processed, they are reassembled into one pattern. The Softframe parameter is the overlap between these virtual tiles. In essence, any feature size that is equal to or smaller than the softframe (overlap) will not be cut in the virtual tile operation. Since feature sizes smaller than 0.300nm can be sensitive to being cut by this operation by producing non-optimal fractures, the default softframe value of 0.300um can be kept if you are unsure about what value to use.

Hierarchical Processing is disabled for this module.

4.2.2 Bias



Bias

Tabs: Bias, Advanced, Label/Comment

● All Layers   Target Layer: 0(0)

○ Per Layer   Layer(s): *

○ Layer Specific

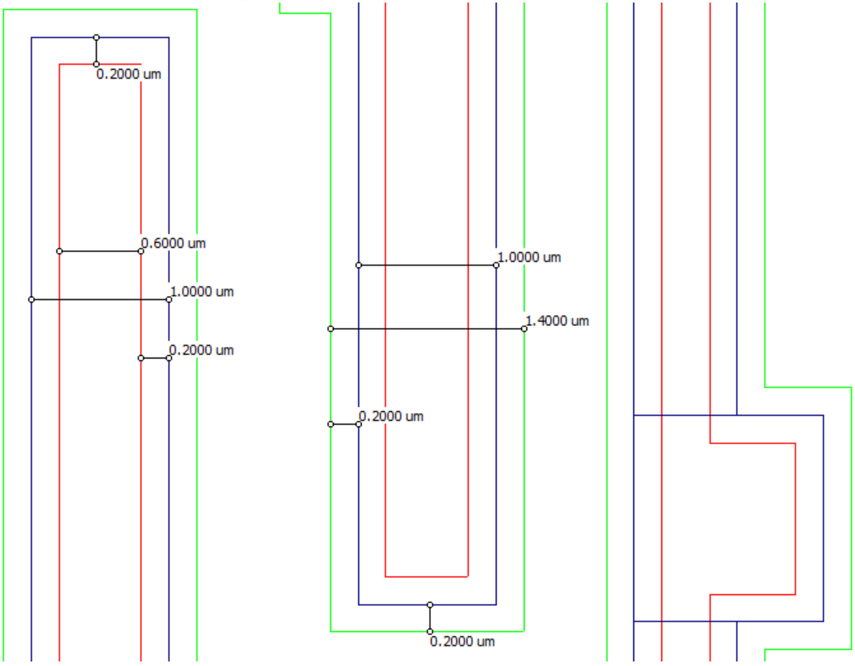
Bias [um]: 0.000000

Mode: ● X-Y ○ X ○ Y

The Bias module allows you to apply a geometrical bias to the polygons in the layout, either positive or negative. If a positive bias is applied to two or more abutting shapes, they will be healed together. Should you desire a bias operating without an implied heal, use the Grow module instead.

The Bias parameter adds or subtracts the entered dimension to all sides of all polygons. In the example below, the dark blue outline indicates the original layout with a 1µm width. A positive bias of 0.2µm has been applied to the original pattern to produce the green outline and a negative bias of -0.2µm has been applied to the original pattern to produce the red outline.

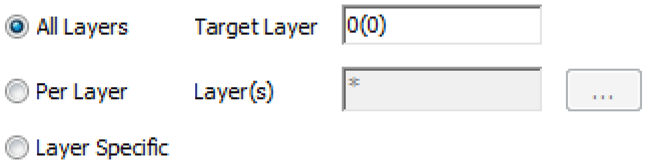
Here the operation was applied using the X-Y mode.



Should an X or Y only bias is required on a pattern, the X or Y mode can be selected respectively.

4.2.2.1 Layerwise Operation

The Bias and Grow modules support layerwise operation on a pattern. By default, all layers are biased together. As a result, layer information is lost and the resulting data is placed on a single target layer. In the case below, the target layer/datatype is set to 0(0).

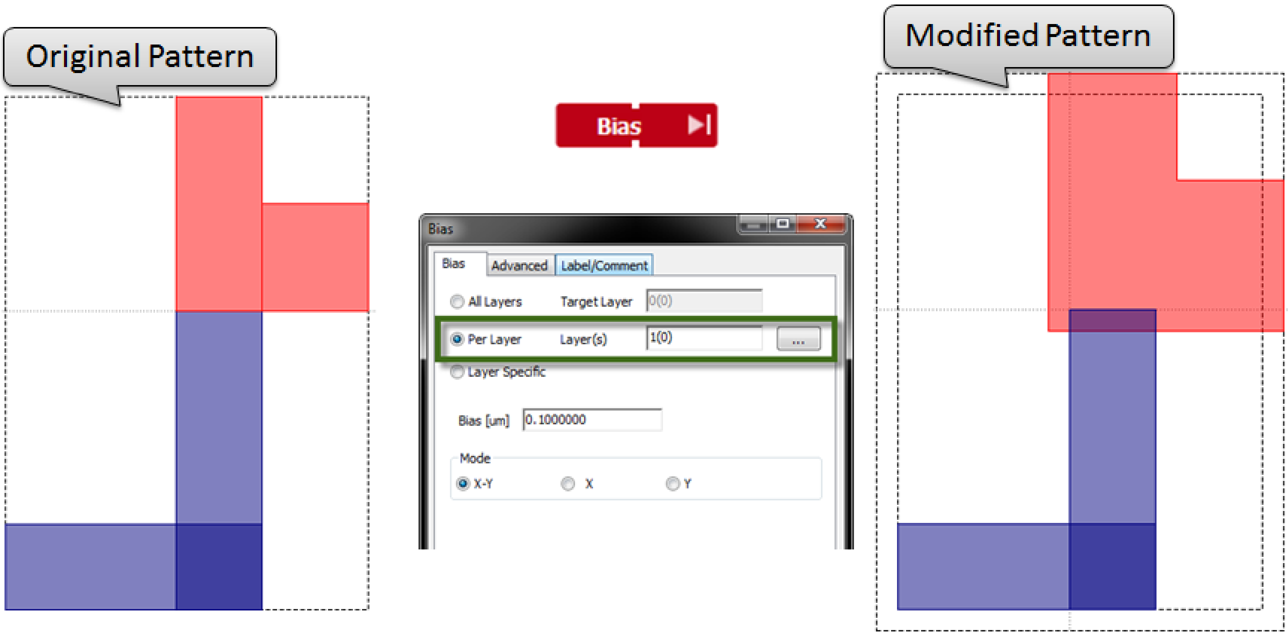


● All Layers   Target Layer 0(0)

○ Per Layer   Layer(s) *   ...

○ Layer Specific

When Per Layer is selected, layers can be explicitly selected for modification. Here only layer 1(0) will be modified uniformly leaving layer 2(0) untouched. A wildcard character (*) can be used to denote all layers in this mode of operation.



○ All Layers   Target Layer 0(0)

● Per Layer   Layer(s) 1(0)   ...

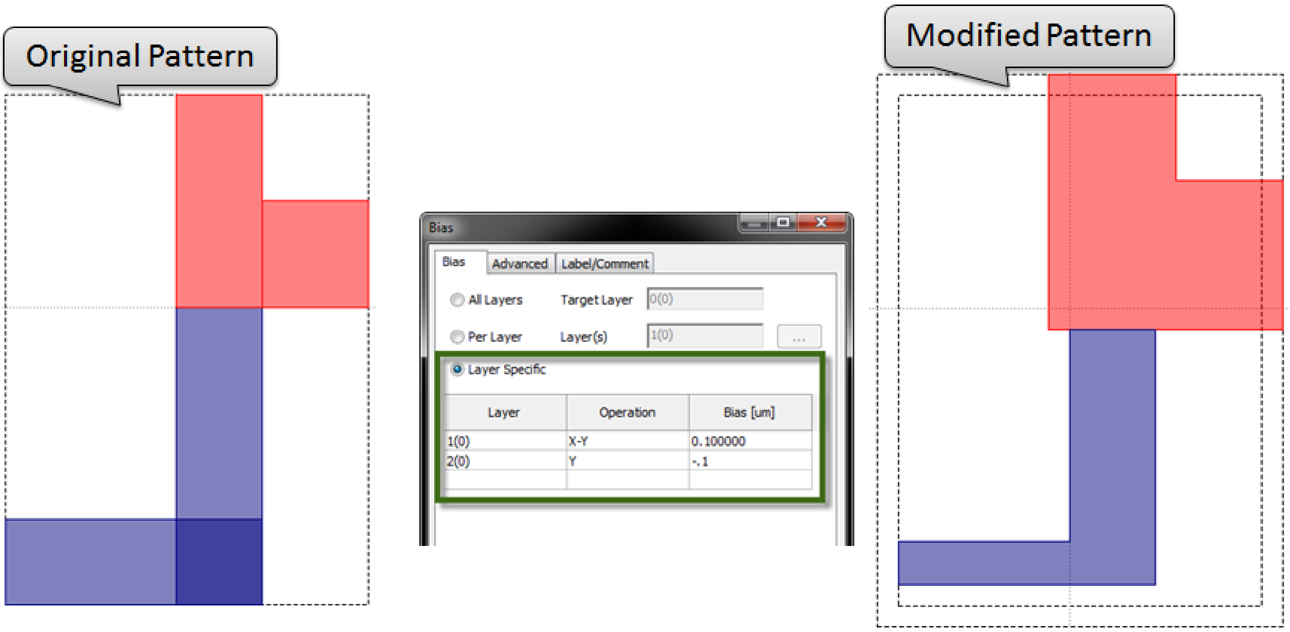
○ Layer Specific

Bias [um] 0.1000000

Mode

● X-Y   ○ X   ○ Y

Lastly, Layer Specific operations allow the user to select layers available in the pattern and apply a specific bias operation (X-Y, X, Y) to each specified layer. Moreover, each layer can have a different bias value applied.



○ All Layers   Target Layer 0(0)

○ Per Layer   Layer(s) 1(0)   ...

● Layer Specific

Layer	Operation	Bias [um]
1(0)	X-Y	0.100000
2(0)	Y	-.1

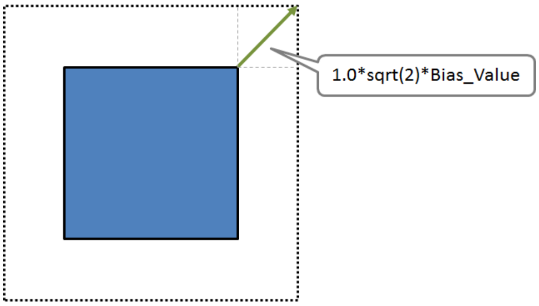
4.2.2.2 Corner Extension

The **Corner Extension** parameter in the **Advanced** tab controls the distance where these extensions are cut off. By default the Corner Extension (CE) is 1.0. We define the placement of a corner vertex using a bias with the formula:

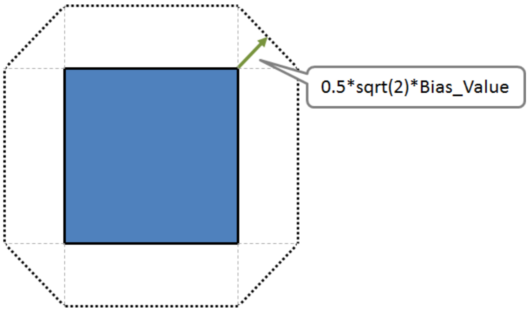
*CE*sqrt(2)*Bias_Value*

This formula yields the length of the orthogonal line that is bisecting the corner's angle. This length is the maximum distance that a corner vertex can be placed using a bias.

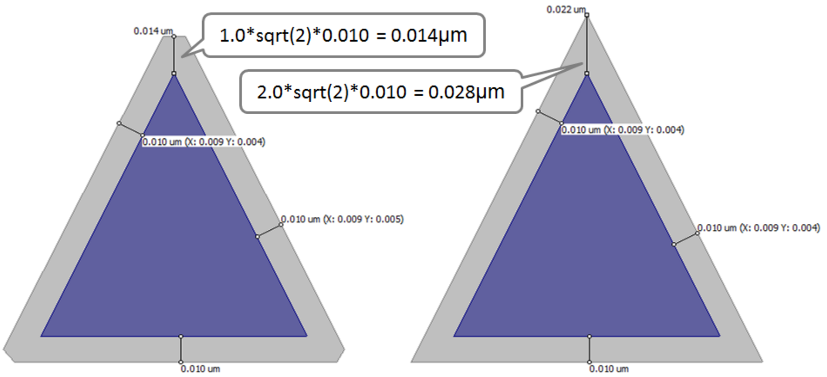
CE=1.0 means that the cutoff length is equal to the square root of two times the bias amount. Restated, the CE value imposes a limit on the new placement of a corner vertex with a bias. A simple example using a 90 degree corner illustrates the impact of the CE. The shape we wish to bias is in blue and the intended shape after a positive bias is outlined as dotted black. Below, CE=1.0 is used and the square is positively biased as expected with no changes in the corner value. To be clear, the formula does not dictate where a vertex is placed if CE >> 1; it simply allows the vertex to move without as much limitation as compared to a CE < 1.



By using a CE value of 0.5 when applying a positive bias, the corner is truncated by half, effectively rounding the corner in the litho. Since the limit of the corner placement is half, the limit imposes a cutoff on the corners as illustrated below. In this case, the formula says that a corner cannot extend beyond the calculated length of the line that is orthogonally bisecting the corner angle. As a result the corners are truncated.

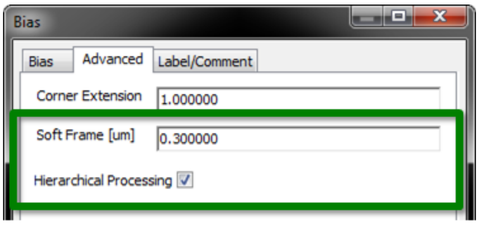


In particular, biasing triangles pose an interesting challenge. Illustrated below is an example that positively biases a triangle in blue to the triangle in gray by 10nm with CE=1. Since the distance of the original corner position to the biased position is limited to 14nm, the corners are truncated as seen on the bottom left. When CE=2 is used, the limit of the distance is increased to 28nm which allows the corner to be placed 22nm above the original position.


0.014 um
1.0*sqrt(2)*0.010 = 0.014µm
0.010 um (X: 0.009 Y: 0.004)
0.010 um (X: 0.009 Y: 0.005)
0.010 um
0.022 um
2.0*sqrt(2)*0.010 = 0.028µm
0.010 um (X: 0.009 Y: 0.004)
0.010 um (X: 0.009 Y: 0.004)
0.010 um

4.2.2.3 Softframe and Hierarchical Processing

In the **Advanced** tab, the Softframe and Hierarchical Processing parameters provide extended functionality in the Bias module. We will explain the softframe operation followed by the Hierarchical Processing operation.



Corner Extension 1.000000

Soft Frame [um] 0.300000

Hierarchical Processing ☑

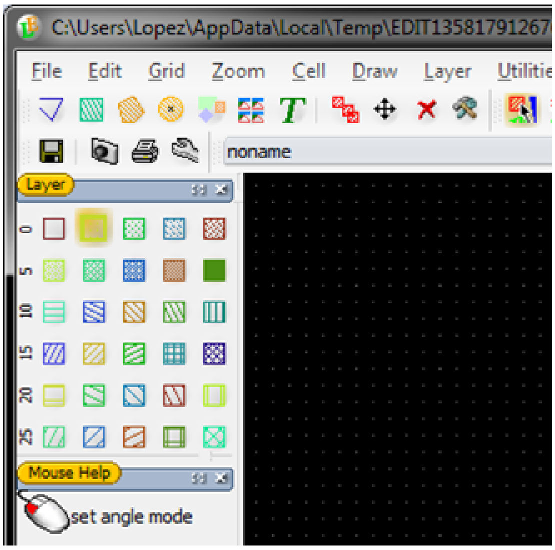
This operation uses virtual tiles in memory to execute this operation quickly. The pattern as a whole is divided into virtual tiles, where each tile is kicked off to an available processing core for execution. Depending on the size of the pattern and available memory, few or many tiles are generated and the desired operation is executed on the tiles in parallel. Once all the tiles have been processed, they are reassembled into one pattern. The Softframe parameter is the overlap between these virtual tiles. In essence, any feature size that is equal to or smaller than the softframe (overlap) will not be cut in the virtual tile operation. Since feature sizes smaller than 0.300nm can be sensitive to being cut by this operation by producing non-optimal fractures, the default softframe value of 0.300um can be kept if you are unsure about what value to use.

Selecting **Hierarchical Processing** takes advantage of hierarchical layouts providing a shorter processing time. Depending on the operation and the design of the pattern some hierarchy may be lost.

4.2.3 Edit

The Edit module can be used to generate layouts from scratch or to edit layouts that have been imported using LayoutEditor.

LayoutEditor, from Juspertor UG in Germany, is a fully functional layout editing program with sophisticated functions that is included in BEAMER via the Edit module.



Edit

It has a schematic capturing function where one can place a device in the schematic window and simply move the mouse to the layout window to get the corresponding device design. LayoutEditor is also equipped with a powerful macro functionality that allows for the programming of pattern generation. A full online manual and wiki is available at http://layouteditor.net/.

Adding the Edit module to a flow will open the LayoutEditor when the flow is executed which results in one of two operations:

1. It will load a layout from the previous module in the flow or
2. When there is no previous module, a blank layout will open, allowing for the creation of a new design.

If a new pattern is generated, it is recommended that the pattern be saved immediately using the Export module. Be sure to immediately run the Export module to save pattern is written to file.

Edit
Export

4.2.4 Export

The Export module is used to save layouts in standard layout formats and supported e-beam tool formats.

Export standard file formats:

- LEDB
- GDSII
- CIF
- DXF
- LTXT (compatible with CTXT/TXL)
- Generic Bitmap (grayscale PNG)

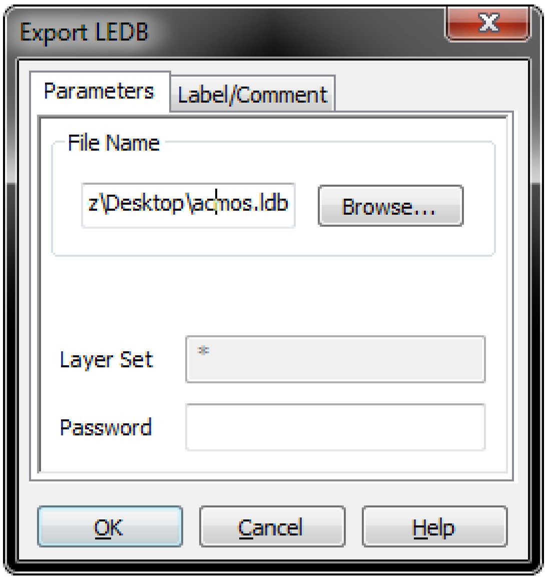
Export (fracturing) of e-beam tool formats: (license required)

- JEOL (JEOL51, JEOL52)
- Vistec VB6 (VEP, FRE) and EBPG (GPF, IWFL)
- Raith EBL
- Elionix CEL
- MEBES
- NuFlare
- Heidelberg DWL 2000 format

Refer to detailed descriptions below for each of the supported file formats. Export supports hierarchical fracturing for all machine formats for faster processing.

4.2.4.1 LDB

LEDB is an proprietary, internal BEAMER database format that provides high performance access for loading and viewing. This format will require less memory and provide extremely fast access. It is the recommended format if the data needs to be accessed multiple times. You can optionally enter a password for controlling access to the file.



File Name: z\Desktop\acmos.ldb

Layer Set: *

Password:

Layer Set

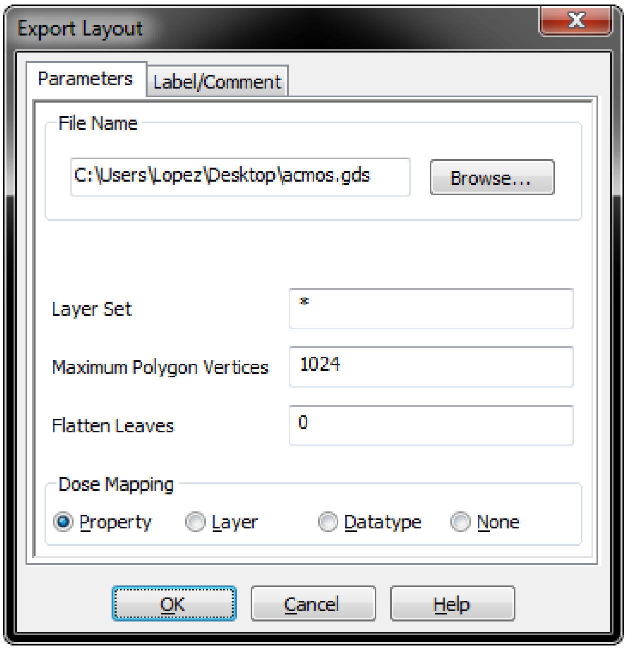
Disabled at this time

Password

Provide a password to protect or access when exporting or importing sensitive files.

4.2.4.2 GDS

The layout will be saved in the GDSII format. BEAMER supports all GDSII features.



Export Layout

File Name: C:\Users\Lopez\Desktop\acmos.gds

Layer Set: *

Maximum Polygon Vertices: 1024

Flatten Leaves: 0

Dose Mapping: ● Property ○ Layer ○ Datatype ○ None

Layer Set

Defines the layers and datatypes to be imported or exported (depending if you are using the Import or Export module). This field can make use of the special annotations available for layer datatype handling.

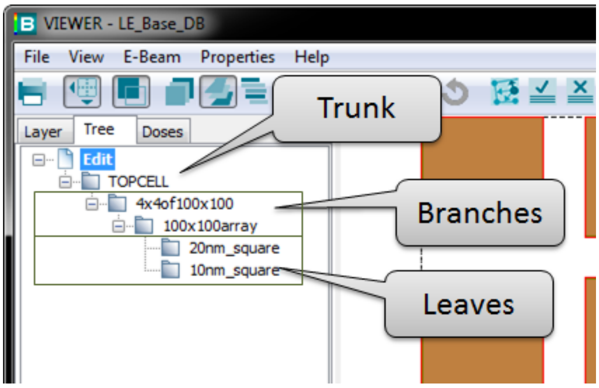
Maximum Polygon Vertices

Provides the maximum number of vertices in a polygon.

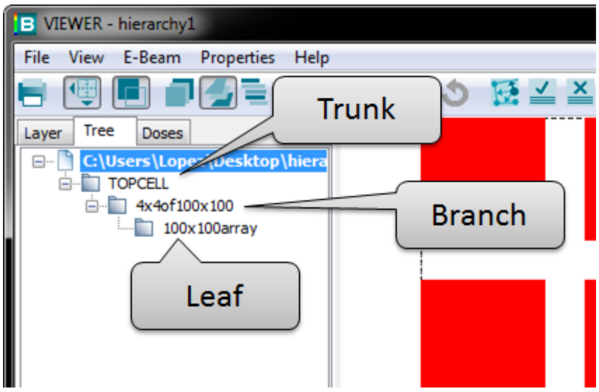
Flatten Leaves

The levels of the GDS hierarchy are similar to a tree with branches and leaves. The trunk of the tree is the topmost part of the hierarchy. When you expand the hierarchy the branches can be seen as cells in the hierarchy. These cells can be made up of other cells (or branches). Eventually you reach the bottommost elements that have no branches; these are the leaves. Leaves are the most primitive cells. **Flatten Leaves** indicates which part of the hierarchy that is flattened inward from the leaves. Below is an example GDS that has hierarchy and uses **Flatten Leaves**.

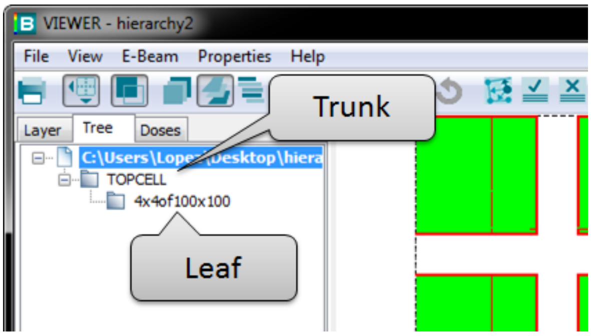
With **Flatten Leaves** = 0 (default value), the original hierarchy is preserved. No leaves are flattened.



Flatten Leaves = 1 will flatten the parent branch the original leaves, making the former branch (3x3Array) the bottommost level of the hierarchy (a new leaf). The original leaves 10nmDots and 20nmDot have been removed and flattened into the parent cell 3x3Array.



Flatten Leaves = 2 will flatten the two levels in from the original leaves into the parent branch 2x2ArrayOf3x3, making the former branch (2x2ArrayOf3x3) the bottommost level of the hierarchy (a new leaf).



Dose mapping

Dose mapping provides ways to assign a dose for proximity effect corrected layouts. **Property** adds the dose information to each shape. Selecting **None** will remove all dose information upon export.

Selecting **Layer** or **Datatype** will save each dose class in one layer (or datatype) producing a layer dose table (layer or datatype number and dose value). This layer dose table will be saved under the directory using the same name as the layout with the extension *.ldt (layer dose table).

An example of a layer dose table file is as follows:

```
BEAMER Dose table V1.0
Dose Assignment by Layer
( 0, 1.391 )
( 1, 1.396 )
( 2, 1.401 )
( 3, 1.406 )
...
```

It is possible to edit this text file, adjust the doses and save it back under the same name. This is an alternative way of manually assigning doses to layers.

The GDS export with dose mapping can be used to load PEC corrected data to some e-beam tools (e.g. Nanobeam, Crestec).

4.2.4.3 OASIS

The OASIS export driver will export any pattern to the OASIS format with basic compression capabilities. The OASIS file format intelligently stores geometries by implementing the use of repetition flags for layer, datatype, x-position, y-position, width and height. The OASIS export driver does not reorder elements in order to make better use of these repetitions. It also does not use the OASIS internal zLib block compression.

4.2.4.4 CIF

CIF is the CalTech Intermediate Format file format. No parameter settings are needed for this format.

CIF is a human readable text based format. CIF files can be opened with a text editor (e.g. Wordpad) and modified. This format supports polygons and circles.

For a detailed description see: http://en.wikipedia.org/wiki/Caltech_Intermediate_Form.

4.2.4.5 DXF

This function exports the layout into the common DXF format. No parameter settings are needed for this format.

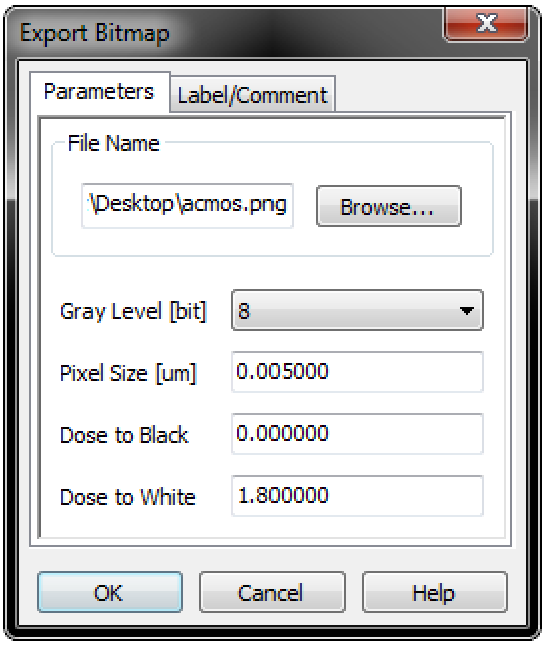
4.2.4.6 LTxt

The layout will be saved in a human readable hierarchical text format that is compatible with the standard TextLib (txl) or CTXT file format, including support of quasi-standard file format extensions. The files can be opened and modified using a text editor (e.g. Wordpad). No parameter settings are needed for this format.

Please contact technical support for further information.

4.2.4.7 Generic Bitmap (PNG)

Exports the pattern into a grayscale .PNG file format. Grayscale PNG images can vary in the number of shades of gray that are available in their color palette. This is dictated by the number of bits in the image. An 8-bit grayscale will have 256 (2^8) shades of gray. The color black has a value of 0, and the color white has a value of 255.



Export Bitmap

File Name: \Desktop\acmos.png

Gray Level [bit]: 8

Pixel Size [um]: 0.005000

Dose to Black: 0.000000

Dose to White: 1.800000

Gray Level [bit]

The number of gray tone levels per pixel are determined by the number of bits: 1, 2, 4, or 8bit.

Pixel Size [nm]

Defines the size to convert an individual pixel.

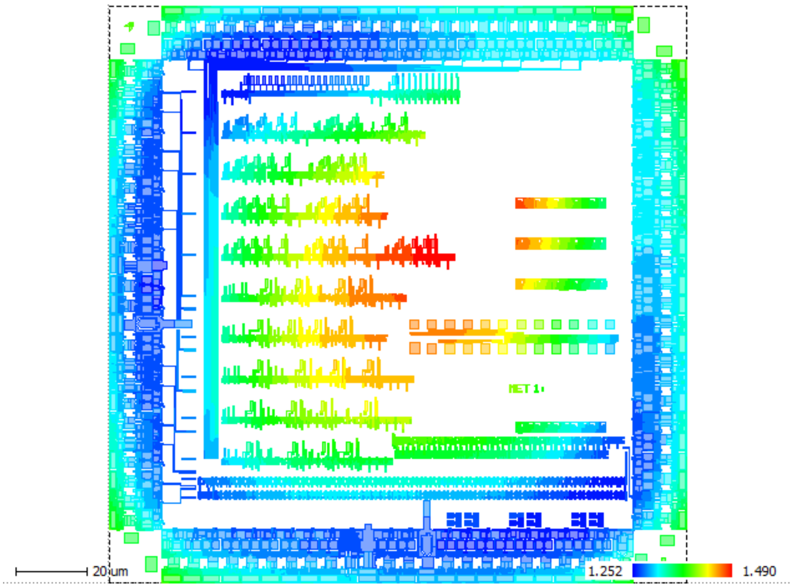
Dose to Black

Defines the dose to assign to the color black from the pattern.

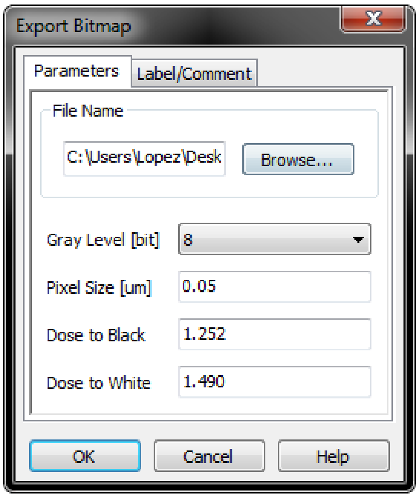
Dose to White

Defines the dose to assign to the color white from the pattern.

An example of **Dose to Black** and **Dose to White** is provided here using the corrected pattern illustrated to the right. The pattern has a minimum and maximum dose of 1.252 and 1.490, respectively. All doses in between are mapped to a gray level depending on the **Gray Level** chosen. Here, The grayscale palette will be set to 8-bit (256 shades of gray), and the pattern will be exported with a fixed pixel size of 50nm.



If **Dose to Black** = 1.252 and **Dose to White** = 1.490, the resulting PNG that will have any pixels with a dose below 1.252 will be black and any pixels with a dose above 1.49 will be white.


Export Bitmap

Parameters | Label/Comment

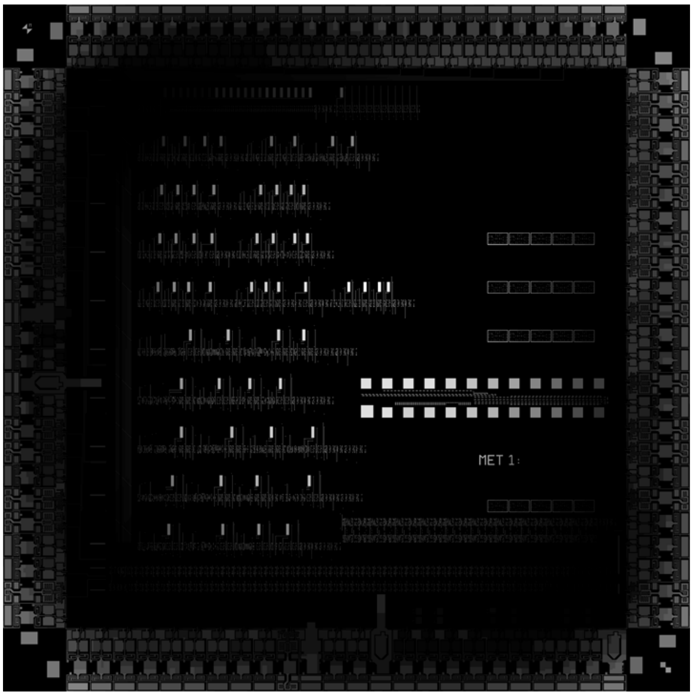
File Name: C:\Users\Lopez\Desk

Gray Level [bit]: 8

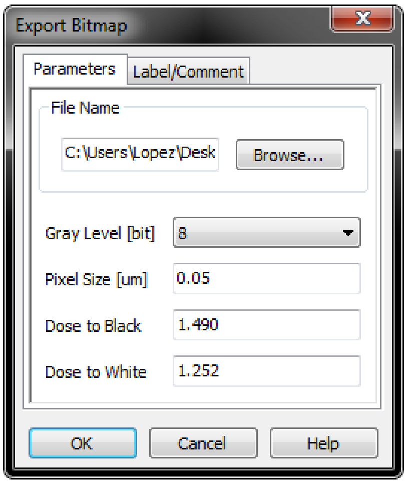
Pixel Size [um]: 0.05

Dose to Black: 1.252

Dose to White: 1.490



If **Dose to Black** = 1.490 and **Dose to White** = 1.252, the resulting PNG that will have any pixels with a dose above 1.490 will be black and any pixels with a dose below 1.252 will be white.


Export Bitmap

Parameters | Label/Comment

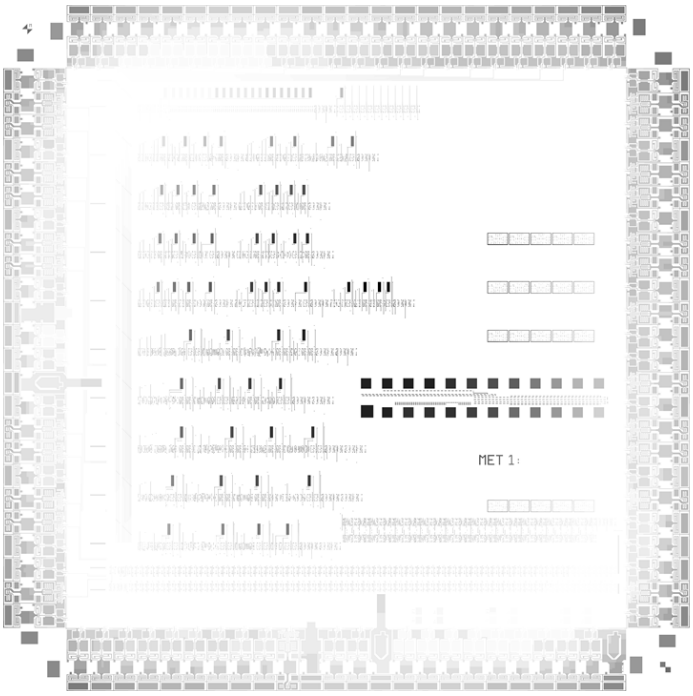
File Name: C:\Users\Lopez\Desk

Gray Level [bit]: 8

Pixel Size [um]: 0.05

Dose to Black: 1.490

Dose to White: 1.252



As you can see from the examples, the tone of the pattern can be quickly reversed depending on your needs.

4.2.4.8 JEOL 01

JEOL v01 is the machine format for some older JEOL Gaussian beam exposure systems. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to JEOL licensed users.

4.2.4.9 JEOL V51

JEOL v51 is the machine format for some older JEOL Gaussian beam exposure systems:

- JBX-5DII
- JBX-5DII(U)
- JBX-5000LS
- JBX-5FE
- JBX-6000FS
- JBX-6000FS/E

Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to JEOL licensed users.

4.2.4.10 JEOL V52

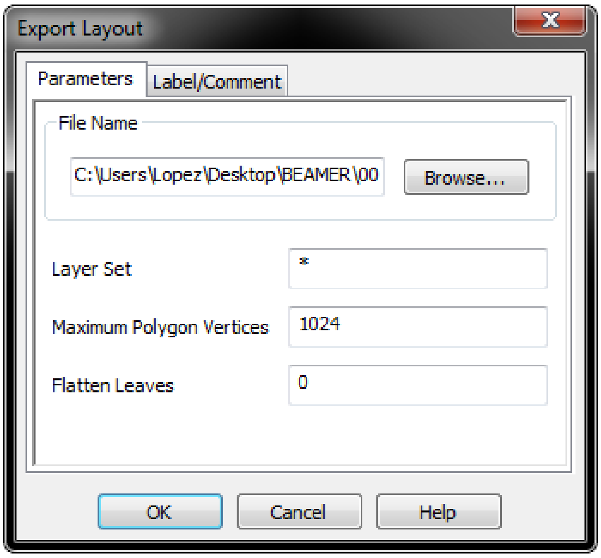
JEOL v52 is the machine format for newer JEOL Gaussian beam exposure systems:

- JBX-9000MV(50kV)
- JBX-9300FS(50kV)
- JBX-9300FS(100kV)
- JBX-6300SA
- JBX-6300FS/2
- JBX-5500

Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to JEOL licensed users.

4.2.4.11 Raith EBL

This is the machine format for the e-beam system from Raith. It is based on GDS including Raith specific modifications to retain shape dose assignments made from PEC. The following parameters for export as as follows:



File Name: C:\Users\Lopez\Desktop\BEAMER\00

Layer Set: *

Maximum Polygon Vertices: 1024

Flatten Leaves: 0

Layer Set

Defines the layers and datatypes to be imported or exported (depending if you are using the Import or Export module). This field can make use of the special annotations available for layer datatype handling.

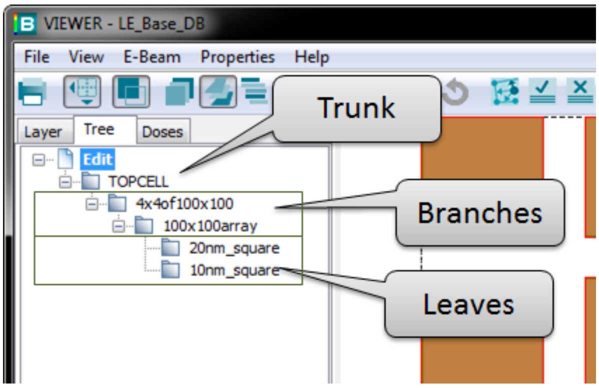
Maximum Polygon Vertices

Provides the maximum number of vertices in a polygon.

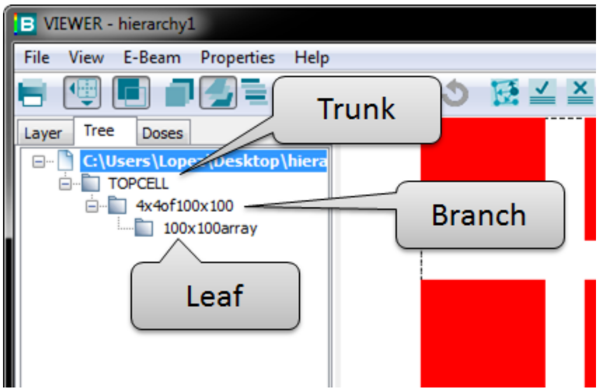
Flatten Leaves

The levels of the GDS hierarchy are similar to a tree with branches and leaves. The trunk of the tree is the topmost part of the hierarchy. When you expand the hierarchy the branches can be seen as cells in the hierarchy. These cells can be made up of other cells (or branches). Eventually you reach the bottommost elements that have no branches; these are the leaves. Leaves are the most primitive cells. **Flatten Leaves** indicates which part of the hierarchy that is flattened inward from the leaves. Below is an example GDS that has hierarchy and uses **Flatten Leaves**.

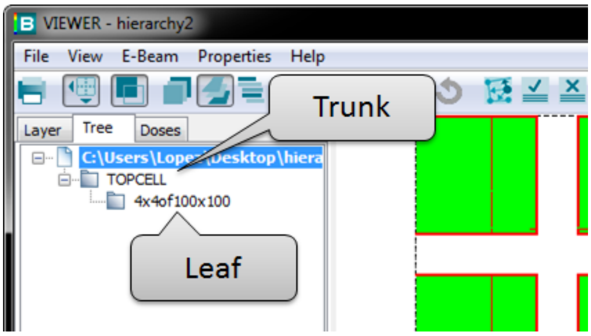
With **Flatten Leaves** = 0 (default value), the original hierarchy is preserved. No leaves are flattened.



Flatten Leaves = 1 will flatten the parent branch the original leaves, making the former branch (3x3Array) the bottommost level of the hierarchy (a new leaf). The original leaves 10nmDots and 20nmDot have been removed and flattened into the parent cell 3x3Array.



Flatten Leaves = 2 will flatten the two levels in from the original leaves into the parent branch 2x2ArrayOf3x3, making the former branch (2x2ArrayOf3x3) the bottommost level of the hierarchy (a new leaf).



4.2.4.12 CEL

This machine format is for Elionix e-beam systems. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Elionix licensed users.

4.2.4.13 CON

The CON machine format is for Elionix e-beam systems. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Elionix licensed users.

4.2.4.14 MEBES

This format option exports data in the MEBES mask making format.

Address Units defines the address unit or resolution of the pattern

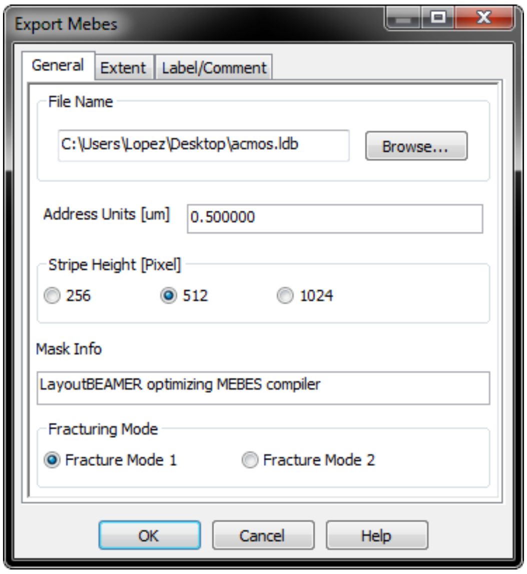
Stripe Height defines he height of the writing stripe in address units

- 256
- 512
- 1024

Mask Info is the text field that provides mask information in the exported pattern.

Fracture Mode defines the fracture algorithm:

- Fracture Mode 1 (Conventional)
- Fracture Mode 2 (LRFT)



File Name: C:\Users\Lopez\Desktop\acmos.ldb

Address Units [um]: 0.500000

Stripe Height [Pixel]: ○ 256 ● 512 ○ 1024

Mask Info: LayoutBEAMER optimizing MEBES compiler

Fracturing Mode: ● Fracture Mode 1 ○ Fracture Mode 2

4.2.4.15 VEP

VEP is the machine format for the VB6 vector beam systems from Vistec/Leica. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Vistec licensed users.

4.2.4.16 GPF

GPF is the machine format for the EBPG systems from Vistec/Leica. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Vistec licensed users.

4.2.4.17 FRE

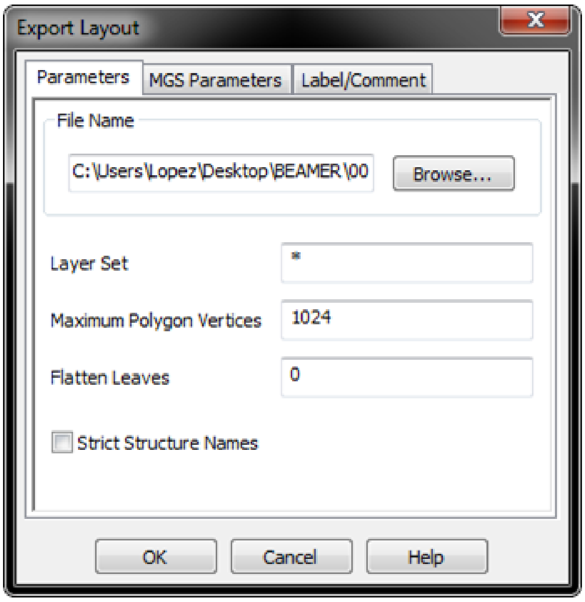
FRE is an older machine format for VB6 systems from Vistec/Leica. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Vistec licensed users.

4.2.4.18 IWFL

IWFL is an older machine format for EBPG systems from Vistec/Leica. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Vistec licensed users.

4.2.4.19 MGS GDS

This is a modified GDSII format with the capability to load proximity effect corrected data to the MGS system from Vistec. For PEC-ed layouts, the export includes a script which can be read by the MGS system and enables the correct mapping of the dose information.



File Name: C:\Users\Lopez\Desktop\BEAMER\00

Layer Set: *

Maximum Polygon Vertices: 1024

Flatten Leaves: 0

☐ Strict Structure Names

The following parameters need to be set:

Layer Set

Defines the layers and datatypes to be imported or exported (depending if you are using the Import or Export module). This field can make use of the special annotations available for layer datatype handling.

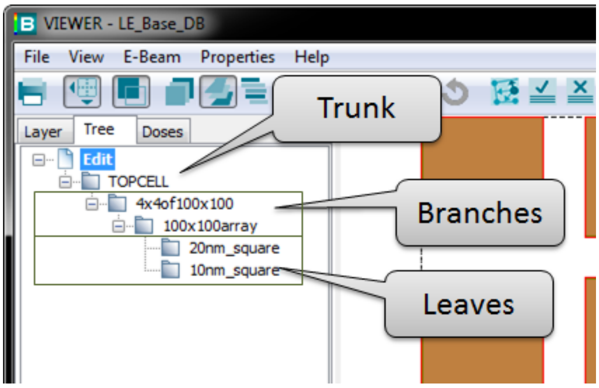
Maximum Polygon Vertices

Provides the maximum number of vertices in a polygon.

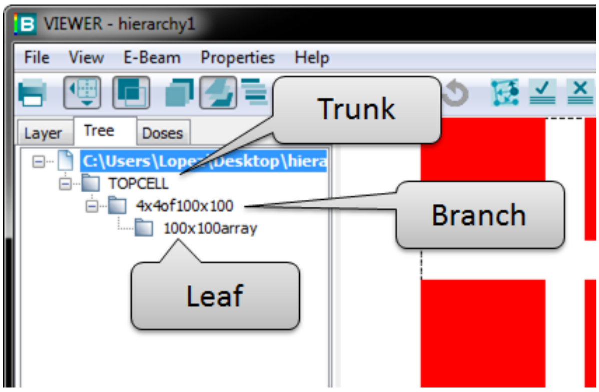
Flatten Leaves

The levels of the GDS hierarchy are similar to a tree with branches and leaves. The trunk of the tree is the topmost part of the hierarchy. When you expand the hierarchy the branches can be seen as cells in the hierarchy. These cells can be made up of other cells (or branches). Eventually you reach the bottommost elements that have no branches; these are the leaves. Leaves are the most primitive cells. **Flatten Leaves** indicates which part of the hierarchy that is flattened inward from the leaves. Below is an example GDS that has hierarchy and uses **Flatten Leaves**.

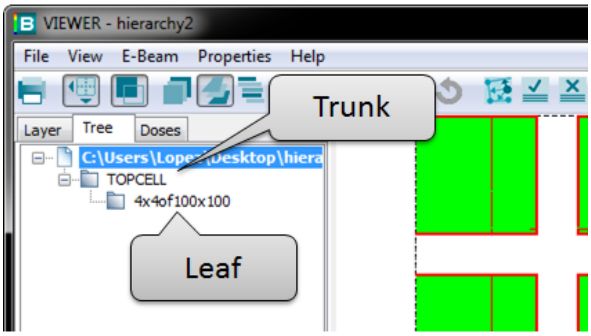
With **Flatten Leaves** = 0 (default value), the original hierarchy is preserved. No leaves are flattened.



Flatten Leaves = 1 will flatten the parent branch the original leaves, making the former branch (3x3Array) the bottommost level of the hierarchy (a new leaf). The original leaves 10nmDots and 20nmDot have been removed and flattened into the parent cell 3x3Array.



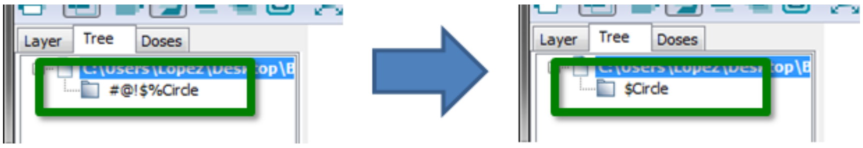
Flatten Leaves = 2 will flatten the two levels in from the original leaves into the parent branch 2x2ArrayOf3x3, making the former branch (2x2ArrayOf3x3) the bottommost level of the hierarchy (a new leaf).



Strict Structure Names

In the actual GDS specification, certain characters are not allowed in the hierarchical cell names. Enabling this feature removes characters not allowed in the original GDS specification before export.

The following example shows how a pattern with the cell name **#@!$%Circle** is converted to the cell name **$Circle** when Strict Structure Names is enabled:



4.2.4.20 NuFlare

This format option exports data in the NuFlare mask making format: VSB12. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Nuflare licensed users.

4.2.4.21 Heidelberg

This is the output format for the Heidelberg Instruments Laser Writer for the supported tools:

- DWL 2000
- DWL 66

Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to Heidelberg licensed users.

4.2.4.22 RSD

RSD is a format for IMS Nanofabrication writers. Please inquire for a detailed description for this format from GenISys Support. A detailed description can only be shipped to RSD licensed users.

4.2.5 Extract

This module allows for the extraction of layers, datatypes, cells, region layers and extent modification of a layout. Extraction operations are controlled in the Extract tab, while the Extent can be modified using the Extent tab. By default, the Extract module will extract the entire pattern while maintaining any layer/shape/dose information in addition to maintaining the original extent of the pattern.

4.2.5.1 Extract Tab

Extraction is handled first at the layer level followed by the extraction type (cell instances, cell definition or region). By default, all layers are extracted using the wildcard symbol (*) and the entire pattern is selected using the Cell Instances extraction type with the CellName = "--- entire layout ---". Should specific layers be typed in the Layer(s) field, only those layers will be extracted for the extraction type chosen. Each interface element is described in greater detail in this section.

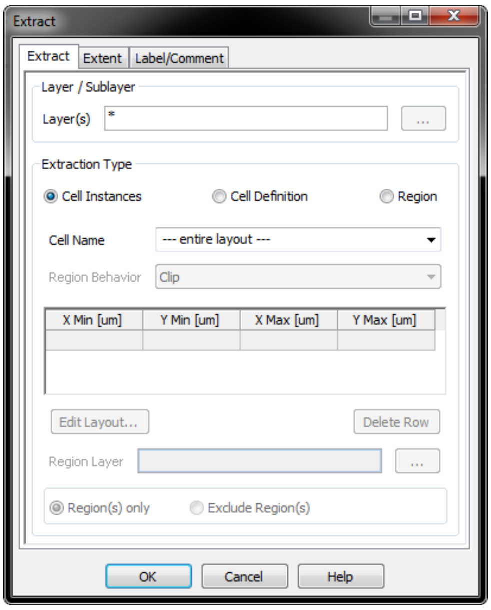
Layer(s)

The default setting, noted by the asterisk sign *, will load all layers and datatypes. To extract a specific layer or datatype, enter the specific value(s) in place of the asterisk sign. Multiple values or a range can be entered into the dialog as described in the following examples:

Layer(s): 5, 12
Will extract layers 5 and 12 from the layout.

Layer(s) 5-12,16
Will extract layers 5 through 12 and layer 16

For more information on special annotations available for this field, please read the section on layer datatype handling. To return to the default setting, enter the asterisk sign into the dialog, which will then load all available layers and datatypes.



Extract

Layer / Sublayer

Layer(s): *

Extraction Type: ● Cell Instances ○ Cell Definition ○ Region

Cell Name: --- entire layout ---

Region Behavior: Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

Region Layer:

○ Region(s) only ○ Exclude Region(s)

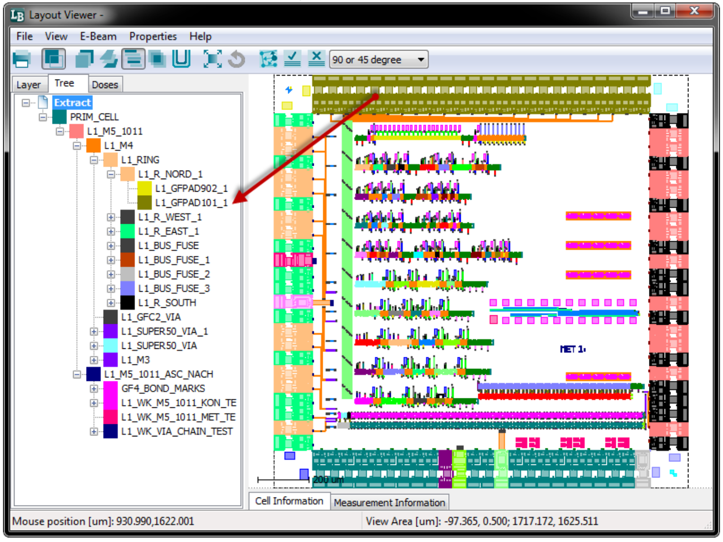
When the Extract is connected to an already executed module, the button, ..., will open a dialog displaying a list to select the layers interactively.

Extraction Types

Extraction by *Cell Name* or Region is available using the extract module. Extraction using a cell name is possible with hierarchical layouts by choosing the cell name from the drop-down menu when the either *Cell Instances* or *Cell Definition* extraction types are selected. The cell name can also be typed in the dropdown. In Windows, auto-completion of the cell name is supported; however, Linux does not support auto-completion of this field. Region selection is possible by manually defining regions of interest (ROI) using the mouse and keyboard, entering the coordinates of the ROI directly in the table or using Region Layers. Below are examples on how to use the Extract module.

Cell Instances

When selected, the module will extract all instances of a specified cell indicated in the *Cell Name* drop down. For example, in the pattern below, there are several instances of L1_GFPAD101_1 at the top of the layout.

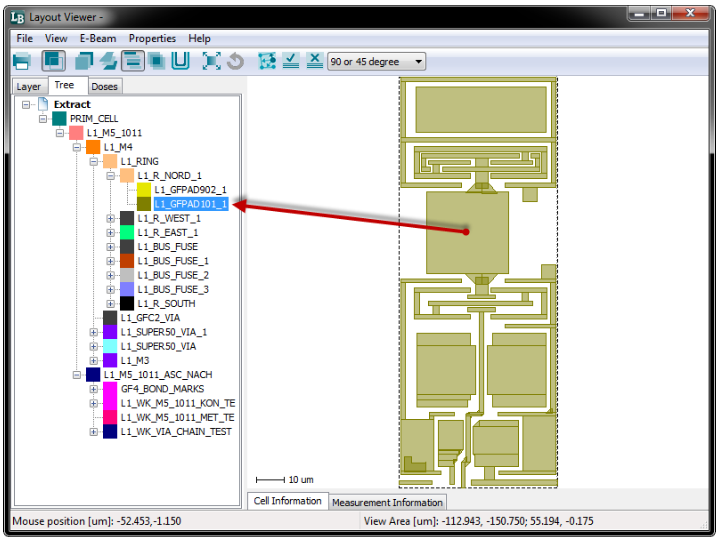


Layout Viewer -

Mouse position [um]: 930.990,1622.001

View Area [um]: -97.365, 0.500; 1717.172, 1625.511

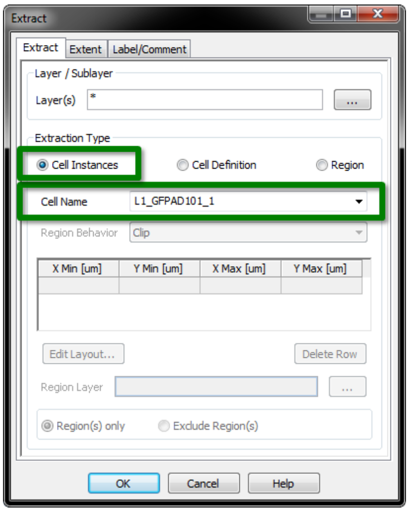
The actual cell of L1_GFPAD101_1 actually looks like the following:


Layout Viewer -
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
10 um
Cell Information Measurement Information

Extract
- PRIM_CELL
  - L1_M5_1011
    - L1_M4
      - L1_RING
        - L1_R_NORD_1
          - L1_GFPAD902_1
          - L1_GFPAD101_1
        - L1_R_WEST_1
        - L1_R_EAST_1
        - L1_BUS_FUSE
        - L1_BUS_FUSE_1
        - L1_BUS_FUSE_2
        - L1_BUS_FUSE_3
        - L1_R_SOUTH
      - L1_GFC2_VIA
      - L1_SUPER50_VIA_1
      - L1_SUPER50_VIA
      - L1_M3
    - L1_M5_1011_ASC_NACH
      - GF4_BOND_MARKS
      - L1_WK_M5_1011_KON_TE
      - L1_WK_M5_1011_MET_TE
      - L1_WK_VIA_CHAIN_TEST

Mouse position [um]: -52.453,-1.150 View Area [um]: -112.943, -150.750; 55.194, -0.175

If a user wanted to simply expose all instances of this cell, the export module configuration would have Cell Instances selected and the Cell Name set to L1_GFPAD101_1 like on the bottom left. The result will have the following layout to the bottom right.



Layer / Sublayer
Layer(s) *

Extraction Type
● Cell Instances ○ Cell Definition ○ Region

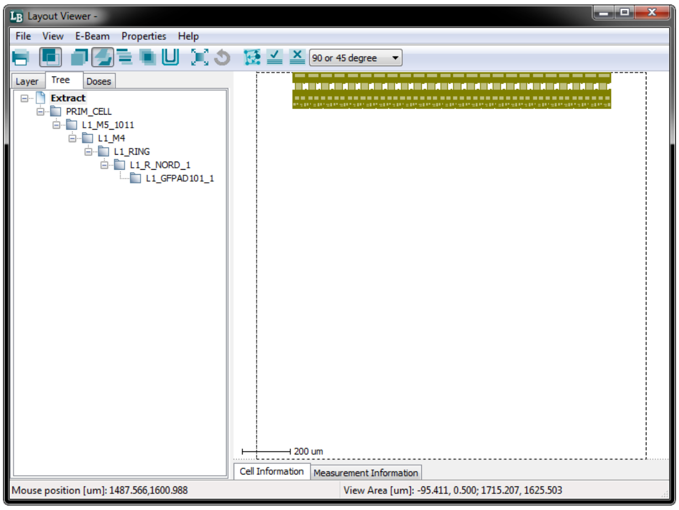
Cell Name L1_GFPAD101_1

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

Region Layer

● Region(s) only ○ Exclude Region(s)

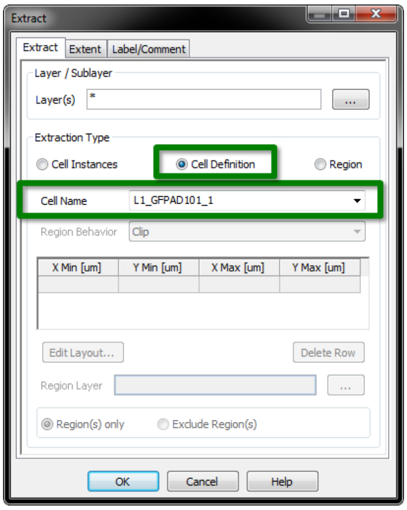

Layout Viewer -
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
200 um
Cell Information Measurement Information

Extract
- PRIM_CELL
  - L1_M5_1011
    - L1_M4
      - L1_RING
        - L1_R_NORD_1
          - L1_GFPAD101_1

Mouse position [um]: 1487.566,1600.988 View Area [um]: -95.411, 0.500; 1715.207, 1625.503

Cell Definition

When selected, the module will extract only the design cell selected from the *Cell Name* drop down. For example, if the Extraction Type was set to Cell Definition and the Cell Name was set to L1_GFPAD101_1 (seen below left), only the cell would be extracted (below right).



Layer / Sublayer
Layer(s) *

Extraction Type
○ Cell Instances ● Cell Definition ○ Region

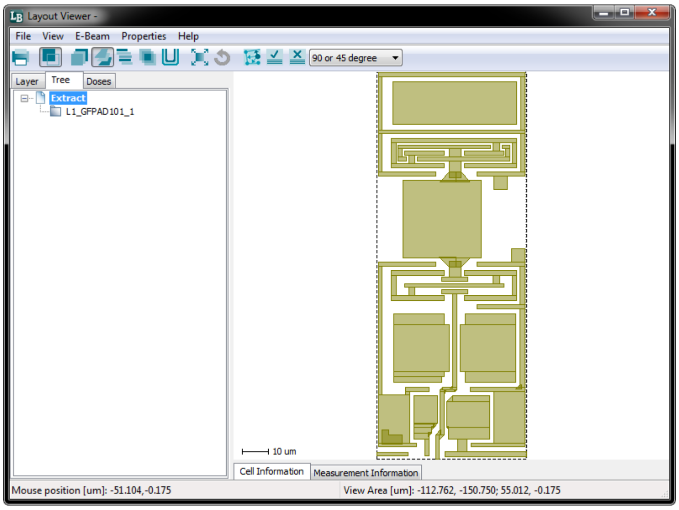
Cell Name L1_GFPAD101_1

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

Region Layer

● Region(s) only ○ Exclude Region(s)

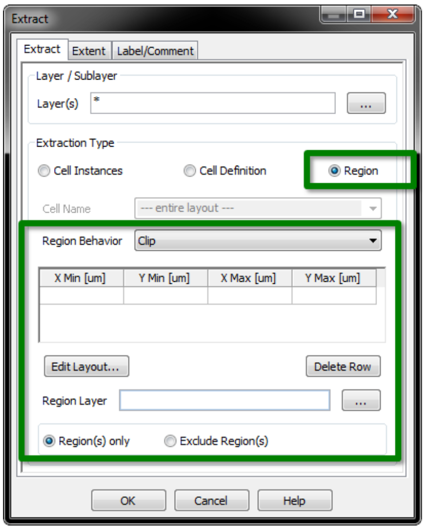

Layout Viewer -
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
10 um
Cell Information Measurement Information

Extract
- L1_GFPAD101_1

Mouse position [um]: -51.104,-0.175 View Area [um]: -112.762, -150.750; 55.012, -0.175

Region

Use this option to extract a user specified ROI from the layout. When *Region* is selected, the *Region Behavior* dropdown, ROI table, *Edit Layout...* button, *Delete Row* button, *Region Layer* field and the radio buttons, Region(s) only and Exclude Region(s) are enabled.


Extract

Extract | Extent | Label/Comment

Layer / Sublayer

Layer(s) * ...

Extraction Type

○ Cell Instances ○ Cell Definition ● Region

Cell Name --- entire layout ---

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

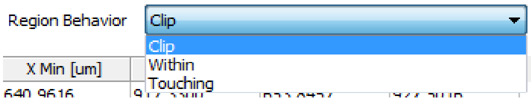
Edit Layout... Delete Row

Region Layer ...

● Region(s) only ○ Exclude Region(s)

OK Cancel Help

Region Behavior controls the behavior for elements that cross the boundaries of the a defined region boundary.


Region Behavior Clip

Clip

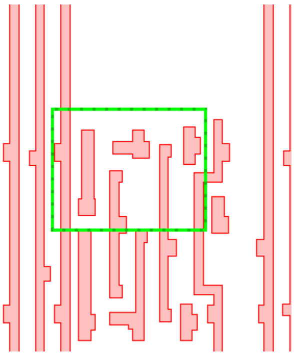
Within

Touching

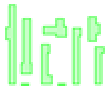
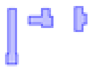
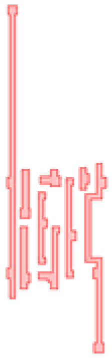
X Min [um]

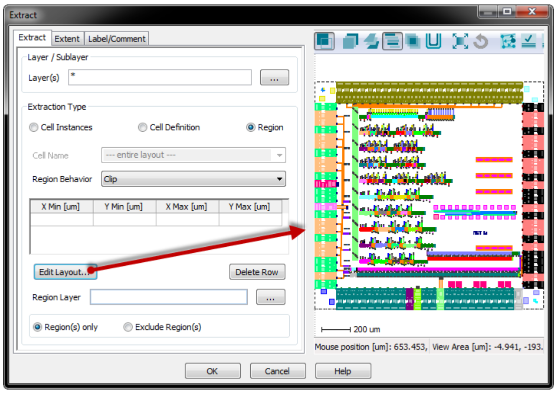
640.9616

Quick examples of the Region Behaviors are illustrated in the following. Consider the green bounding box atop the pattern in red (we will show you how to selected a region like this shortly).



Depending on the Region Behavior, the following extractions will occur:

		
Clip cuts the elements at the region boundary.	**Within** ignores the element that crosses the region boundary and extracts only those elements whose boundaries are completely within the frame.	**Touching** extracts all elements within the region and that are crossing the frame of the extraction box.



Extract

Layer / Sublayer

Layer(s) *

Extraction Type

○ Cell Instances ○ Cell Definition ● Region

Cell Name --- entire layout ---

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

Region Layer

● Region(s) only ○ Exclude Region(s)

Mouse position [um]: 653.453, View Area [um]: -4.941, -193.

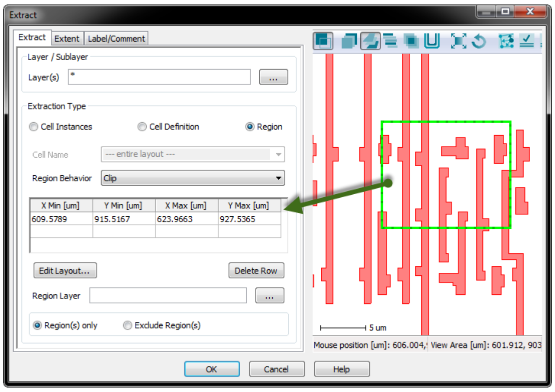
An ROI can be defined in one of three ways:

1. Manually entering coordinates into the ROI Table,
2. Defining regions of interest (ROI) using the mouse and keyboard or
3. Using *Region Layer(s)* defined in the pattern.

It is possible to use both the ROI table and the region layer(s) defined in the layout at the same time for a region extraction.

Extracting a region using the mouse and keyboard is performed by clicking the **Edit Layout...** button. Be sure that the module before the Extract module has been ran so that the **Edit Layout...** button is enabled. When this button is clicked, an inline VIEWER will appear as shown in the image above.To select a region inside the VIEWER:

1. Set starting corner of bounding box by holding the SHIFT key on the keyboard and clicking once with the left mouse button in the VIEWER.



Extract

Layer / Sublayer

Layer(s) *

Extraction Type

○ Cell Instances ○ Cell Definition ● Region

Cell Name --- entire layout ---

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
609.5789	915.5167	623.9663	927.5365

Region Layer

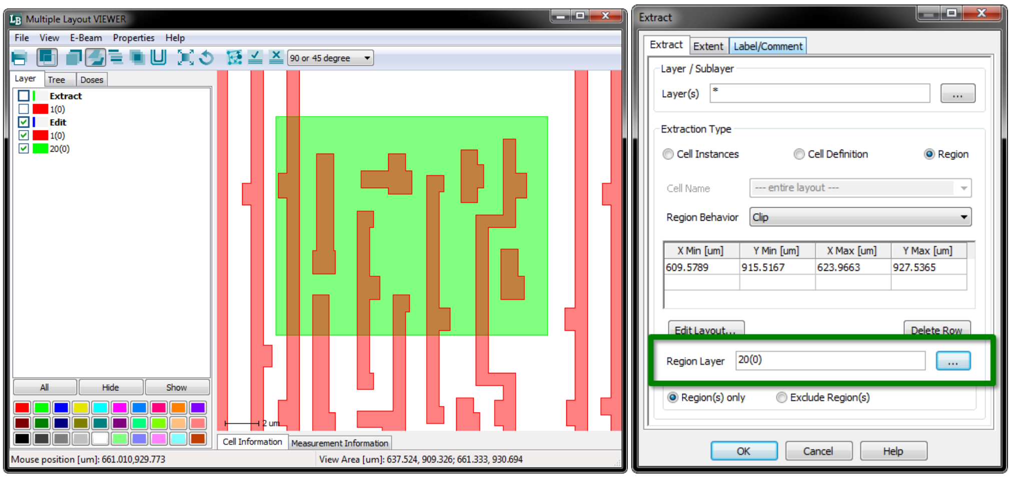
● Region(s) only ○ Exclude Region(s)

Mouse position [um]: 606.004, View Area [um]: 601.912, 903

2. Let go of the SHIFT key (and left mouse button) and move the mouse in the VIEWER to set the size of the bounding box.

3. Once you have moved the mouse to the desired size, confirm the size by holding the SHIFT key and clicking once with the left mouse button while maintaining the mouse position within the VIEWER. The coordinates of the green bounding box will be placed in the ROI Table as indicated with the arrow in the image.

As mentioned, ROIs can be defined by a layer within the pattern. This layer is referred to as a *Region Layer*. In the next example, layer 20(0), the green layer, will be used as the region layer. To use this layer as a region layer, enter the layer 20(0) in the Region Layer(s) text field.



Multiple Layout VIEWER

☐ Extract
☐ 1(0)
☑ Edit
☑ 1(0)
☑ 20(0)

Mouse position [um]: 661.010,929.773

View Area [um]: 637.524, 909.326; 661.333, 930.694

Extract

Layer / Sublayer

Layer(s) *

Extraction Type

○ Cell Instances ○ Cell Definition ● Region

Cell Name --- entire layout ---

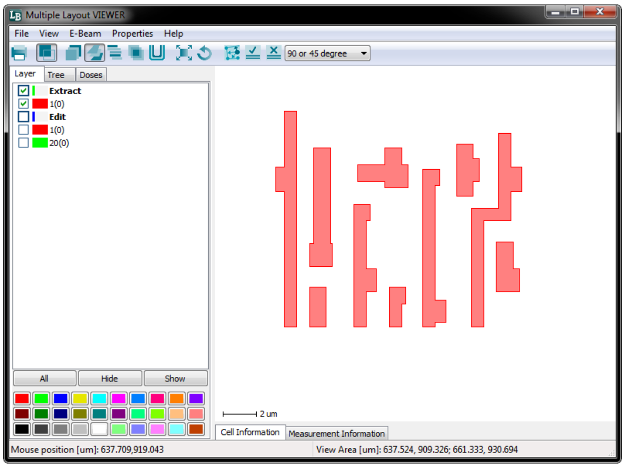
Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
609.5789	915.5167	623.9663	927.5365

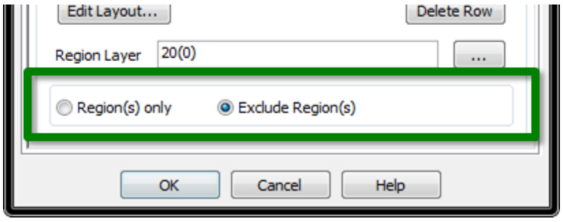
Region Layer 20(0)

● Region(s) only ○ Exclude Region(s)

As a result, any shapes on layer 20(0) will be taken out of the pattern and their position and size will be used as bounding boxes (a.k.a. ROIs) to extract the data. Below, the region behavior is set to *Clip:*


Multiple Layout VIEWER
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
Extract
1(0)
Edit
1(0)
20(0)
All Hide Show
2 um
Cell Information Measurement Information
Mouse position [um]: 637.709,919.043
View Area [um]: 637.524, 909.326; 661.333, 930.694

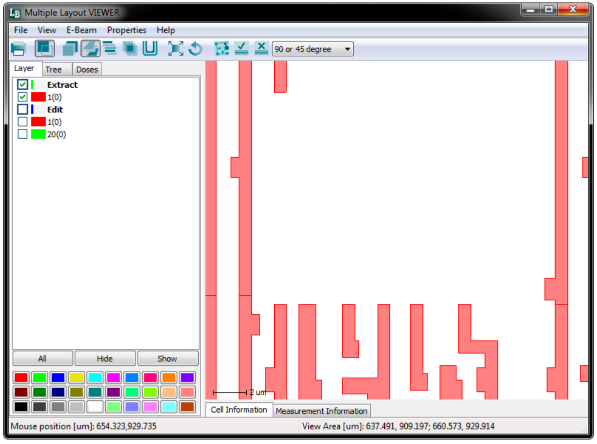
So far, the region extraction examples have been shown using the **Region(s) only** mode. The option **Exclude Region(s)** will do the opposite: it will remove the ROIs from the pattern:



Region Layer 20(0)

○ Region(s) only ● Exclude Region(s)

Using the previous example but with *Exclude Regions* selected, the result will remove the area within the ROI (in this case that is defined by layer 20):


Multiple Layout VIEWER
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
Extract
1(0)
Edit
1(0)
20(0)
All Hide Show
2 um
Cell Information Measurement Information
Mouse position [um]: 654.323,929.735
View Area [um]: 637.491, 909.197; 660.573, 929.914

Here the region was removed using the region behavior *Clip.* Alternative region behaviors (Within and Touching) could have been used to remove the data as well.

4.2.5.2 Extent Tab

The extent of the layout is a box from the lower left limit to the upper right limit of the originally designed data. In other words, it is the furthest edge of a pattern and is considered to be the writable area of the pattern. This means that no shapes can exist outside the extent boundary. The extent is not to be used to extract an area or region of interest for exposure. It is intended to only define the area to expose. By default, the original extent of a pattern is always maintained when using the Extract module.

The extent of the extracted region can be defined in the following ways:

Maintain

Keeps the same extent as the original layout.

Minimum

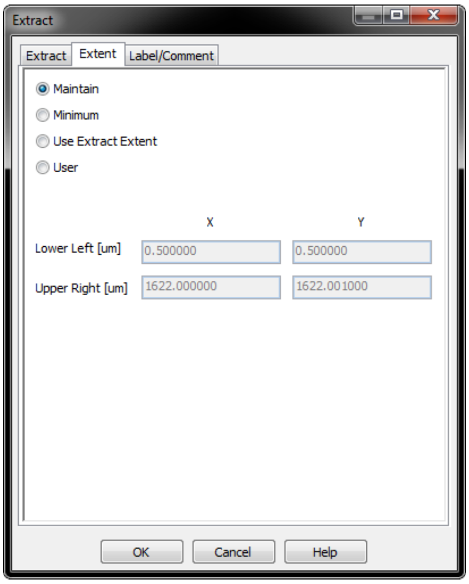
Takes the minimum dimensions possible, surrounding all polygons.

Use Extract Extent

Uses the extent given by the rectangle defined for the extraction.

User

The user can specify the lower left and upper right coordinates. When defining an extent, it is only possible adjust the User extent to be larger than the furthest edge of the data. It is not possible to make it smaller.

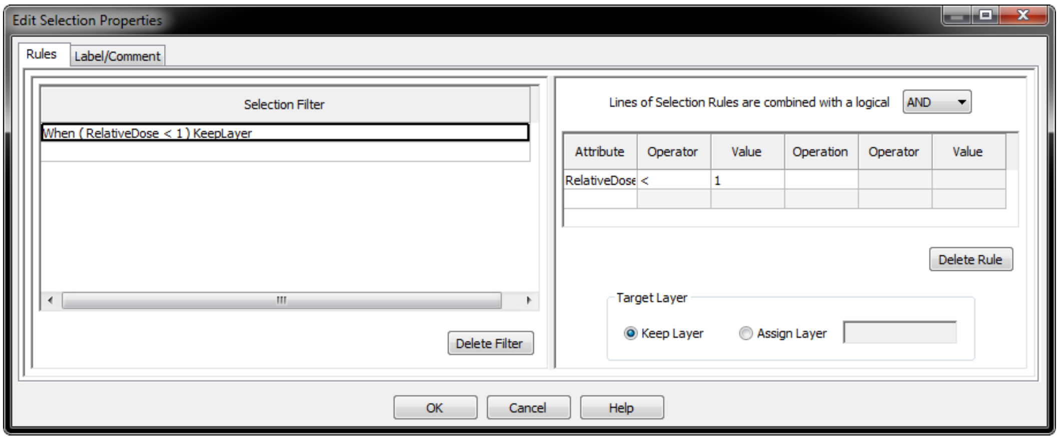


● Maintain
○ Minimum
○ Use Extract Extent
○ User

	X	Y
Lower Left [um]	0.500000	0.500000
Upper Right [um]	1622.000000	1622.001000

4.2.6 Filter

The Filter module is driven by the Rules tab that contains a left and right pane as shown above. The left pane contains the Selection Filter table that lists one or many kinds of filtering criteria. The right pane contains the definition of the filtering criteria that is set by the user.



Selection Filter
When (RelativeDose < 1) KeepLayer

Lines of Selection Rules are combined with a logical AND

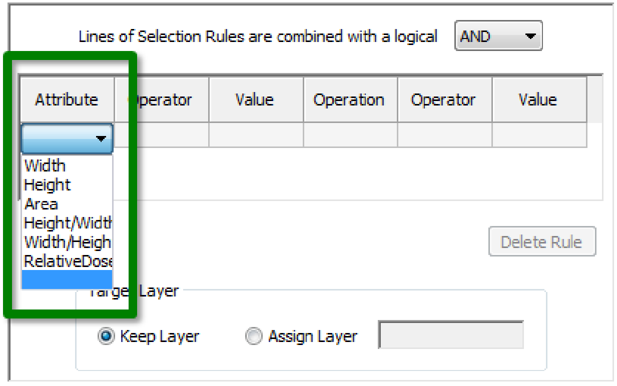
Attribute	Operator	Value	Operation	Operator	Value
RelativeDose	<	1			

Target Layer
● Keep Layer ○ Assign Layer

By default, the first row in the Selection Filter table is selected. This means that a user can define the filtering criteria in the right pane. Basically, the Filter module allows the selection of shapes within a layout via the following element-specific attributes listed in the table below. Some of the filtering attributes rely on the on the attributes of the bounding box of the shapes within the layout. The size of a bounding box is identical for rectangles. However, polygons will have a bounding box that can be somewhat different from the design.

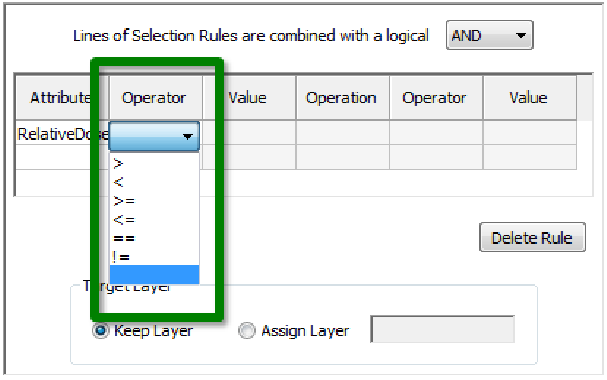
Attribute	Definition
Width	This is the width of a shape within a layout. The width is the end-to-end distance parallel to the X-axis.
Height	This is the height of a shape within a layout. The width is the end-to-end distance parallel to the Y-axis.
Area	This is the area of a shape within a layout.
Height/Width	This is the ratio calculated by dividing the height of the shape by its width.
Width/Height	This is the ratio calculated by dividing the width of the shape by its height.
Relative Dose	This is the dose assigned to the shape by PEC or FDA

A user can select one of the attributes in the table in the right pane via a dropdown.



Lines of Selection Rules are combined with a logical AND

After selecting an attribute, an operator can be chosen in the next column to the right:

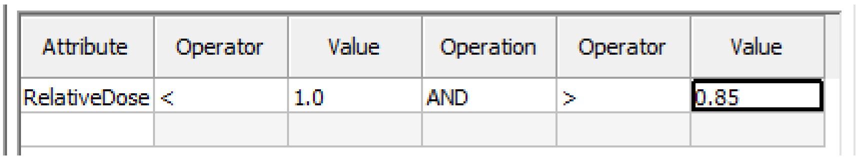


Lines of Selection Rules are combined with a logical AND

The operators below are then applied to any of the above attributes to define the criteria on which to filter out shapes:

Symbol	Definition
<	Less than
>	Greater than
<=	Less than or equal to
=>	Greater than or equal to
==	Equal to
!=	Not Equal

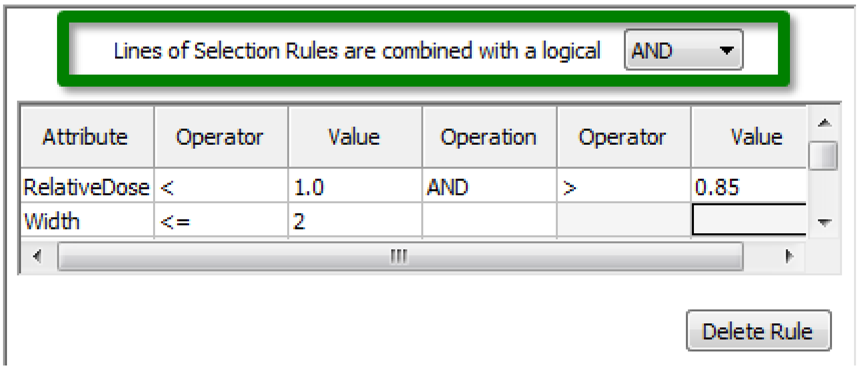
For example, if a user wanted to filter out all the shapes that had a relative dose less than 1.0 and greater than 0.85, the selection rule would like the the following in the right pane:



Attribute	Operator	Value	Operation	Operator	Value
RelativeDose	<	1.0	AND	>	0.85

If the operation for this selection criteria was OR instead of AND, all the shapes would be selected.

Moreover, if the user wanted to add additional criteria like restricting the width of the shapes to be less than or equal to 2um, the table would look like the following:



Lines of Selection Rules are combined with a logical AND

Attribute	Operator	Value	Operation	Operator	Value
RelativeDose	<	1.0	AND	>	0.85
Width	<=	2			

The selected attributes here are joined with an AND operators. This means that any shapes that have an assigned relative dose between 0.85 and 1.0 AND are less than or equal to 2um will be filtered as the result. If OR was chosen, any shapes that have an assigned relative dose between 0.85 and 1.0 OR have a width less than or equal to 2um will be filtered as the resulting pattern. To summarize thus far, individual attributes can be checked against two rules with the AND or OR operator, where both rules must be true or where only one rule needs to be true, respectively. To make sure that all selection criteria are true, AND must be selected at the top of the table, otherwise, should any be true, OR must be selected.

Finally, the **Target Layer** defines what layer the filtered shapes should be placed.

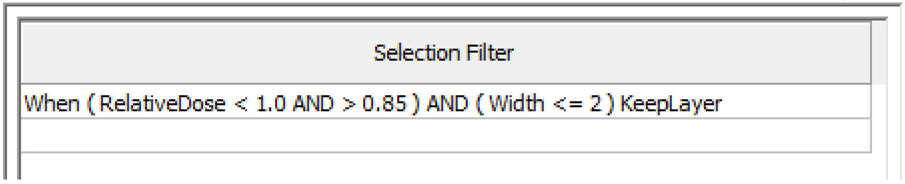
Keep Layer

The layer on which the shapes already reside will be maintained when the shapes are filtered.

Assign Layer

The layer on which the shapes already reside will be changed to the specified layer when the shapes are filtered.

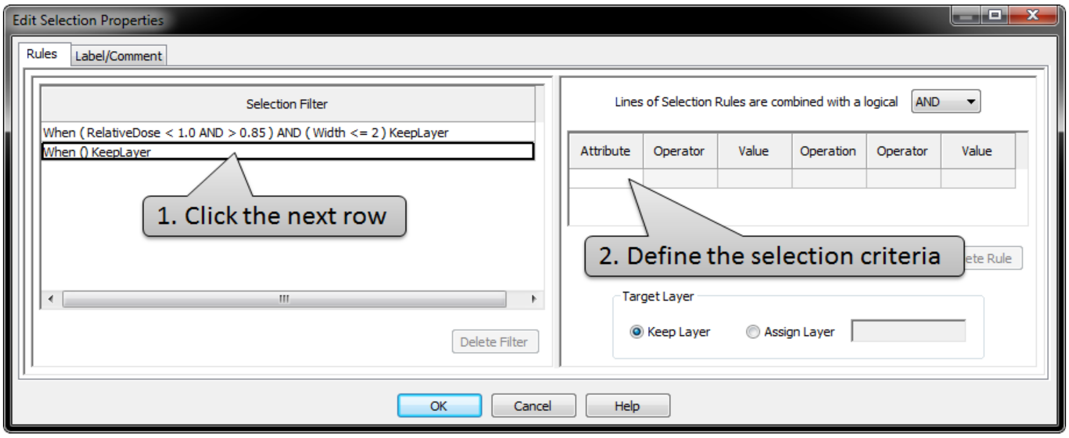
With the selection criteria defined, the table in the left pane will look like the following:



Selection Filter
When (RelativeDose < 1.0 AND > 0.85) AND (Width <= 2) KeepLayer

This is intended to be human-readable selection filter description. In this case, any shape that has a relative dose less than 1.0 and greater than 0.85 AND a width that is less than or equal to 2um will be filtered as the result and will remain on the same layer as in the original pattern prior to filtering.

To add additional selection filters, click the row below the defined selection criteria and edit the right pane to define the next filter.



Selection Filter

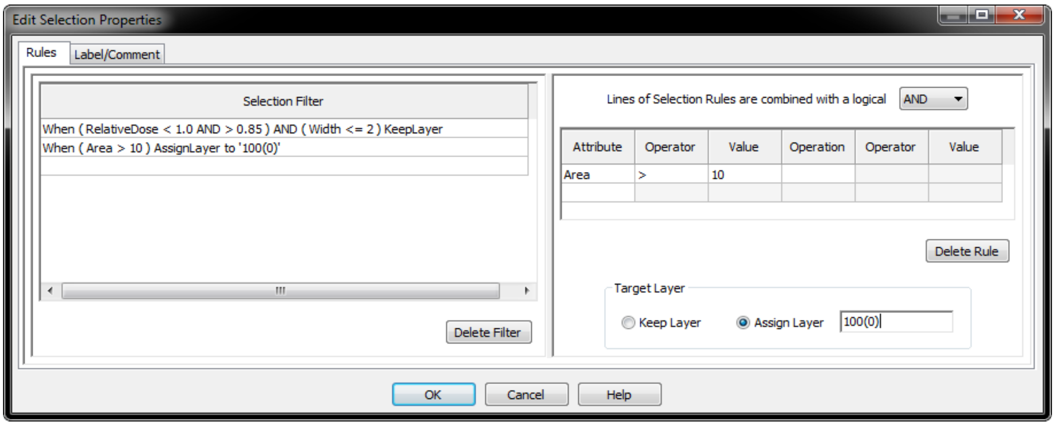
When (RelativeDose < 1.0 AND > 0.85) AND (Width <= 2) KeepLayer
When () KeepLayer

Lines of Selection Rules are combined with a logical AND

Attribute	Operator	Value	Operation	Operator	Value

Target Layer: ● Keep Layer ○ Assign Layer

Here a second selection filter has been added such that any shape that has an area greater than 10um^2 will be extracted and put to layer 100(0):



Selection Filter

When (RelativeDose < 1.0 AND > 0.85) AND (Width <= 2) KeepLayer
When (Area > 10) AssignLayer to '100(0)'

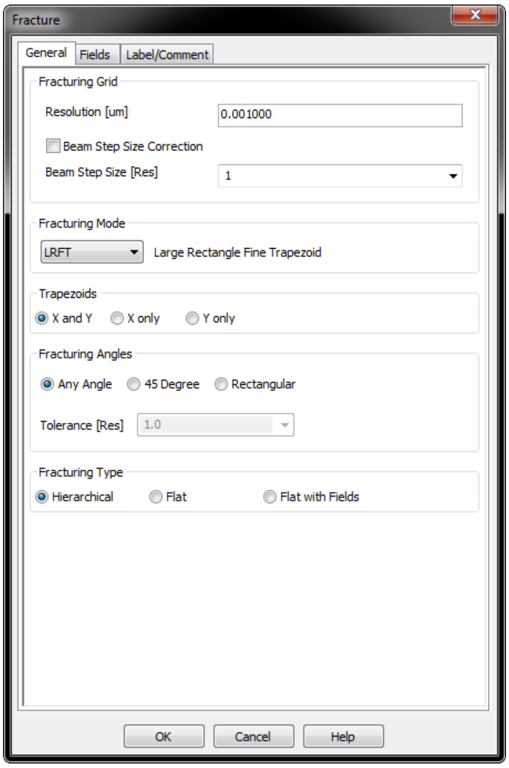
Lines of Selection Rules are combined with a logical AND

Attribute	Operator	Value	Operation	Operator	Value
Area	>	10			

Target Layer: ○ Keep Layer ● Assign Layer 100(0)

Be forewarned that multiple selection filters can result in the same shape being extract more than once. As a result, careless data prep may result in poor lithography due to overexposure.

4.2.7 Fracture



Fracturing Grid
Resolution [um]: 0.001000
□ Beam Step Size Correction
Beam Step Size [Res]: 1

Fracturing Mode
LRFT — Large Rectangle Fine Trapezoid

Trapezoids
● X and Y ○ X only ○ Y only

Fracturing Angles
● Any Angle ○ 45 Degree ○ Rectangular
Tolerance [Res]: 1.0

Fracturing Type
● Hierarchical ○ Flat ○ Flat with Fields

The Fracture module renders more control to the user by allowing machine independent pre-fracturing of layout data with special attention to the algorithm and fracturing angles that are used. The module can be inserted at any position of a flow to fracture any data that is passed to it. Extended functionality supports pre-fracturing with respect to field size to mitigate statistical error (e.g. beam positional accuracy, jitter, etc.) and systematic errors (e.g. field stitching).

As an overview, this module provides:

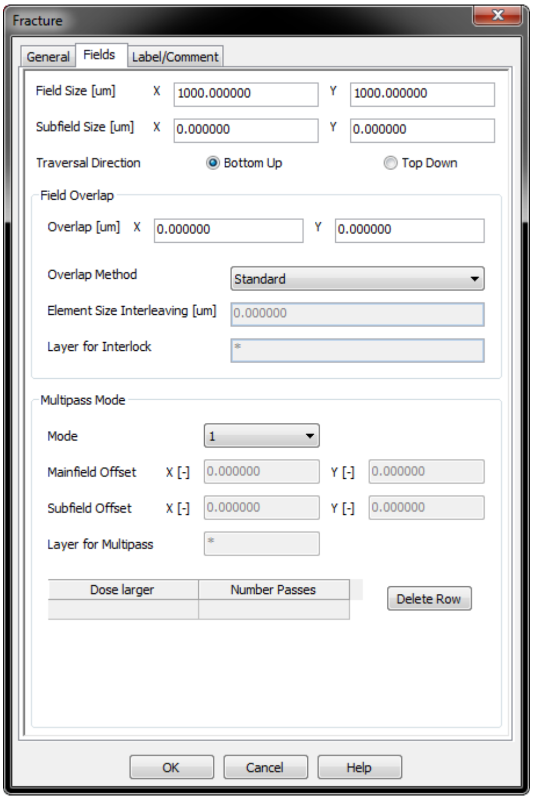
- Machine independent fracturing using any resolution and beam step size combination
- Beam step size fracturing
- Insertion at any position in a flow, which implies
  - The ability to fracture parts of a layout with different settings and re-merge those parts for export and
  - Pre-fracturing before PEC
- Various fracture modes: Conventional, LRFT and Curved
- Selection for trapezoid and fracturing angle types
- Fracturing limitation to rectangles (Manhattan geometry)

The General tab consists of 5 sub-panels:

Sub-Panel Name	Field Name and Definition
Fracturing Grid	**Resolution [um]** This is the fracturing resolution. In most cases this is the database grid of the design. This value should be set to the writing grid resolution (a.k.a. pattern units or fracturing resolution) of the tool. **Beam Step Size [Res]** This is the beam step size (or shot pitch) parameter as a multiple of resolution. It is used for 1. The polygonal approximation resolution of circular structures; 2. The approximation resolution for the 45° and rectangular fracturing mode; and 3. The beam step size correction. **Beam Step Size Correction** (Shot Pitch Fracturing) This is an alignment strategy to preserve original shape outlines. Beam Step Size Correction is as known as Shot Pitch Fracturing or Beam Step Size Fracturing.
Fracturing Mode	**Fracture Modes** The Fracture Mode defines the fracturing strategy to be used to fracture a layout. See Fracture Modes in the Application section for more information. Three fracturing modes available in BEAMER are: *Conventional* The conventional algorithm performs mainly horizontal fractures at polygon vertices that maintain the original design intent as good as possible. This leads to many slices of trapezoids to best represent

	the original layout. *LRFT* (Large Rectangle Fine Trapezoid) The LRFT algorithm attempts to minimize the number of shapes using very large rectangles for large areas and fine trapezoids for smaller features. In other words, this mode fractures the polygons with as many large rectangles as possible and completes the filling with small trapezoids. *Curved* This algorithm should typically be used to fracture circles and rings. This mode allows BEAMER to shift the vertices along curves to get fractures consistent with the specified resolution and beam step size. In other words, this mode will try to detect curves in the given polygons and fracture these curves within a specified tolerance consistent with the specified resolution and beam step size. It is always necessary to inspect the fracturing, since every mode may yield a dramatically different result. The fracture mode default is set based on the tool manufacturers recommendation. **Tolerance [Res]** This is available for *Curved* fracturing to define the approximation scope for detecting rings and circles.
Trapezoids	Primitive trapezoids that are generated can have an orientation that can be both *X and Y*, *X only* and *Y only*.
Fracturing Angles	**Fracturing Angles** limits how the fracturing algorithm can draw trapezoid corners using *Any Angle*, *45 Degree* angles, or *Rectangular* (90 Degrees). **Tolerance [Res]** The tolerance defines the approximation scope for the angles that are replaced with either a 45 degree angled trapezoid or rectangle.
Fracturing Type	**Hierarchical** This mode will preserve the hierarchical layout structure. NOTE: The hierarchical fracturing mode does not consider rounding errors in structure placement for arrays and nested references. **Flat** This mode will remove the hierarchy in the pattern while preserving the layer information of the shapes. **Flat with Fields** This mode will enable the use of the the Fields tab. By flattening the pattern with fields, the original hierarchy is removed and replaced with a hierarchy where the virtual fields are the cells that make up the pattern. The maximum size of each cell is the size of a field, which can be controlled in the Field tab.

When the Fracturing Type *Flat with Fields* is selected, field control operations can be performed independent of a machine format in the Fields tab. The Fields tab allows a user to fracture a pattern at the field and subfield level. As a result, the pattern is separated into fields, where each field is a cell in the pattern. Advanced operations like overlapping fields, interleaving fields and multipass can be managed independent of a machine format.



Field Size [um] X 1000.000000 Y 1000.000000

Subfield Size [um] X 0.000000 Y 0.000000

Traversal Direction ● Bottom Up ○ Top Down

Field Overlap

Overlap [um] X 0.000000 Y 0.000000

Overlap Method Standard

Element Size Interleaving [um] 0.000000

Layer for Interlock *

Multipass Mode

Mode 1

Mainfield Offset X [-] 0.000000 Y [-] 0.000000

Subfield Offset X [-] 0.000000 Y [-] 0.000000

Layer for Multipass *

Dose larger	Number Passes

Field Size [um]

This is the intended field size that can be used for the machine export. By default the size is set to 1000um.

Subfield Size [um]

This is the intended sub-field size to be used for the machine export. The subfield size values in X and Y are only used for multipass fracturing.

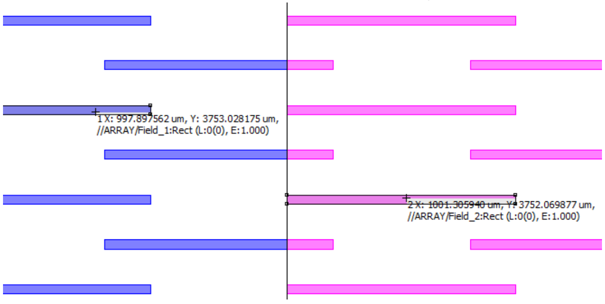
Traversal Direction

Traversal Direction impacts the order of the fields. The **Bottom Up** option orders the field cells in a Boustrophedon fashion starting from the bottom left. The **Top Down** option orders the field cells in a Boustrophedon fashion starting from the top left.

4.2.7.1 Overlapping Fields

Field overlap reduces fractures at main field boundaries in the Fixed Fields mode. In case of no overlap, all features crossing the field borders will have to be cut at the field boundary. The overlap parameter defines a common region between the main fields where fractures (trapezoids split into two main fields) are avoided if the feature can be exposed completely in either one of the fields. The maximum overlap is half of the field size.

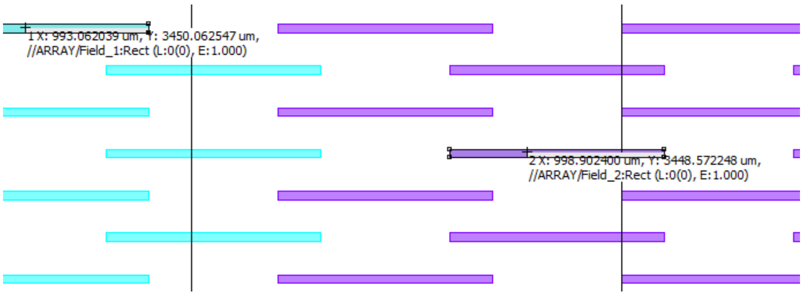
Consider the example where at the field boundary of Field 1 and Field 2, cutting of shapes occur for the following pattern. Assume the mainfield size is 1000um by 1000um square.


1 X: 997.897562 um, Y: 3753.028175 um,
//ARRAY/Field_1:Rect (L:0(0), E:1.000)
2 X: 1001.305940 um, Y: 3752.069877 um,
//ARRAY/Field_2:Rect (L:0(0), E:1.000)

One can specify a field overlap in the X and Y direction as shown by going to the Tab:

Overlap [um] X 5 Y 5

The overlap results in shapes being completely placed in one field, avoiding cutting at the field boundaries:


1 X: 993.062039 um, Y: 3450.062547 um,
//ARRAY/Field_1:Rect (L:0(0), E:1.000)
2 X: 998.902400 um, Y: 3448.572248 um,
//ARRAY/Field_2:Rect (L:0(0), E:1.000)

With field overlap enabled, a physical stage move from one field to the next is changed from 1000um to 995um, making field stitch prevention at the field boundaries possible.

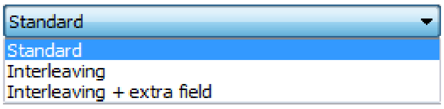
4.2.7.2 Field Overlap

Overlap

This specifies the overlap between fields for a fixed field traversal. More information can be found in the section on Overlapping Fields should the *Overlap Method* chosen is *Standard*.

Overlap Method

This specifies the type of overlap between fields for a fixed field traversal.

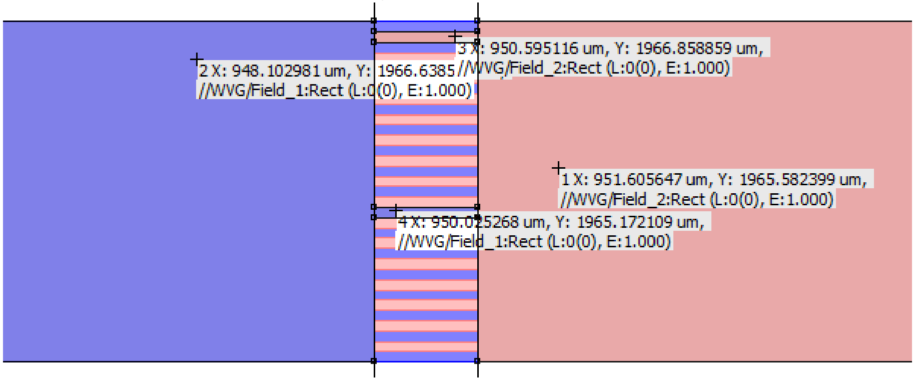


Standard

This default mode does not invoke a multipass writing strategy. Instead it overlaps fields enabling data to be completely written in one field or another. More information can be found in the section on Overlapping Fields for the *Standard* overlap method.

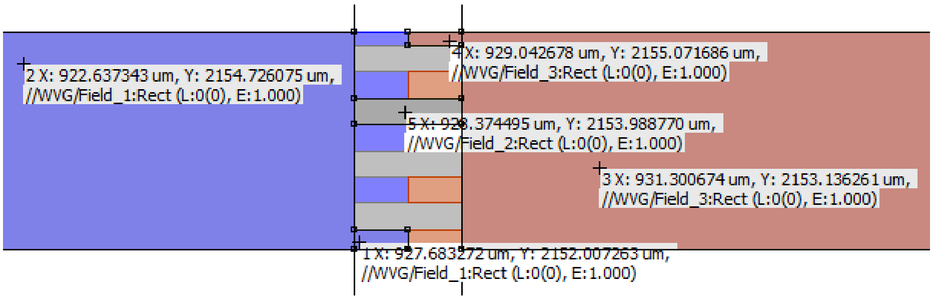
Interleaving

Interleaves elements that would normally be cut between two fields with an overlap:


3 X: 950.595116 um, Y: 1966.858859 um,
//WVG/Field_2:Rect (L:0(0), E:1.000)
2 X: 948.102981 um, Y: 1966.6385
//WVG/Field_1:Rect (L:0(0), E:1.000)
1 X: 951.605647 um, Y: 1965.582399 um,
//WVG/Field_2:Rect (L:0(0), E:1.000)
4 X: 950.025268 um, Y: 1965.172109 um,
//WVG/Field_1:Rect (L:0(0), E:1.000)

Interleaving + extra field

Introduces an additional field to add additional interleaving elements which 50% of the area overlapping area. For every field overlap, there is a third floating field that writes the remaining 50% of the interleaving region.


2 X: 922.637343 um, Y: 2154.726075 um,
//WVG/Field_1:Rect (L:0(0), E:1.000)
4 X: 929.042678 um, Y: 2155.071686 um,
//WVG/Field_3:Rect (L:0(0), E:1.000)
5 X: 928.374495 um, Y: 2153.988770 um,
//WVG/Field_2:Rect (L:0(0), E:1.000)
3 X: 931.300674 um, Y: 2153.136261 um,
//WVG/Field_3:Rect (L:0(0), E:1.000)
1 X: 927.683272 um, Y: 2152.007263 um,
//WVG/Field_1:Rect (L:0(0), E:1.000)

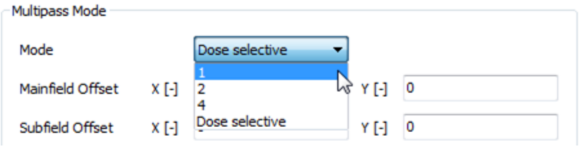
Element Size Interleaving

This specifies the height of the bars in the overlap.

Layer for Interlock

This is the layer for which to apply the *Interleaving Overlap* or *Interleaving Overlap + extra field*. For example, say a pattern has layer 0, 1, and 2. The most critical structures are on layer 0. Therefore, the user can decide to exclusively overlap with the interleaving structures for only layer 0.

4.2.7.3 Multipass Mode


Multipass Mode

Mode: Dose selective

Mainfield Offset X [-] Y [-] 0

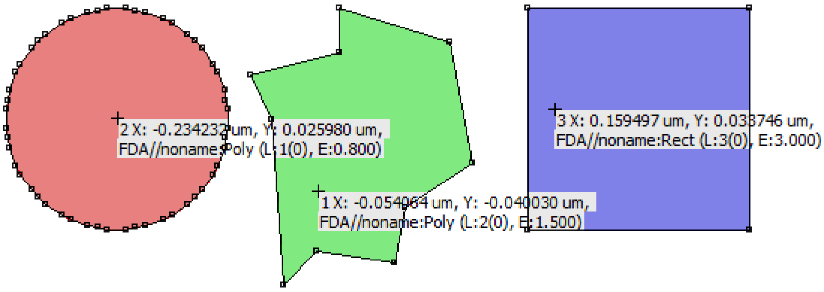
Subfield Offset X [-] Y [-] 0

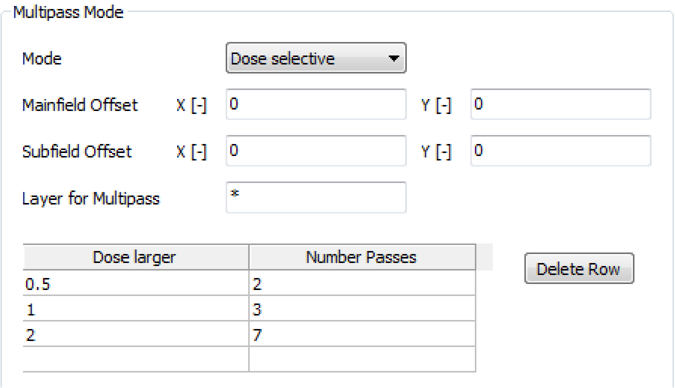
Mode

The number of passes can be 1 (default), 2 or 4. When using 2 passes, the base dose should be set to half of the base dose in a single pass. Likewise, when using 4 passes, the base dose should be set to a quarter of the base dose in a single pass.

Dose selective

When using the *Dose selective* mode, this will enable the table below the *Layer for Multipass* field. In the table, the user can specify that for any dose larger than a specific value, the number of passes will be performed on that shape whose dose assignment is larger than the indicated value. For the example below, any shapes with a dose between 0.5 and up to 1; between 1 and up to 2; and greater than 2 will receive 2, 3, and 7 passes, respectively.


2 X: -0.234232 um, Y: 0.025980 um,
FDA//noname:Poly (L:1(0), E:0.800)
3 X: 0.159497 um, Y: 0.038746 um,
FDA//noname:Rect (L:3(0), E:3.000)
1 X: -0.054064 um, Y: -0.040030 um,
FDA//noname:Poly (L:2(0), E:1.500)


Multipass Mode

Mode: Dose selective

Mainfield Offset X [-] 0 Y [-] 0

Subfield Offset X [-] 0 Y [-] 0

Layer for Multipass: *

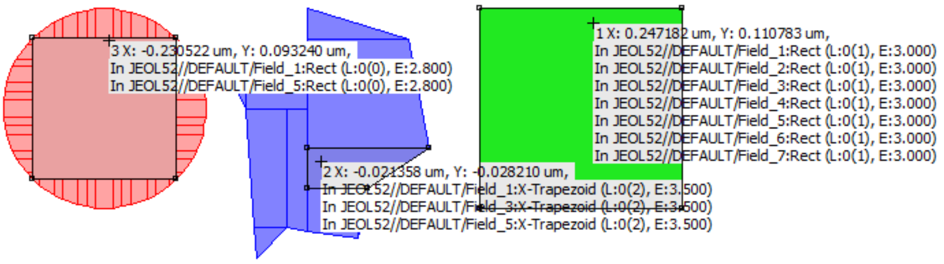
Dose larger	Number Passes
0.5	2
1	3
2	7

When using the Dose selective mode, the dose with the highest number of passes (*dose*$_{passesmax}$) is used to normalize the final applied relative doses in the output of the machine format. Here, the highest number of passes (*passes*$_{max}$) of 7 for a relative dose assignment of 3 (*dose*$_{passesmax}$) will be maintained. The other doses: 0.8 and 1.5 both of which will receive 2 and 3 passes, respectively, will have their dose assignments adjusted using the following formula:

$$new\_dose = original\_dose * passes_{max} / passes_{dose\ range}$$

As such, the new relative dose assignments are:

- 0.8 * 7 / 2 = 2.8
- 1.5 * 7 / 3 = 3.5



```
3 X: -0.230522 um, Y: 0.093240 um,
In JEOL52//DEFAULT/Field_1:Rect (L:0(0), E:2.800)
In JEOL52//DEFAULT/Field_5:Rect (L:0(0), E:2.800)

2 X: -0.021358 um, Y: -0.028210 um,
In JEOL52//DEFAULT/Field_1:X-Trapezoid (L:0(2), E:3.500)
In JEOL52//DEFAULT/Field_3:X-Trapezoid (L:0(2), E:3.500)
In JEOL52//DEFAULT/Field_5:X-Trapezoid (L:0(2), E:3.500)

1 X: 0.247182 um, Y: 0.110783 um,
In JEOL52//DEFAULT/Field_1:Rect (L:0(1), E:3.000)
In JEOL52//DEFAULT/Field_2:Rect (L:0(1), E:3.000)
In JEOL52//DEFAULT/Field_3:Rect (L:0(1), E:3.000)
In JEOL52//DEFAULT/Field_4:Rect (L:0(1), E:3.000)
In JEOL52//DEFAULT/Field_5:Rect (L:0(1), E:3.000)
In JEOL52//DEFAULT/Field_6:Rect (L:0(1), E:3.000)
In JEOL52//DEFAULT/Field_7:Rect (L:0(1), E:3.000)
```

This normalization method chosen here is used to maintain the reference for dose assignment in this differential multipass strategy.

Mainfield Offset

The default mainfield offset factor of 0.5 implies half of the actual mainfield size. Any value between 0 and 0.5 can be used to define the mainfield offset.

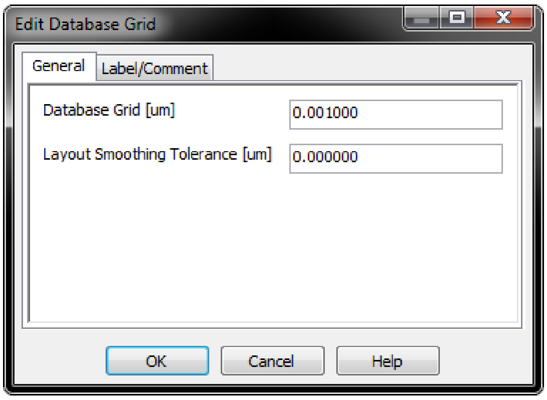
Subfield Offset

The default subfield offset factor of 0.5 implies half of the actual subfield size. Any value between 0 and 0.5 can be used to define the subfield offset.

Layer for Multipass

The layer a user can specify to impose a multipass exposure within a pattern when outputting to a machine format.

4.2.8 Grid



Database Grid [um] 0.001000

Layout Smoothing Tolerance [um] 0.000000

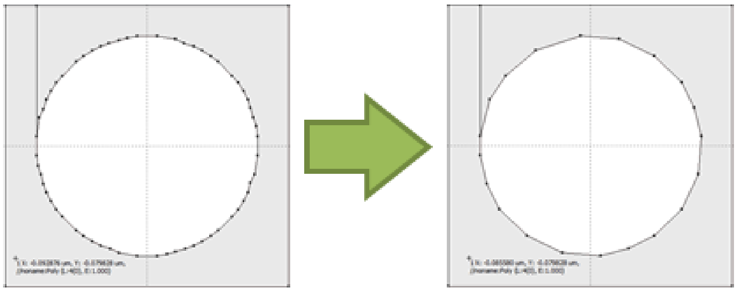
The Grid module modifies the database unit of a pattern and provides the capability to reduce the number of vertices for curved structures.

Database Grid [um]

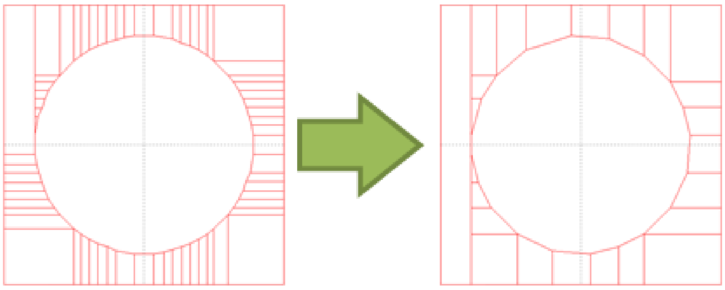
The database grid is the design unit to which vertices snap. The default value is 1nm. If a coarser value is used, snapping may occur in the vertices of the pattern do not sit on a grid that is an integer multiple of the coarser grid.

Layout Smoothing Tolerance

The smoothing parameter will eliminate a vertex with a distance smaller than the smoothing value to the next vertex. In other words, if a value of 0.005um is entered as the Layout Smoothing Tolerance, then any existing vertices that yield an approximated curve less than 0.005um will be removed, thus eliminating unnecessary vertices. By reducing the number of vertices, this will significantly reduce fracturing time. An example of layout smoothing tolerance is below. To the left, the image can be seen with several vertices that make up the hole in the square pattern. When applying layout smoothing, the number of vertices is reduced significantly. It is advised to always test and inspect different Layout Smoothing Tolerance values to be sure it meets your fracturing needs.

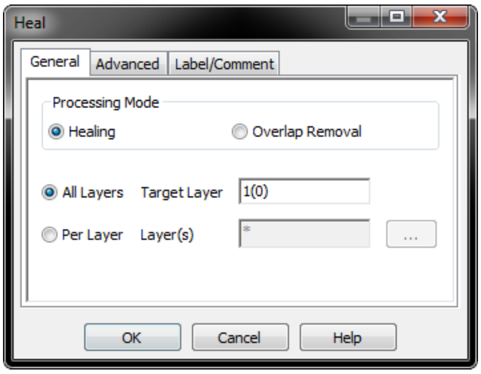


A direct result of the reduced number of vertices is the reduced number of shapes. In other words, the number of fractured shapes is reduced since the number of vertices is reduced. Reducing the number of shapes can reduce data volume and therefore any possible shape overhead delay for the tool.



4.2.9 Heal

The Heal module removes overlaps in the layout and joins abutting polygons into one. Overlaps can cause unwanted double exposure during writing resulting in poor shape fidelity. The default Heal settings will remove overlaps and merge the remaining elements into a single polygon onto a single layer. The default target layer is 1(0). If the Overlap Removal (OLR) radio button is selected only overlaps are removed with the pattern placed on layer 1(0). With OLR, polygon merging will not take place.


Heal

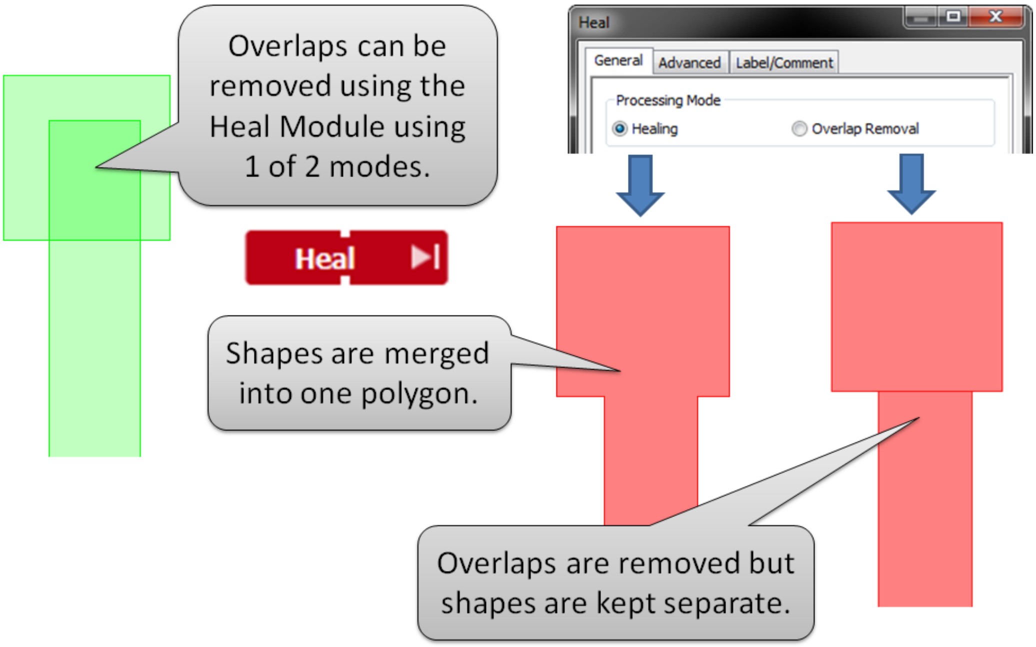
General | Advanced | Label/Comment

Processing Mode

● Healing ○ Overlap Removal

● All Layers Target Layer 1(0)

○ Per Layer Layer(s) *



Processing Mode: ● Healing ○ Overlap Removal

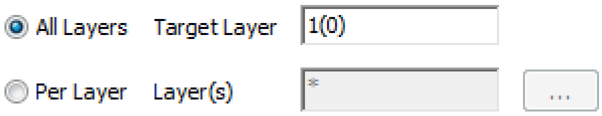
Overlaps can be removed using the Heal Module using 1 of 2 modes.

Shapes are merged into one polygon.

Overlaps are removed but shapes are kept separate.

4.2.9.1 Layerwise Operation

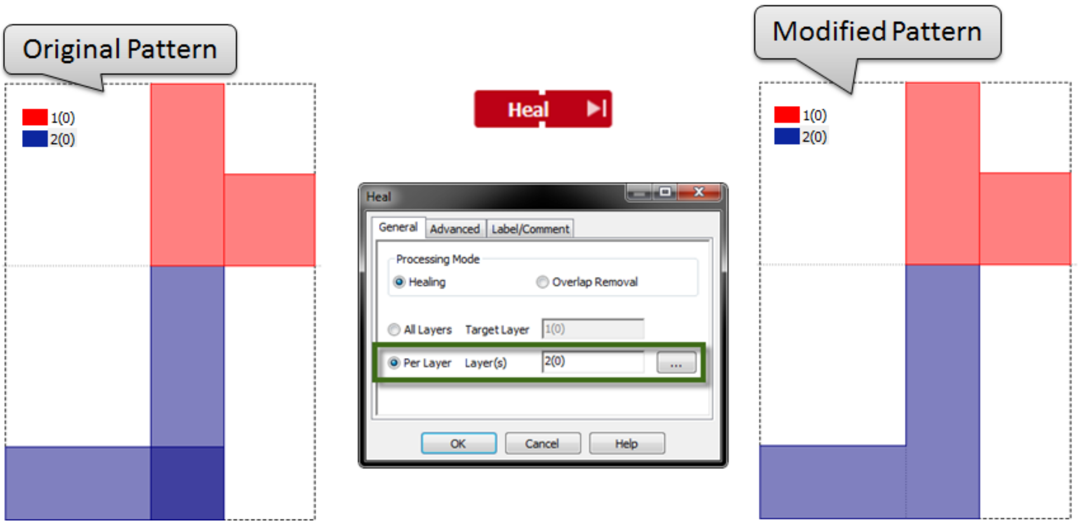
The Heal modules supports layerwise operations on a pattern. By default, all layers are healed together. As a result, layer information is lost and the resulting data is placed on a single target layer. In the case below, the target layer/datatype is set to 1(0).



● All Layers Target Layer 1(0)

○ Per Layer Layer(s) * ...

When Per Layer is selected, layers can be explicitly selected for modification. Here only layer 2(0) will be healed leaving the layer 1(0) in the pattern untouched. A wildcard character (*) can be used to denote all layers in this mode of operation.



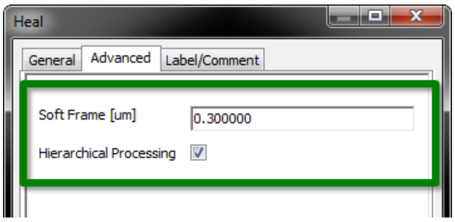
Processing Mode: ● Healing ○ Overlap Removal

○ All Layers Target Layer 1(0)

● Per Layer Layer(s) 2(0) ...

4.2.9.2 Softframe and Hierarchical Processing

In the **Advanced** tab, the Softframe and Hierarchical Processing parameters provide extended functionality in the Heal module. We will explain the softframe operation followed by the Hierarchical Processing operation.



Soft Frame [um] 0.300000
Hierarchical Processing ☑

This operation uses virtual tiles in memory to execute this operation quickly. The pattern as a whole is divided into virtual tiles, where each tile is kicked off to an available processing core for execution. Depending on the size of the pattern and available memory, few or many tiles are generated and the desired operation is executed on the tiles in parallel. Once all the tiles have been processed, they are reassembled into one pattern. The Softframe parameter is the overlap between these virtual tiles. In essence, any feature size that is equal to or smaller than the softframe (overlap) will not be cut in the virtual tile operation. Since feature sizes smaller than 0.300nm can be sensitive to being cut by this operation by producing non-optimal fractures, the default softframe value of 0.300um can be kept if you are unsure about what value to use.

Selecting **Hierarchical Processing** takes advantage of hierarchical layouts providing a shorter processing time. Depending on the operation and the design of the pattern some hierarchy may be lost.

4.2.10 Import

The Import module import layouts in standard formats without any size limitation.

- GDSII
- DXF
- OASIS (Standard & Mask)
- CIF
- LTXT (compatible with CTXT/TEXTLIB)
- LDB
- Generic Bitmaps (PNG)
- JEOL (JEOL51, JEOL52, J01)
- Vistec (VB6, FRE, GPF, IWFL)
- Raith EBL
- NuFlare
- MEBES
- Micronic
- NPI

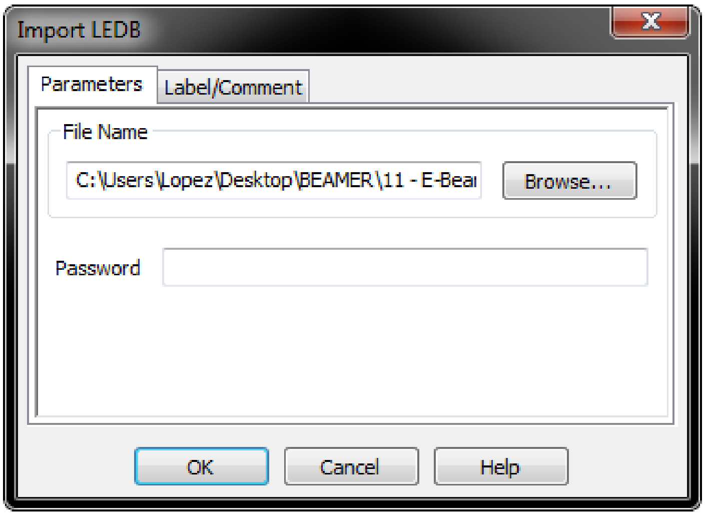
If you do not see a particular format you use, please inquire about the support of other file formats.
NOTE: To import e-beam machine formats, a license is required.

4.2.10.1 LEDB

LEDB is an proprietary, internal BEAMER database format that provides high performance access for loading and viewing.

Password

Provide a password to protect or access when exporting or importing sensitive files.



File Name: C:\Users\Lopez\Desktop\BEAMER\11 - E-Beai

Password:

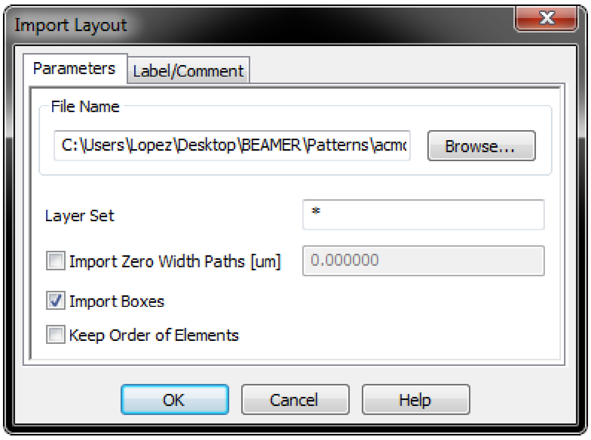
4.2.10.2 GDS

Layer Set

Defines the layers and datatypes to be imported or exported (depending if you are using the Import or Export module). This field can make use of the special annotations available for layer datatype handling.

Import Zero Width Paths

Paths that have zero-width can be imported by checking the optional box in the Import dialog menu.



File Name: C:\Users\Lopez\Desktop\BEAMER\Patterns\acm

Layer Set: *

☐ Import Zero Width Paths [um] 0.000000

☑ Import Boxes

☐ Keep Order of Elements

Import Boxes

The GDSII *box* elements will be imported as data with this feature enabled.

Keep Order of Elements

By default, this feature is disabled. When disabled, BEAMER will spatially-sort the shapes within the pattern in an optimal fashion for faster data handling/processing down stream; however, this is not always desired. When enabled, the original order of elements (shapes) are maintained. Some e-beam lithographers design their patterns with shape order in mind to minimize large deflections, avoiding placement errors in their tool. As such, the order in which elements are drawn in the CAD impacts the quality of the litho. This feature essentially allows these users to keep their intended shape writing order upon import.

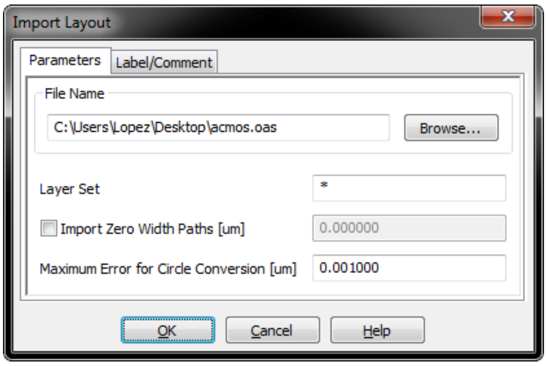
Caution should be taken when using this feature. The Export module with *No Compaction* enabled, FDA, some Transform module operations will not alter the shape order if used directly after the Import module. All other modules are not guaranteed to preserve the original writing order.

Transform operations that will not alter the shape order are:

1. Shift in X and/or Y
2. Mirror in X and/or Y
3. Rotation by an integer multiple of 90 degrees

4.2.10.3 OASIS

The OASIS format can be imported into BEAMER. Subsets of the OASIS format exists primarily due to the mask specific data sorting. This sorting is also available to choose as import option. The subset format, OASIS-MASK, has the layout data stored in a positional sort while the standard OASIS does not store data in an optimized way for mask writers.



File Name: C:\Users\Lopez\Desktop\acmos.oas

Layer Set: *

□ Import Zero Width Paths [um]: 0.000000

Maximum Error for Circle Conversion [um]: 0.001000

Layout Set

Defines the layers and datatypes to be imported.

Import Zero Width Paths

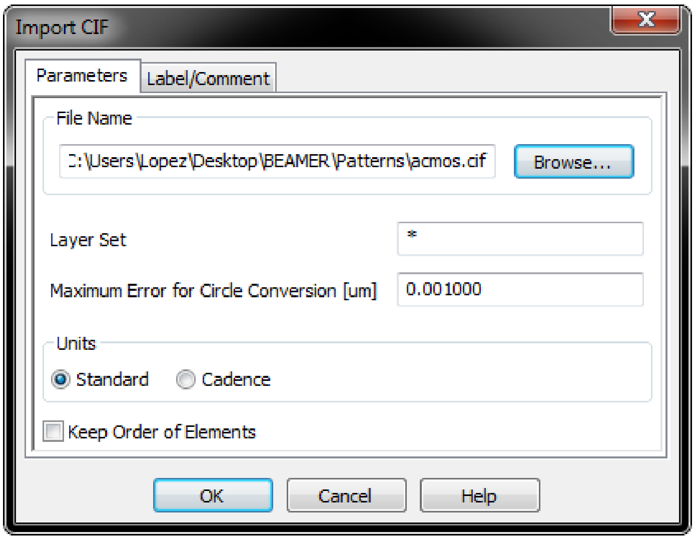
To assign a width to zero width paths during the import enable this feature.

Maximum Error for Circle Conversion [um]

This parameter is the maximal allowable amount of deviation from the original circle object to the approximated polygon. Before importing any circles, they are digitized with some degree of accuracy. This field is set to 1nm by default which implies that the digitized approximated design will not deviate more than 1nm away from the original parametric circle upon import.

4.2.10.4 CIF

CIF is the CalTech Intermediate Format file format. It is a human readable text based that can be opened with a text editor (e.g. Wordpad).



File Name: :\Users\Lopez\Desktop\BEAMER\Patterns\acmos.cif

Layer Set: *

Maximum Error for Circle Conversion [um]: 0.001000

Units: ● Standard ○ Cadence

□ Keep Order of Elements

Layer Set

Defines the layers and datatypes to be imported or exported (depending if you are using the Import or Export module). This field can make use of the special annotations available for layer datatype handling.

Maximum Error for Circle Conversion [um]

This parameter is the maximal allowable amount of deviation from the original circle object to the approximated polygon. Before importing any circles, they are digitized with some degree of accuracy. This field is set to 1nm by default which implies that the digitized approximated design will not deviate more than 1nm away from the original parametric circle upon import.

Units

CIF files contain only unitless integer numbers. Therefore, it is important to get the correct physical dimension when importing the file. A file is imported using **Standard** units by default. Any patterns exported as CIF in BEAMER will use the **Standard** units. Alternatively, **Cadence** units can be selected. With this option selected, the units is adjusted to a tenth of the **Standard** units. Depending on the how the CIF file was designed, be sure to inspect the imported pattern to make sure the features are at the desired size. If the features are 10x too large, open the Import module again and change the Units to **Cadence**.

Keep Order of Elements

By default, this feature is disabled. When disabled, BEAMER will spatially-sort the shapes within the pattern in an optimal fashion for faster data handling/processing down stream;

however, this is not always desired. When enabled, the original order of elements (shapes) are maintained. Some e-beam lithographers design their patterns with shape order in mind to minimize large deflections, avoiding placement errors in their tool. As such, the order in which elements are drawn in the CAD impacts the quality of the litho. This feature essentially allows these users to keep their intended shape writing order upon import.

Caution should be taken when using this feature. The Export module with *No Compaction* enabled, FDA, some Transform module operations will not alter the shape order if used directly after the Import module. All other modules are not guaranteed to preserve the original writing order.

Transform operations that will not alter the shape order are:

1. Shift in X and/or Y
2. Mirror in X and/or Y
3. Rotation by an integer multiple of 90 degrees

4.2.10.5 DXF

The DXF import loads a DXF file into BEAMER.

Layer Set

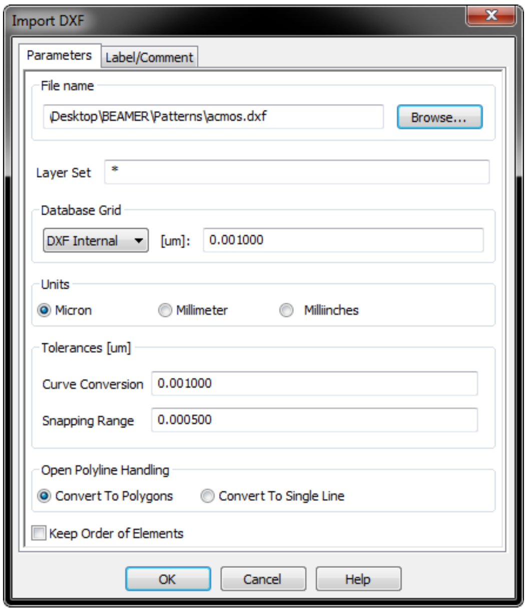
Defines the layers and datatypes to be imported or exported (depending if you are using the Import or Export module). This field can make use of the special annotations available for layer datatype handling.

Database Grid

DXF Internal uses resolution variables defined in the DXF header. *Fixed* applies the resolution grid specified in the input box.

Units

The physical units to be used when importing the DXF file.



File name: \Desktop\BEAMER\Patterns\acmos.dxf

Layer Set: *

Database Grid: DXF Internal [um]: 0.001000

Units: ● Micron ○ Millimeter ○ Milliinches

Tolerances [um]

Curve Conversion: 0.001000

Snapping Range: 0.000500

Open Polyline Handling: ● Convert To Polygons ○ Convert To Single Line

□ Keep Order of Elements

Tolerances

When importing certain DXF patterns into BEAMER some structures like Bezier curves must be approximated and digitized using vertices. Additionally polygons that are generated with single lines must be imported with some degree of accuracy to import the intended design. Two parameters assist with importing patterns in DXF format.

Curve Conversion is the maximal allowable amount of deviation from the original curved structure to the approximated polygon that will replace it upon import. Before importing parametric curves (e.g. Bezier curves), they are digitized with some degree of accuracy. This field is set to 1nm by default which implies that the digitized approximated design will not deviate more than 1nm away from the original parametric curve upon import.

Snapping Tolerance is used to for polygon recognition for polygons generated using single lines. In DXF polygons can be drawn using abutting single lines. Because of snapping or rounding errors, line ends may not touch as expected, creating an open in the shape. By allowing a snapping tolerance, the recognition of polygons drawn with single lines is increased. In other words, the snapping snapping tolerance is the maximal distance allowed between two abutting lines that make up a polygon for the polygon to be considered closed and imported properly.

Open Polyline Handling

Polylines that are not closed in the DXF can either be converted to polygons or to single lines when *Convert to Polygons* or *Convert to Single Line* is selected, respectively.

Supported objects in the DXF file import are:

- LINE
- CIRCLE
- ARC
- ELLIPSE
- TRACE
- SOLID
- SPLINE
- BLOCK
- ENDBLK
- INSERT
- POLYLINE
- LWPOLYLINE

Keep Order of Elements

By default, this feature is disabled. When disabled, BEAMER will spatially-sort the shapes within the pattern in an optimal fashion for faster data handling/processing down stream; however, this is not always desired. When enabled, the original order of elements (shapes) are maintained. Some e-beam lithographers design their patterns with shape order in mind to minimize large deflections, avoiding placement errors in their tool. As such, the order in which elements are drawn in the CAD impacts the quality of the litho. This feature essentially allows these users to keep their intended shape writing order upon import.

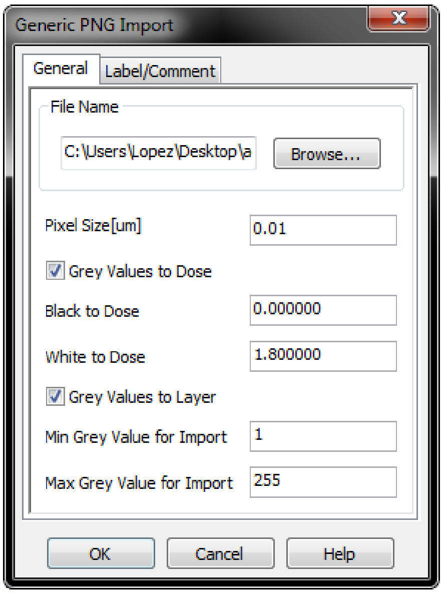
Caution should be taken when using this feature. The Export module with *No Compaction* enabled, FDA, some Transform module operations will not alter the shape order if used directly after the Import module. All other modules are not guaranteed to preserve the original writing order.

Transform operations that will not alter the shape order are:

1. Shift in X and/or Y
2. Mirror in X and/or Y
3. Rotation by an integer multiple of 90 degrees

4.2.10.6 Generic Bitmap (PNG)

The Import module has the ability to import grayscale PNG files. Grayscale images can vary in the number of shades of gray that are available in their color palette. This is dictated by the number of bits in the image. An 8-bit grayscale will have 256 (2^8) shades of gray. The color black has a value of 0, and the color white has a value of 255.


Generic PNG Import

File Name: C:\Users\Lopez\Desktop\a

Pixel Size[um]: 0.01

☑ Grey Values to Dose

Black to Dose: 0.000000

White to Dose: 1.800000

☑ Grey Values to Layer

Min Grey Value for Import: 1

Max Grey Value for Import: 255

Pixel Size [nm]

Defines the size to convert an individual pixel.

Grey Values to Dose

Enables the conversion of gray values in the image to dose values upon import.

Black to Dose

When **Grey Values to Dose** is enabled the color black in the image can be assigned a specific relative dose value. All doses in between the black dose and white dose are interpolated. The number of doses generate depends on the number of bits of the image.

White to Dose

When **Grey Values to Dose** is enabled the color white can be assigned a specific relative dose value. All doses in between the black dose and white dose are interpolated. The number of doses generate depends on the number of bits of the image.

Grey Values to Layer

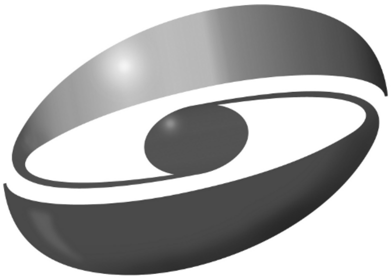
Enables the assignment of gray values to specific layers upon import.

Min/Max Grey Value for Import

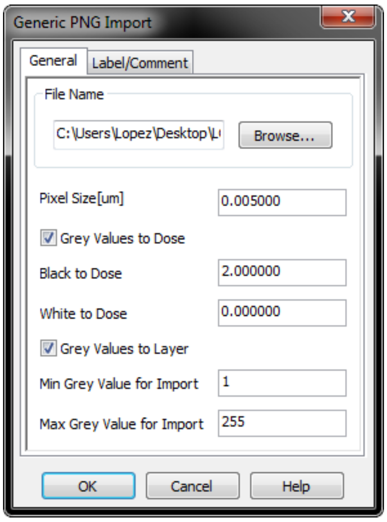
The range of gray values to import can be set using the fields *Min Grey Value for Import* and *Max Grey Value for Import*. Any grayscale colors whose value fall in between this range will be imported. If the **Grey Values to Layer** option is enabled, each pixel of a specific gray value will be assigned a unique layer.

An example of importing a grayscale PNG is show below. Here we are going to import a grayscaled version of the GenISys logo to the right. We will use various settings to demonstrate the various Import module settings.

The image will be imported using a 5nm pixel size.



First, let's import the image using the default settings. This implies that completely black pixels will be imported with a relative dose of 2 and completely white pixels will receive a relative dose of 0, which means no dose. All doses in between are interpolated based on the number of shades of gray available for import. The Import module will also import all gray values (between 1-255) as layers upon import. Effectively, this image could be used for grayscale lithography where doses correspond to a particular resist height after development. The result of the import is shown below to the right. The lowest dose is in blue and the highest is in red.



Generic PNG Import

General | Label/Comment

File Name: C:\Users\Lopez\Desktop\L

Pixel Size[um]: 0.005000

☑ Grey Values to Dose

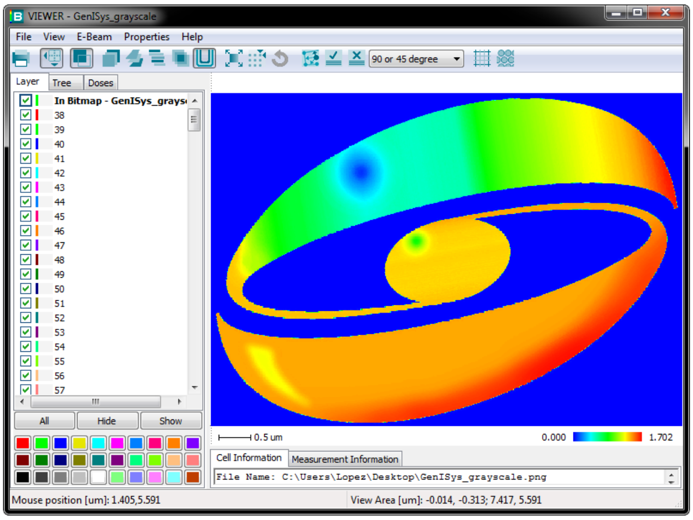
Black to Dose: 2.000000

White to Dose: 0.000000

☑ Grey Values to Layer

Min Grey Value for Import: 1

Max Grey Value for Import: 255



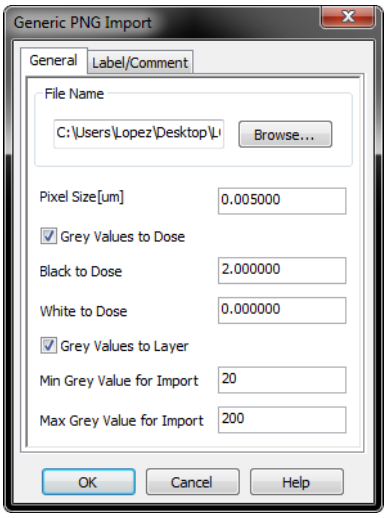
VIEWER - GenISys_grayscale

File Name: C:\Users\Lopez\Desktop\GenISys_grayscale.png

Mouse position [um]: 1.405,5.591

View Area [um]: -0.014, -0.313; 7.417, 5.591

In the next example, the same color to dose assignments are maintained, however, the gray value range is limited to 20 to 200. This means that the module will only import grayscale values that fall in this range and assign them a layer. You'll notice that the white pixels are excluded. Again, the lowest dose is in blue and the highest is in red.



Generic PNG Import

General | Label/Comment

File Name: C:\Users\Lopez\Desktop\L

Pixel Size[um]: 0.005000

☑ Grey Values to Dose

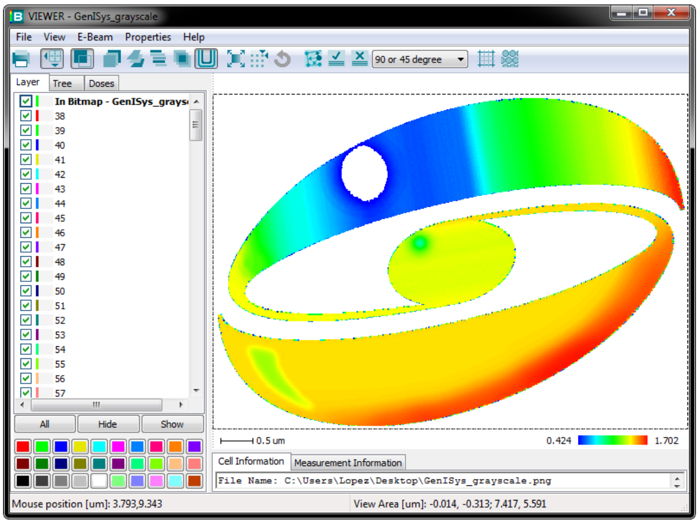
Black to Dose: 2.000000

White to Dose: 0.000000

☑ Grey Values to Layer

Min Grey Value for Import: 20

Max Grey Value for Import: 200



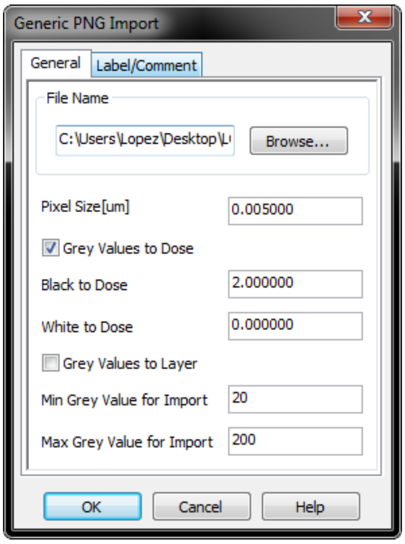
VIEWER - GenISys_grayscale

File Name: C:\Users\Lopez\Desktop\GenISys_grayscale.png

Mouse position [um]: 3.793,9.343

View Area [um]: -0.014, -0.313; 7.417, 5.591

The next example is the same as the previous, except that the option Gray Values to Layer is disabled. The same color to dose assignments are maintained with the gray value range limited to 20 to 200. However, the module will not assign the selected colors a unique layer. Again, the lowest dose is in blue and the highest is in red.



Generic PNG Import

File Name: C:\Users\Lopez\Desktop\L

Pixel Size[um]: 0.005000

☑ Grey Values to Dose

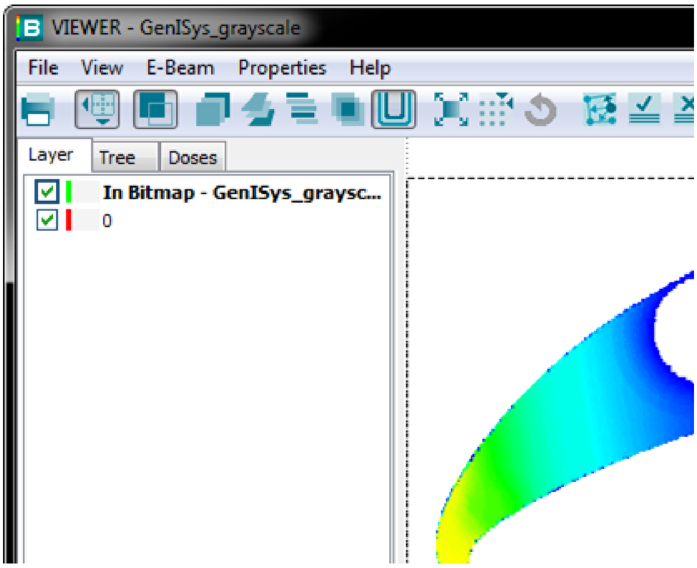
Black to Dose: 2.000000

White to Dose: 0.000000

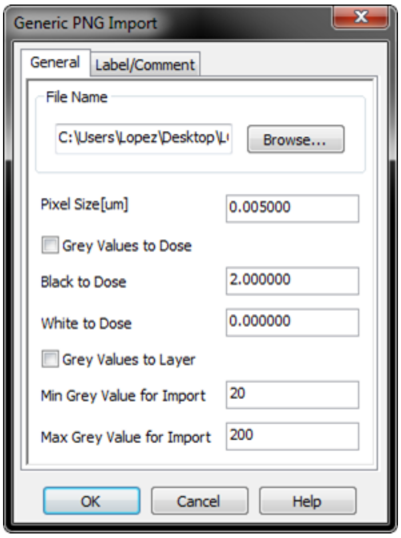
☐ Grey Values to Layer

Min Grey Value for Import: 20

Max Grey Value for Import: 200



In the fourth example below, the options **Gray Values to Dose** and **Gray Values to Layer** are disabled. No doses will be assigned to any pixels. Also, like the last example, the gray value range to import is 20 to 200, but the module will not assign the selected colors a unique layer.



Generic PNG Import

File Name: C:\Users\Lopez\Desktop\L

Pixel Size[um]: 0.005000

☐ Grey Values to Dose

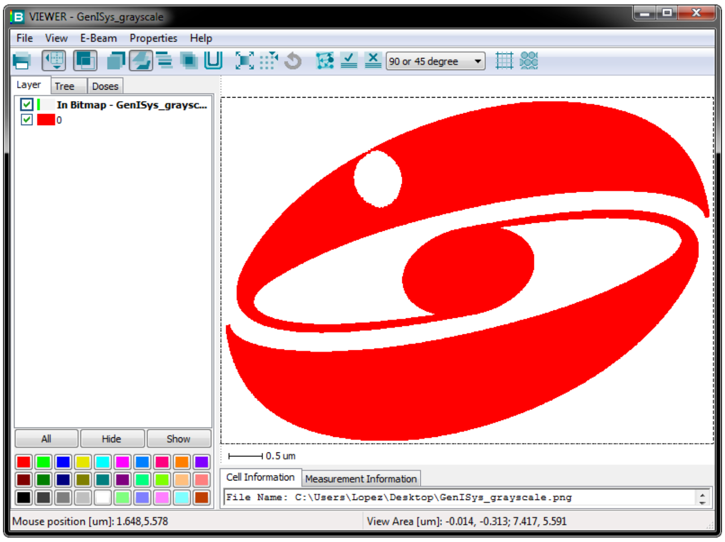
Black to Dose: 2.000000

White to Dose: 0.000000

☐ Grey Values to Layer

Min Grey Value for Import: 20

Max Grey Value for Import: 200

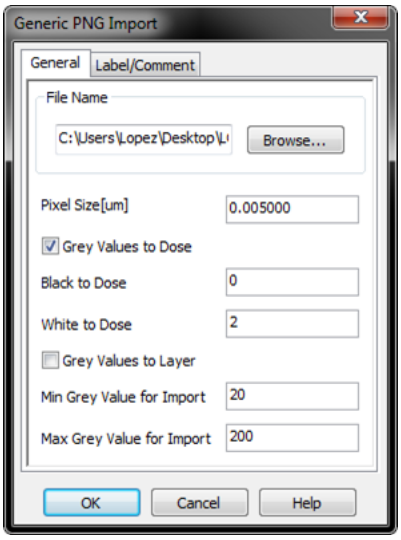


File Name: C:\Users\Lopez\Desktop\GenISys_grayscale.png

Mouse position [um]: 1.648,5.578

View Area [um]: -0.014, -0.313; 7.417, 5.591

In the last example, **Gray Values to Dose** is enabled with **Gray Values to Layer** disabled. Again, the gray value range to import is 20 to 200, but the module will not assign the selected colors a unique layer. This time the doses will be assigned with black pixels receiving a dose of 0 and while pixels receiving a dose of 2. This completely reverses the dose assignments used in the first 3 examples should your application require it.



Generic PNG Import

File Name: C:\Users\Lopez\Desktop\L

Pixel Size[um]: 0.005000

☑ Grey Values to Dose

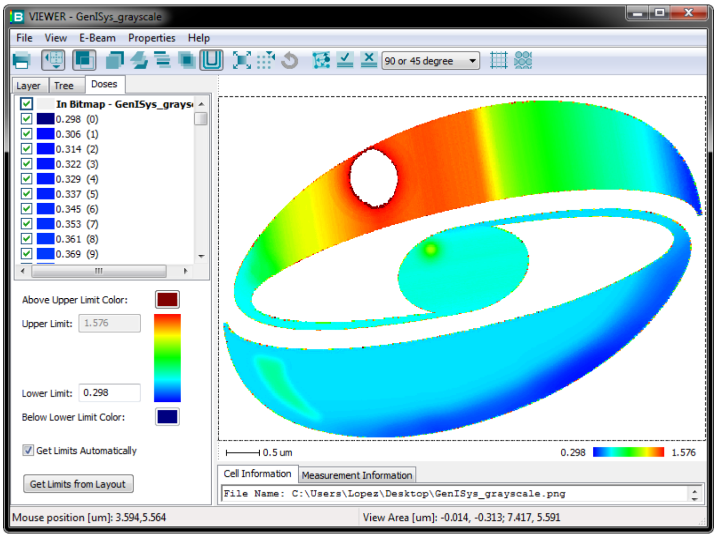
Black to Dose: 0

White to Dose: 2

☐ Grey Values to Layer

Min Grey Value for Import: 20

Max Grey Value for Import: 200



VIEWER - GenISys_grayscale

☑ In Bitmap - GenISys_grays

☑ 0.298 (0)

☑ 0.306 (1)

☑ 0.314 (2)

☑ 0.322 (3)

☑ 0.329 (4)

☑ 0.337 (5)

☑ 0.345 (6)

☑ 0.353 (7)

☑ 0.361 (8)

☑ 0.369 (9)

Above Upper Limit Color:

Upper Limit: 1.576

Lower Limit: 0.298

Below Lower Limit Color:

☑ Get Limits Automatically

File Name: C:\Users\Lopez\Desktop\GenISys_grayscale.png

Mouse position [um]: 3.594,5.564

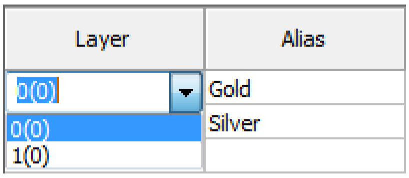
View Area [um]: -0.014, -0.313; 7.417, 5.591

4.2.11 Mapping

The Mapping module renames a layer to a new name by simply modifying the layer name map. In the example to the right and below, the layers 0(0) and 1(0) will be renamed to a new name (alias), *Gold* and *Silver*, respectively.

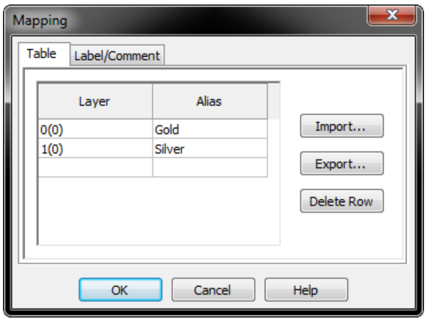
If the the previous module is ran, the Layers should be available using a dropdown by clicking on the Layer field in the row. as seen to the right.

In GDSII
Mapping



Layer	Alias
0(0)	Gold

0(0)
1(0)
	Silver



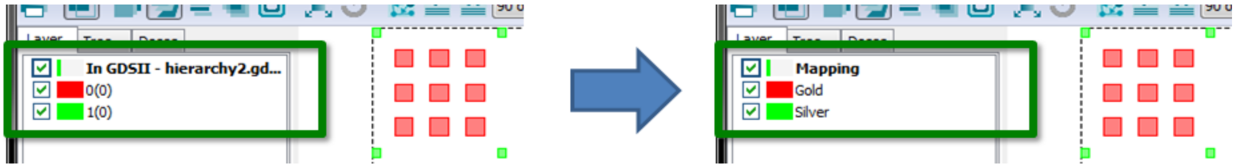
Mapping

Layer	Alias
0(0)	Gold
1(0)	Silver

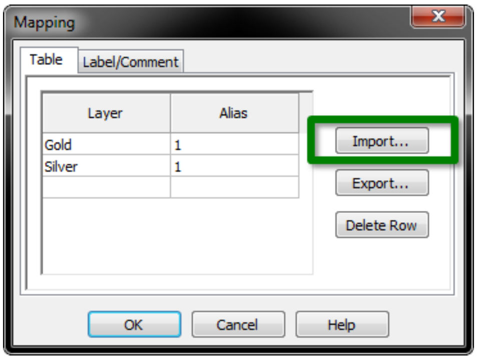
To the right, a complete remap of layers 0(0) and 1(0) are defined in the Mapping dialog. When the user click OK and runs the module, the layers will be renamed. By using the mapping module this provides a facility to make layers more readable during dataprep.

Below is a comparison before (left) and after (right) mapping of the layers.

It is clear that the layers 0(0) and 1(0) have been changed to *Gold* and *Silver*, respectively, using the Layout VIEWER.



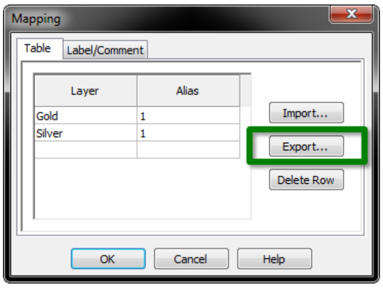
If there are many layers to rename to a new alias, tables can be imported via the **Import...** button in the Mapping dialog. A table can be generated in a text editor; an example of the Mapping format is below. Any lines beginning with a # symbol are comments. In the example, lines 1 and 2 are comments. The last two lines show the remap of the original layer name to a new name (e.g. 0(0) to Gold on line 3). Use this import feature if you have a common mapping you would like to use without having to always re-type the list line-by-line.



Layer	Alias
Gold	1
Silver	1

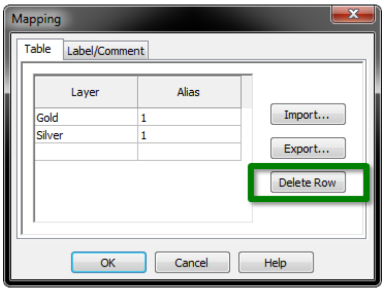
```
# LAYER MAPPING TABLE
# Layer Name,  Layer Alias
0(0),     Gold
1(0),     Silver
```

Exporting a mapping table is facilitated with the **Export...** button. This is exceedingly useful if the remapping has been composed in the Mapping module and you would like to import the table into another Mapping module at a late time. To export the table contents from the Mapping dialog, use the **Export...** button to save the table to a text file.



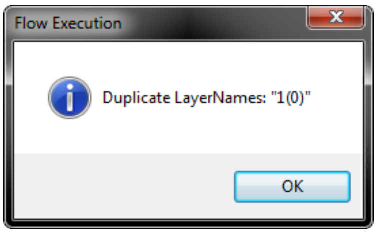
Layer	Alias
Gold	1
Silver	1

Finally, should you want any layer to alias pair to be removed from the table, click on the row you want to remove and click the **Delete Row** button on the right of the table in the dialog.



Layer	Alias
Gold	1
Silver	1

Since the Mapping module is only modifying the layer map in the layout, assigning a layer to a new layer name that is the same as another existing layer is not possible. For example, if a pattern contains layers 0(0) and 1(0), and the desire is to push 0(0) into layer 1(0) using the Mapping module, this is not possible. Attempting such an operation will yield a similar warning to the right.



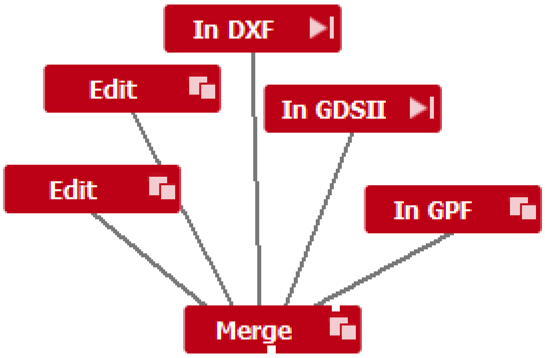
Duplicate LayerNames: "1(0)"

If the desire is to push all layers onto a single layer without losing hierarchy, attach a Size module instead of a Mapping module and apply a zero bias but change the target layer to the desired layer.

4.2.12 Merge

The Merge module accepts the output of multiple modules and merges them into a single layout. Merge preserves layout attributes like layer and datatype while maintaining the original shapes in the pattern.

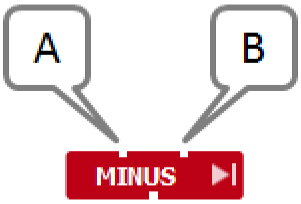
To the left, the Merge module is merging several imported patterns of different format types into one pattern. The Merge module can merge any kind of output from any module while maintaining each patterns information.

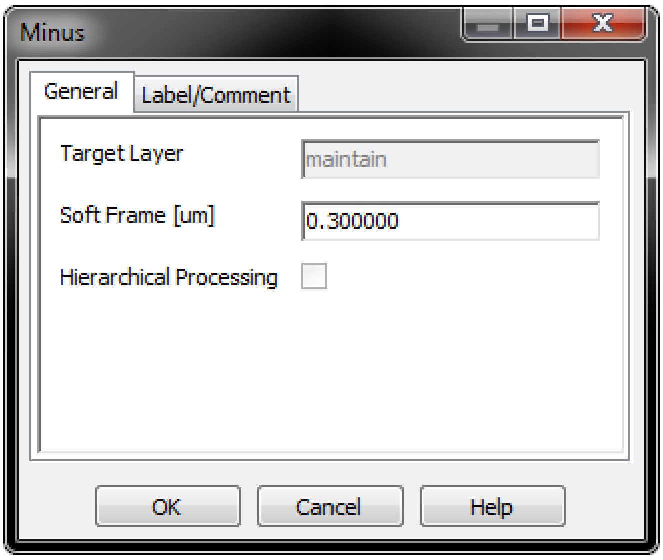


Doing a merge of corrected layers and uncorrected layouts the dose information will be maintained. Uncorrected layouts always have a relative dose of 1.0. When merged with a PEC-ed layout, the relative dose of the uncorrected layout should still be 1.0.

4.2.13 MINUS

The MINUS module takes two layouts and returns the data when Layout B data is subtracted from Layout A data. The MINUS module always subtracts the right input from the left. In the example below, Layout A is in green and Layout B is in blue. The Minus module preserves layer information and overlaps. Dose information is *not* maintained in the operation.



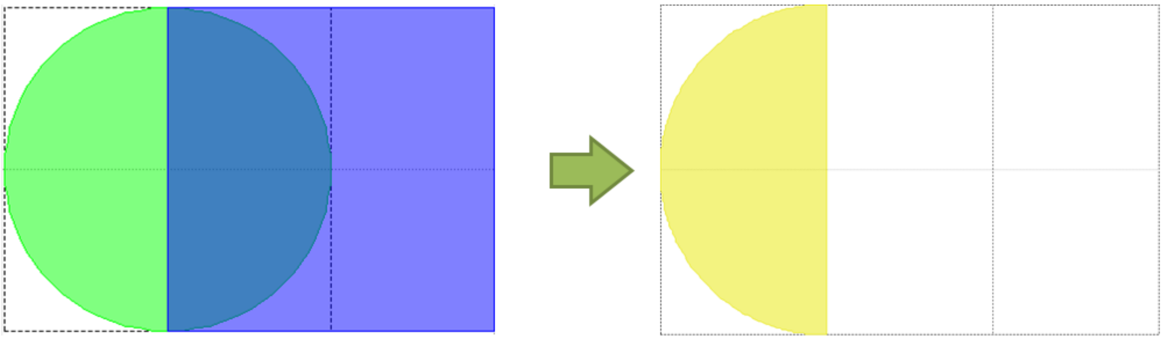


Target Layer: maintain

Soft Frame [um]: 0.300000

Hierarchical Processing: □

The target layer and data type for the result is set to the layer and datatype of Layout A. To avoid unnecessary fracturing of the original layout a soft frame can be defined.



Layout Logic:

MINUS		
A	**B**	**Result**
0	0	0
0	1	0
1	0	1
1	1	1

0 indicates a layout element is absent, 1 indicates a layout element is present

4.2.13.1 Preserving Dose Assignments

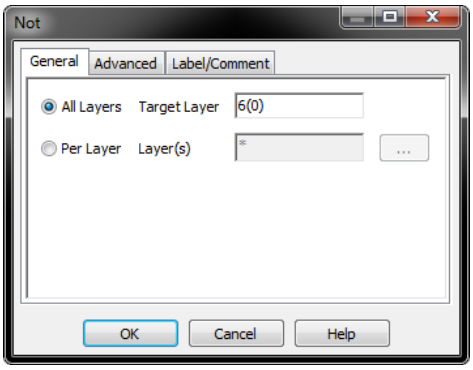
Should you want to preserve the dose information after the Minus, follow the procedure below:

1. Export the layout as a GDSII with the option "Dose to layer" enabled.
2. Run the Minus operation, which will preserve the layer information.
3. Export the result layout with the same file name as the original file to overwrite the prior exported GDSII file.
4. Re-import the exported layout and the dose will be derived from the formerly created dose table and combined with the layer information.

4.2.14 NOT

The NOT module inverts the tone of the data. In the conversion, data becomes non-data (space) and space (non-data) becomes data. Be forewarned that this operation can lead to polygons with a very large number of vertices.

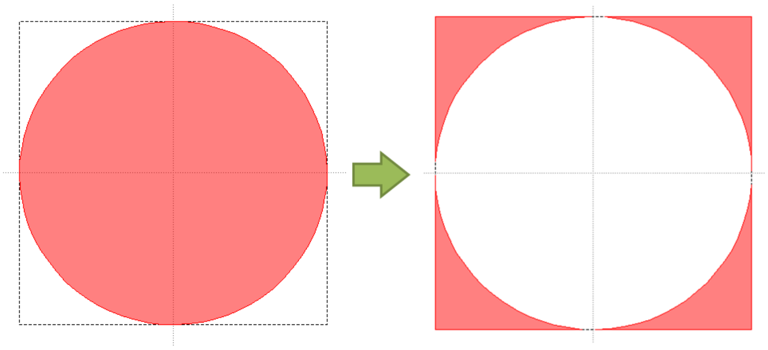
The target layer and datatype in the result can be specified by modifying the target layer value in the NOT dialog. The dialog is accessible by double-clicking the module. By default, the target layer and datatype is 6(0). Below is an example of a reverse tone operation of a circle.


Not

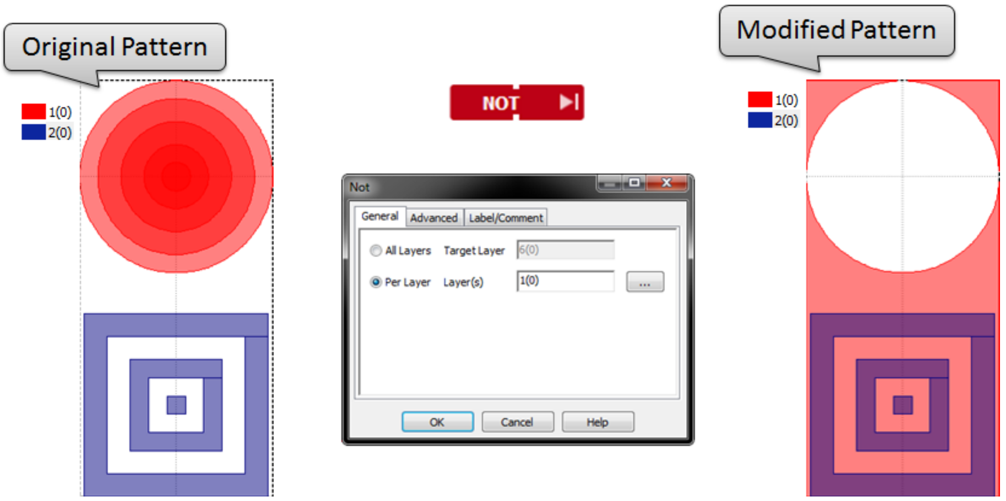
General | Advanced | Label/Comment

● All Layers Target Layer 6(0)

○ Per Layer Layer(s) *



Layerwise operation of the NOT module is also supported. In the example below, there are two layers in the pattern. By selecting Per Layer and indicating layer 1(0), here only layer 1(0) is reverse toned while layer 2(0) is untouched. The target layer is maintained as the original layer.

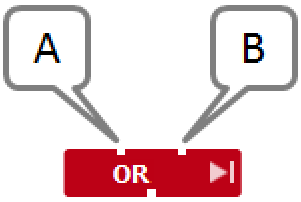


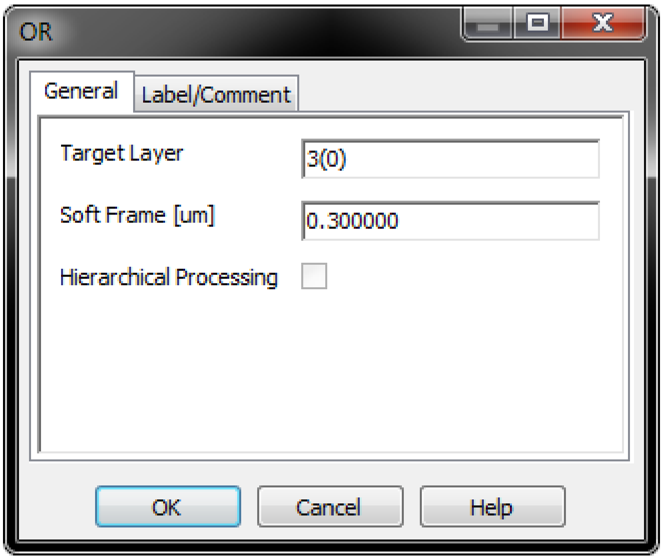
○ All Layers Target Layer 6(0)

● Per Layer Layer(s) 1(0)

4.2.15 OR

The OR module takes two layouts as input and returns data where Layout A data or Layout B data is present.





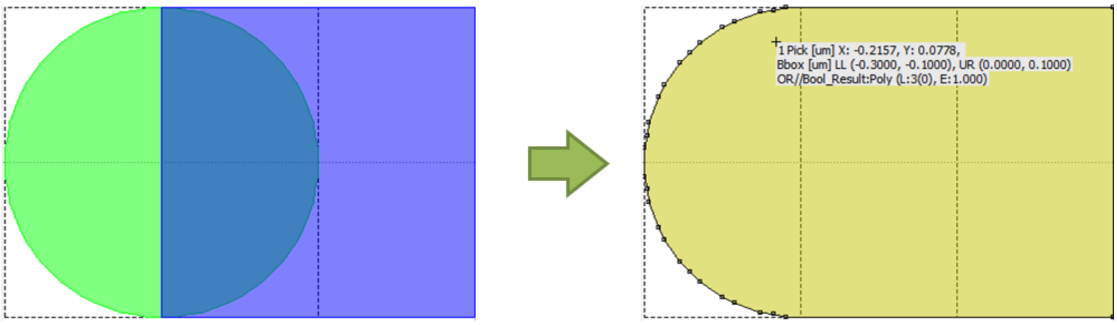
Target Layer: 3(0)

Soft Frame [um]: 0.300000

Hierarchical Processing □

The target layer and datatype for the result can be specified by opening the parameter window with a double click. By default, the target layer and datatype is 3(0). To avoid unnecessary fracturing of the original layout a soft frame can be defined in the parameter window.

The OR operation retains all the shapes in a pattern A and B and heals them together in the output:



```
1 Pick [um] X: -0.2157, Y: 0.0778,
Bbox [um] LL (-0.3000, -0.1000), UR (0.0000, 0.1000)
OR//Bool_Result:Poly (L:3(0), E:1.000)
```

Layout Logic:

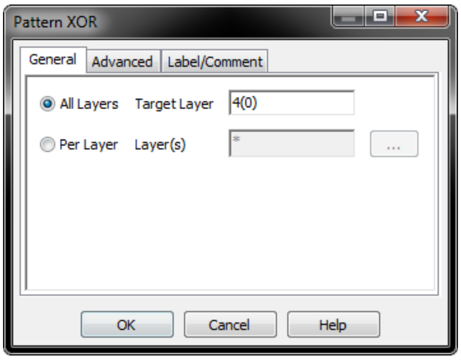
OR		
A	B	Result
0	0	0
0	1	1
1	0	1
1	1	1

0 indicates a layout element is absent, 1 indicates a layout element is present.

4.2.16 P-XOR

The P-XOR (Pattern-XOR) is a special Boolean operation that removes even overlaps and keeps odd overlaps.

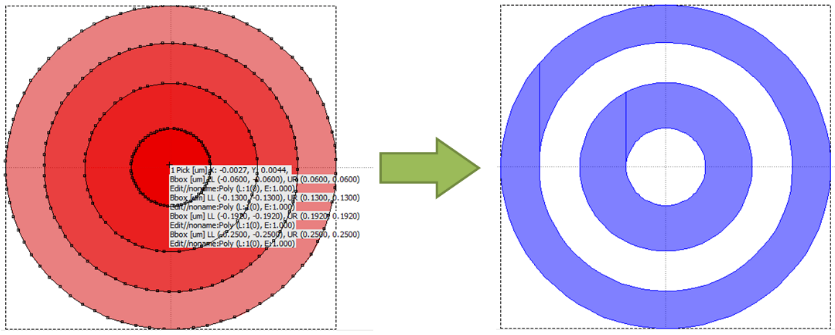
The target layer and datatype in the result can be specified by opening the parameter window by double clicking the module. By default, the target layer and datatype is 4(0). To avoid unnecessary fracturing of the original layout a soft frame can be defined in the parameter window.


Pattern XOR

● All Layers Target Layer: 4(0)

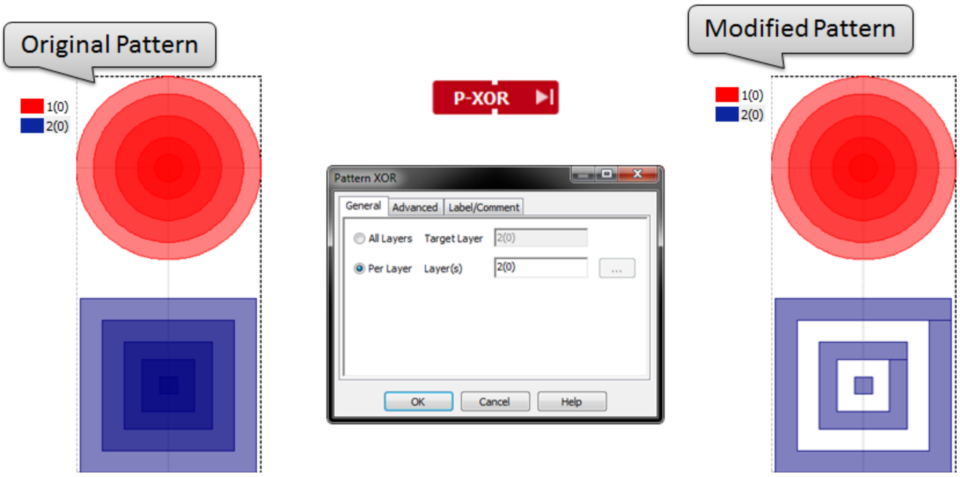
○ Per Layer Layer(s): *

In the example below, a pattern of concentric rings (blue) can be quickly generated from a pattern with concentrically overlapped circles (red).


```
1 Pick [um] X: -0.0027, Y: 0.0044,
Bbox [um] LL (-0.0600, -0.0600), UR (0.0600, 0.0600)
Edit//noname:Poly (L:1(0), E:1.000)
Bbox [um] LL (-0.1300, -0.1300), UR (0.1300, 0.1300)
Edit//noname:Poly (L:1(0), E:1.000)
Bbox [um] LL (-0.1920, -0.1920), UR (0.1920, 0.1920)
Edit//noname:Poly (L:1(0), E:1.000)
Bbox [um] LL (-0.2500, -0.2500), UR (0.2500, 0.2500)
Edit//noname:Poly (L:1(0), E:1.000)
```

Areas with an odd (1,3,5,7…) number of polygons overlaying will remain. Areas with an even number of polygons (2,4,6,8…) will be removed. This special function is very useful to easily generate circular gratings like zone plates.

Layerwise operation is also supported in this module. Below in the example, the original layout contains two layers, 1(0) and 2(0). With Per Layer selected, layer 2(0) is entered as the layer to apply a P-XOR. In the resulting modified pattern, the original layers are maintained and only the pattern in layer 2(0) has the P-XOR applied.



Pattern XOR

General | Advanced | Label/Comment

○ All Layers Target Layer 2(0)

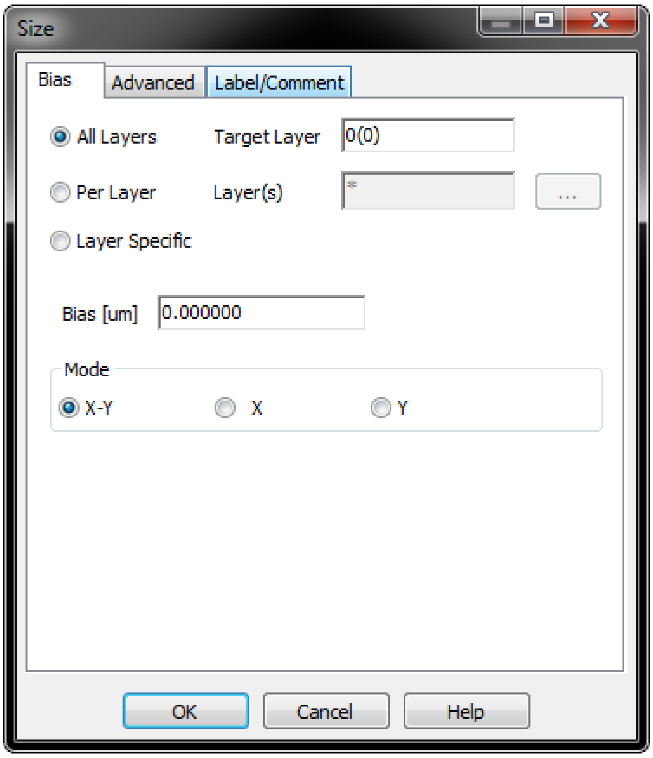
● Per Layer Layer(s) 2(0) ...

OK Cancel Help

4.2.17 Size

The Size module applies either a positive or negative geometrical bias to the polygons in the layout without healing objects together. If a positive bias is applied to two or more abutting shapes, they will be kept separate. Should you desire a bias operating with an implied heal, use the Bias module instead.

The Bias parameter adds on or subtracts the entered dimension to all sides of all polygons. In the example below, the original pattern is in dark blue with a width of a 1µm. The pattern is negatively biased which is shown in red. All the individual shapes in the pattern are reduced by 0.2µm on every side which effectively separates out the shapes.



Size

Bias | Advanced | Label/Comment

● All Layers Target Layer 0(0)

○ Per Layer Layer(s) * ...

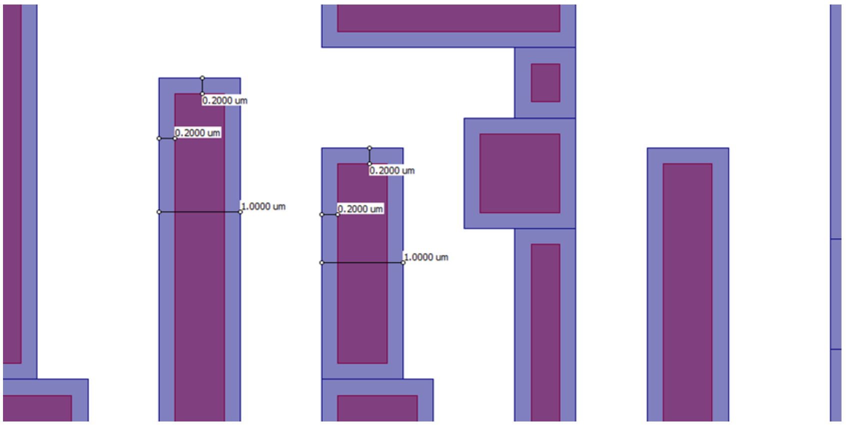
○ Layer Specific

Bias [um] 0.000000

Mode

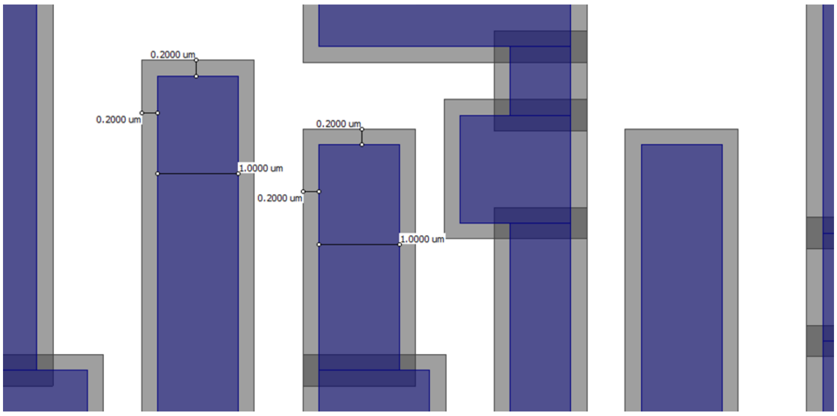
● X-Y ○ X ○ Y

OK Cancel Help



In the next example below, the original pattern is in dark blue with a width of 1µm. The pattern is positively biased which is shown in gray. All the individual shapes in the pattern are grown by 0.2µm.

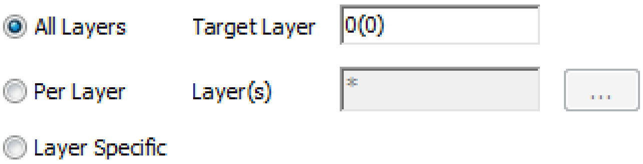
All abutting shapes are now overlapping which is seen in gray.



In both examples, the Grow operation was applied using the X-Y mode. Should an X or Y only bias is required on a pattern, the X or Y mode can be selected respectively.

4.2.17.1 Layerwise Operation

The Bias and Grow modules support layerwise operation on a pattern. By default, all layers are biased together. As a result, layer information is lost and the resulting data is placed on a single target layer. In the case below, the target layer/datatype is set to 0(0).

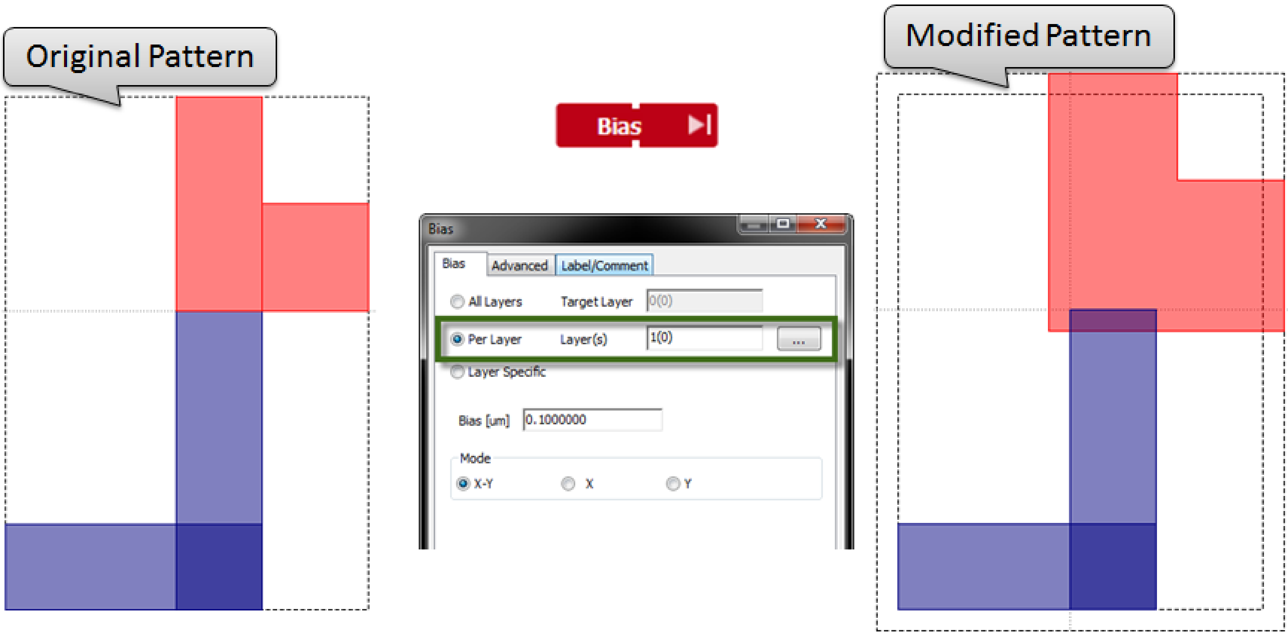


● All Layers Target Layer 0(0)

○ Per Layer Layer(s) * ...

○ Layer Specific

When Per Layer is selected, layers can be explicitly selected for modification. Here only layer 1(0) will be modified uniformly leaving layer 2(0) untouched. A wildcard character (*) can be used to denote all layers in this mode of operation.



Bias

Bias | Advanced | Label/Comment

○ All Layers Target Layer 0(0)

● Per Layer Layer(s) 1(0) ...

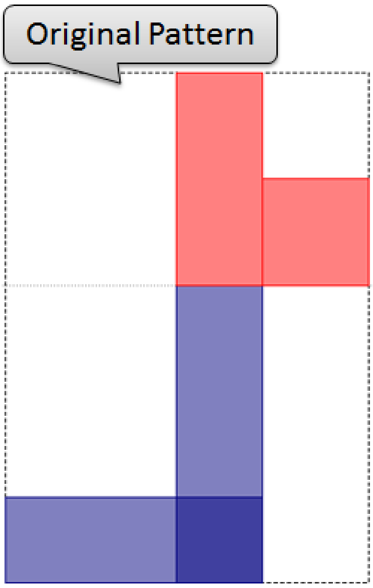
○ Layer Specific

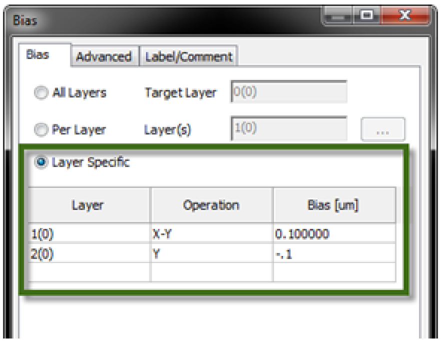
Bias [um] 0.1000000

Mode

● X-Y ○ X ○ Y

Lastly, Layer Specific operations allow the user to select layers available in the pattern and apply a specific bias operation (X-Y, X, Y) to each specified layer. Moreover, each layer can have a different bias value applied.




Bias

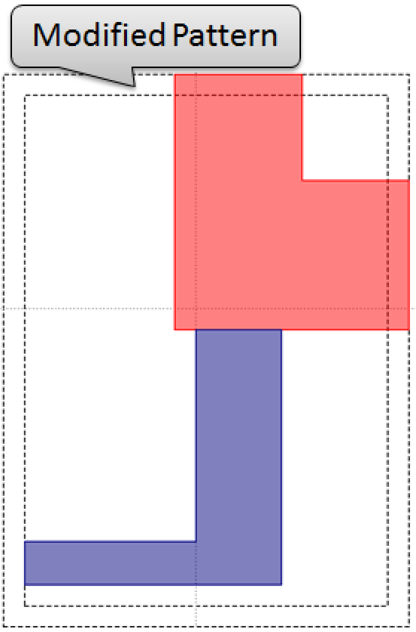
Bias | Advanced | Label/Comment

○ All Layers Target Layer 0(0)

○ Per Layer Layer(s) 1(0) ...

● Layer Specific

Layer	Operation	Bias [um]
1(0)	X-Y	0.100000
2(0)	Y	-.1



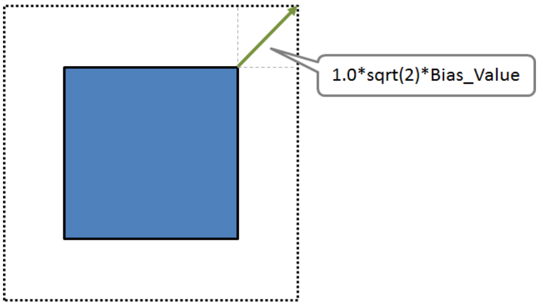
4.2.17.2 Corner Extension

The **Corner Extension** parameter in the **Advanced** tab controls the distance where these extensions are cut off. By default the Corner Extension (CE) is 1.0. We define the placement of a corner vertex using a bias with the formula:

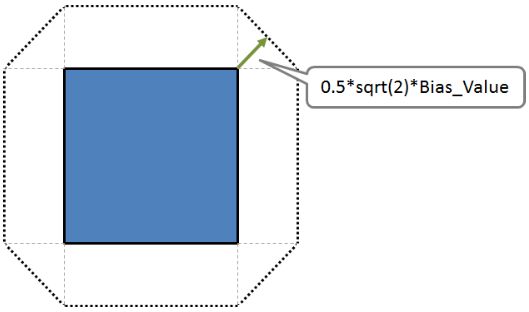
*CE*sqrt(2)*Bias_Value*

This formula yields the length of the orthogonal line that is bisecting the corner's angle. This length is the maximum distance that a corner vertex can be placed using a bias.

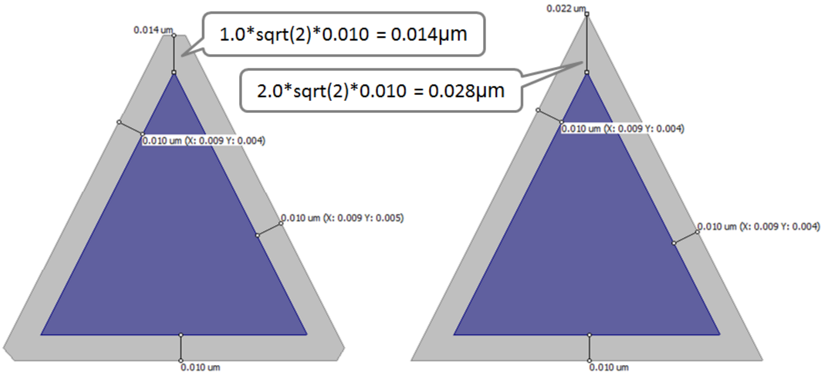
CE=1.0 means that the cutoff length is equal to the square root of two times the bias amount. Restated, the CE value imposes a limit on the new placement of a corner vertex with a bias. A simple example using a 90 degree corner illustrates the impact of the CE. The shape we wish to bias is in blue and the intended shape after a positive bias is outlined as dotted black. Below, CE=1.0 is used and the square is positively biased as expected with no changes in the corner value. To be clear, the formula does not dictate where a vertex is placed if CE >> 1; it simply allows the vertex to move without as much limitation as compared to a CE < 1.



By using a CE value of 0.5 when applying a positive bias, the corner is truncated by half, effectively rounding the corner in the litho. Since the limit of the corner placement is half, the limit imposes a cutoff on the corners as illustrated below. In this case, the formula says that a corner cannot extend beyond the calculated length of the line that is orthogonally bisecting the corner angle. As a result the corners are truncated.

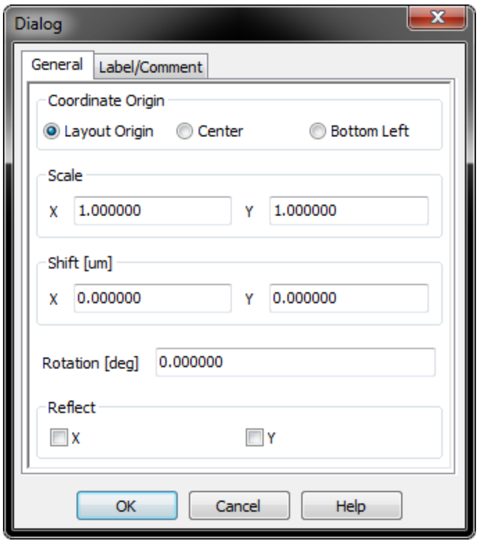


In particular, biasing triangles pose an interesting challenge. Illustrated below is an example that positively biases a triangle in blue to the triangle in gray by 10nm with CE=1. Since the distance of the original corner position to the biased position is limited to 14nm, the corners are truncated as seen on the bottom left. When CE=2 is used, the limit of the distance is increased to 28nm which allows the corner to be placed 22nm above the original position.


0.014 um
1.0*sqrt(2)*0.010 = 0.014µm
0.010 um (X: 0.009 Y: 0.004)
0.010 um (X: 0.009 Y: 0.005)
0.010 um
0.022 um
2.0*sqrt(2)*0.010 = 0.028µm
0.010 um (X: 0.009 Y: 0.004)
0.010 um (X: 0.009 Y: 0.004)
0.010 um

4.2.18 Transform

The Transform module enables the translation and rotation of the layout.



Coordinate Origin
● Layout Origin ○ Center ○ Bottom Left

Scale
X 1.000000 Y 1.000000

Shift [um]
X 0.000000 Y 0.000000

Rotation [deg] 0.000000

Reflect
□ X □ Y

Coordinate Origin

This defines the 0,0 point for the result of the Transform operation.

Layout Origin

Maintains the current coordinate system

Center

Moves the origin into the center of the layout extent

Bottom Left

Moves the origin to the lower left corner of the layout extent

This function is helpful to align different layouts to the same coordinate system, for example to compare fractured and original layouts against each other, as described in the Application Example section of this document.

Scale

This makes the pattern larger or smaller (magnify or shrink the data).

Shift

This will move the data the specified amount in X or Y.

Rotation

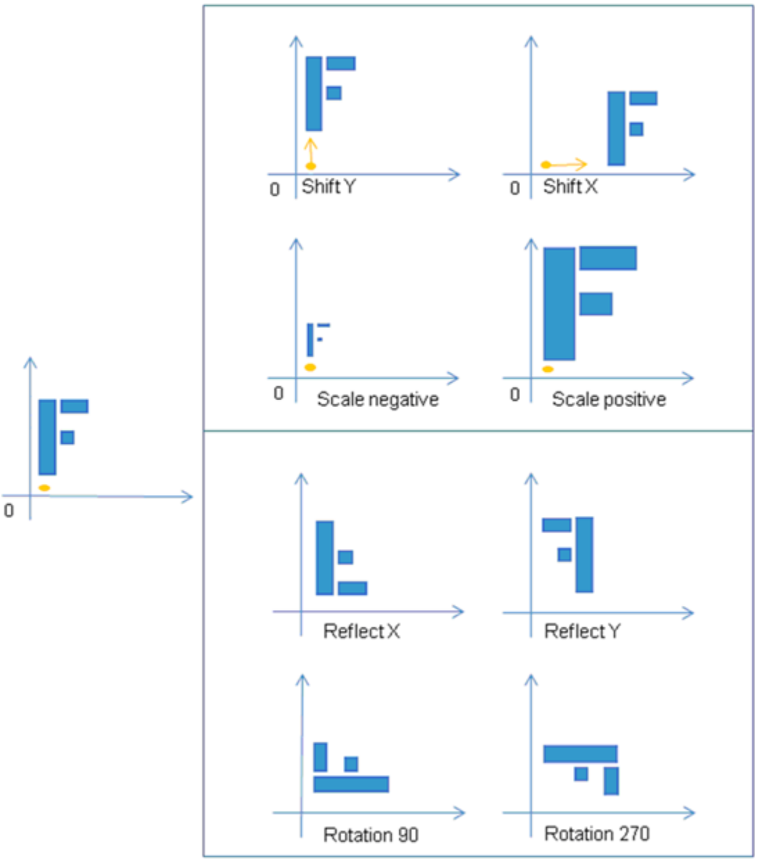
Allows the data to be rotated by any arbitrary angle

.

Reflect

This will mirror the data about the Y or X axis.

Using these functions, you can transform the data as shown.



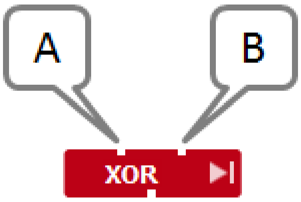
Since operations within the Transform module can be called simultaneously, it is important to know the processing order within the module. When in doubt, you can use multiple Transform modules in a sequence.

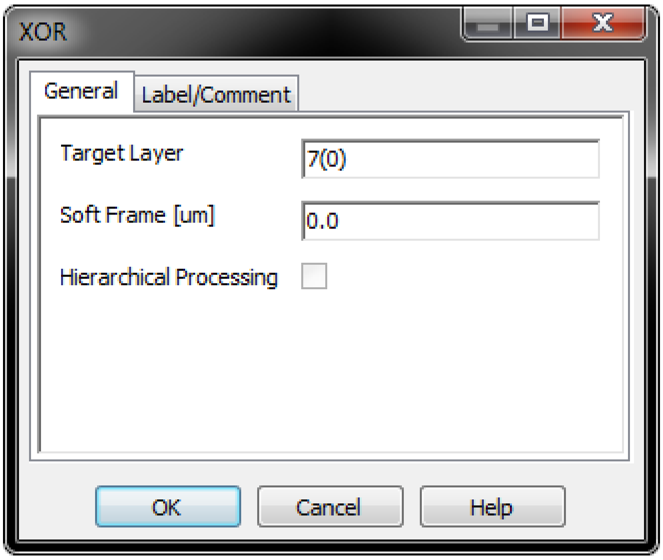
<table>
<tr><th colspan="3">Hierarchy of Transform Operations</th></tr>
<tr><td>Reflect</td><td>Coordinate Origin Transform (Center, Bottom Left)</td><td rowspan="2">Scale</td></tr>
<tr><td colspan="2">Rotate</td></tr>
<tr><td colspan="3">Shift</td></tr>
</table>

- Coordinate Origin Transform (Center, Bottom Left) work before Rotate
- Reflect work before Rotate
- Rotate works before Shift
- Scale works before Shift

4.2.19 XOR

The XOR module takes two layouts as input and returns data where Layout A data is present and Layout B data is not, and where Layout B data is present and Layout A data is not.





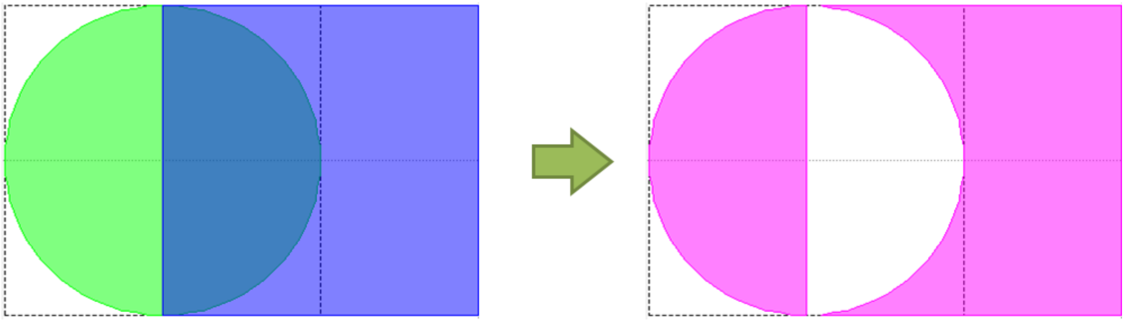
Target Layer 7(0)

Soft Frame [um] 0.0

Hierarchical Processing □

The target layer and datatype for the result can be specified by opening the parameter window with a double click. By default, the target layer and datatype is 7(0).To avoid unnecessary fracturing of the original layout a soft frame can be defined in the parameter window.

The XOR operation retains only the shapes unique to each pattern and removes all the overlaps between A and B in the output:



Layout Logic:

XOR		
A	**B**	**Result**
0	0	0
0	1	1
1	0	1
1	1	0

0 indicates a layout element is absent, 1 indicates a layout element is present

4.3 Process Correction

The Process Correction modules provide proximity effect correction (PEC) that can be tailored to your process specific needs using model- or numerical-based point spread functions (PSFs).

4.3.1 PEC

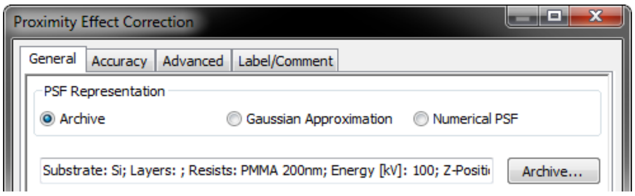
The PEC module performs process effect corrections for long-, mid- and short-range effects. Long-range effects might include electron backscattering, short-range effects might include electron forward scattering, beam blur and process artifacts, and mid-range effects might include addition electron effects such as secondaries as well as process artifacts such as loading effects and developer effects. The electron scattering behavior and the additional process parameters are specified using a Point Spread Function (PSF) consisting of either a sum of Gaussian functions, or a numerical lookup table.

See our FAQ: What is PEC?

See note about Multi-Threading and PEC.

4.3.1.1 General Tab

The General Tab offers users three options to represent a PSF: Archive, Gaussian Approximation and Numerical PSF. The options are described in this section.

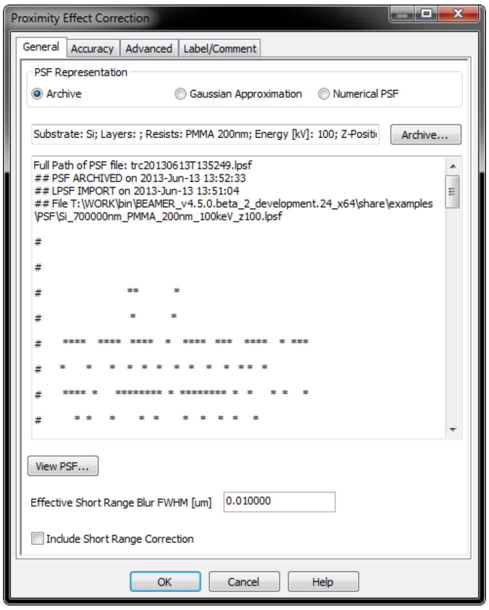


PSF Representation: ● Archive ○ Gaussian Approximation ○ Numerical PSF

Substrate: Si; Layers: ; Resists: PMMA 200nm; Energy [kV]: 100; Z-Positi

Archive

This is the default selection (see right). By clicking the **Archive** button, the PSF archive for BEAMER is loaded into view. From the archive, a PSF that closely matches your material stack and tool settings can be selected for use as the correction function in PEC. The location of the PSF archive is maintained under the BEAMER Properties under the Directories Tab.



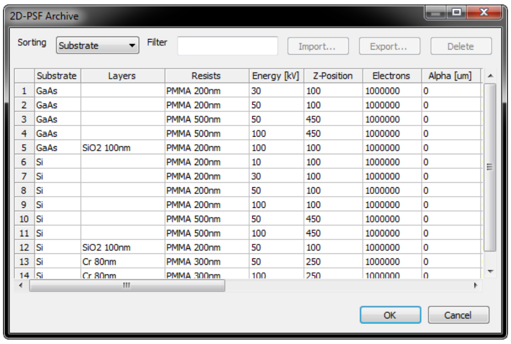
PSF Representation: ● Archive ○ Gaussian Approximation ○ Numerical PSF

Substrate: Si; Layers: ; Resists: PMMA 200nm; Energy [kV]: 100; Z-Positi

```
Full Path of PSF file: trc20130613T135249.lpsf
## PSF ARCHIVED on 2013-Jun-13 13:52:33
## LPSF IMPORT on 2013-Jun-13 13:51:04
## File T:\WORK\bin\BEAMER_v4.5.0.beta_2_development.24_x64\share\examples
\PSF\Si_700000nm_PMMA_200nm_100keV_z100.lpsf

#

#

#               **       *

#                *      *

#    ****  ****  ****  *  ****  ***  ****  * ***

#   *   *   *  *  *  *  *  *  *  *  ** *

#    **** *   ******** * ******** *  *   * *   *

#      * *   *    * *    *  *  *  *   *
```

Effective Short Range Blur FWHM [um]: 0.010000

□ Include Short Range Correction



Sorting: Substrate | Filter:

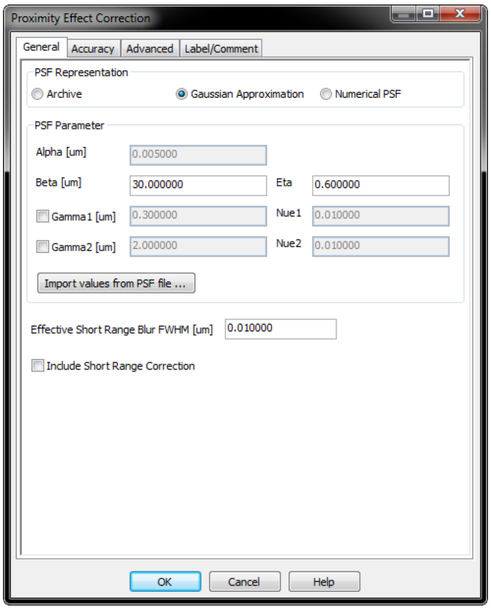
	Substrate	Layers	Resists	Energy [kV]	Z-Position	Electrons	Alpha [um]
1	GaAs		PMMA 200nm	30	100	1000000	0
2	GaAs		PMMA 200nm	50	100	1000000	0
3	GaAs		PMMA 500nm	50	450	1000000	0
4	GaAs		PMMA 500nm	100	450	1000000	0
5	GaAs	SiO2 100nm	PMMA 200nm	100	100	1000000	0
6	Si		PMMA 200nm	10	100	1000000	0
7	Si		PMMA 200nm	30	100	1000000	0
8	Si		PMMA 200nm	50	100	1000000	0
9	Si		PMMA 200nm	100	100	1000000	0
10	Si		PMMA 500nm	50	450	1000000	0
11	Si		PMMA 500nm	100	450	1000000	0
12	Si	SiO2 100nm	PMMA 200nm	50	100	1000000	0
13	Si	Cr 80nm	PMMA 300nm	50	250	1000000	0
14	Si	Cr 80nm	PMMA 300nm	100	250	1000000	0

Gaussian Approximation

Specifies the electron scattering and/or process effects in form of a multi-Gaussian function. The GUI (seen right) allows for the use of up two 4 Gaussians.

PSF Parameters (*Gaussian Approximation Mode*)

For definitions see section on [PSF Parameters](#).



PSF Representation
○ Archive ● Gaussian Approximation ○ Numerical PSF

PSF Parameter
Alpha [um] 0.005000
Beta [um] 30.000000 Eta 0.600000
□ Gamma1 [um] 0.300000 Nue1 0.010000
□ Gamma2 [um] 2.000000 Nue2 0.010000

Effective Short Range Blur FWHM [um] 0.010000

□ Include Short Range Correction

Numerical PSF

Specifies the electron scattering and/or process effects in form of a numerical lookup table file.

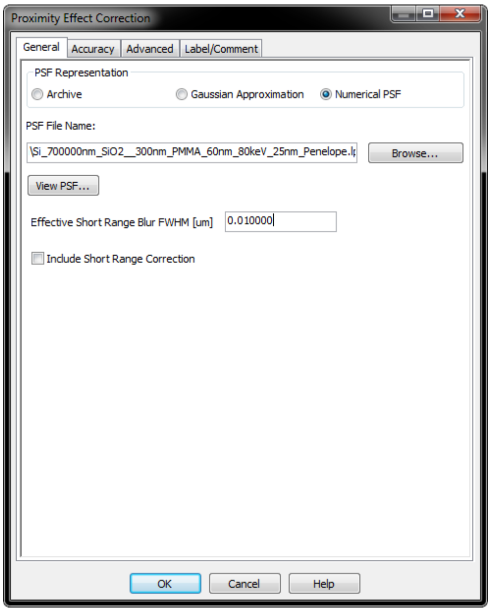
PSF File Name (*Numerical PSF Mode*)

Browse

Allows to load of a numerical PSF file.

Fit Gaussian ...

Described in [Fit Gaussian](#). Note that this conversion is just informational and will not be used in subsequent computations.



PSF Representation
○ Archive ○ Gaussian Approximation ● Numerical PSF

PSF File Name:
\Si_700000nm_SiO2__300nm_PMMA_60nm_80keV_25nm_Penelope.l

Effective Short Range Blur FWHM [um] 0.010000

□ Include Short Range Correction

Effective Short Range Blur FWHM [um]

The PSF described above models the material stack response to a point exposure (infinitely thin electron beam). In reality, no exposure is done using an infinitely thin electron beam - Gaussian Beam machines have a finite spot size, and Vector Shaped Beam machines have a beam blur that limit the edge acuity. The effective short range blur parameter allows users to specify the effective beam/process blur for exposure, which is of importance for short range correction. The effective beam size value is the Full Width Half Max (FWHM) of a Gaussian function.

Include Short Range Correction

Toggles the usage of short range correction mode. The decision for this mode is based on the smallest feature that needs correction and the ratio of its size against the short range process effects including the beam blur. As a rule of thumb, if the smallest feature / gap in the layout is larger than twice the smallest blur, no short range correction is required. See our FAQ [What is PEC?](#) to learn more about Short Range Correction and if you need it.

4.3.1.1.1 PSF Parameters

Alpha

Specifies the forward (short range) scattering range of the electrons.

Beta

Specifies the backward (long range) scattering range of the electrons.

Eta

The ratio of the energy in the beta Gaussian to the energy in the alpha Gaussian.

Gamma$_{1\ \&\ 2}$

Specifies the mid range scattering range of the electrons and/or process related effects.

Nue$_{1\ \&\ 2}$

The ratio of the energy in the mid range Gaussian against the energy in the alpha Gaussian.

Import values from PSF file ...

The values for *Alpha, Beta, Eta, ...* can be imported from a BEAMER lpsf file, or fitted to a numerical PSF table from a Monte Carlo run. The dialog is the same as the *Fit Gaussian* dialog in the Numerical PSF mode and is described in Fit Gaussian.

Note that *Alpha, Beta, Gamma$_1$* and *Gamma$_2$* are expressed as the sigma's in the related Gaussian distribution

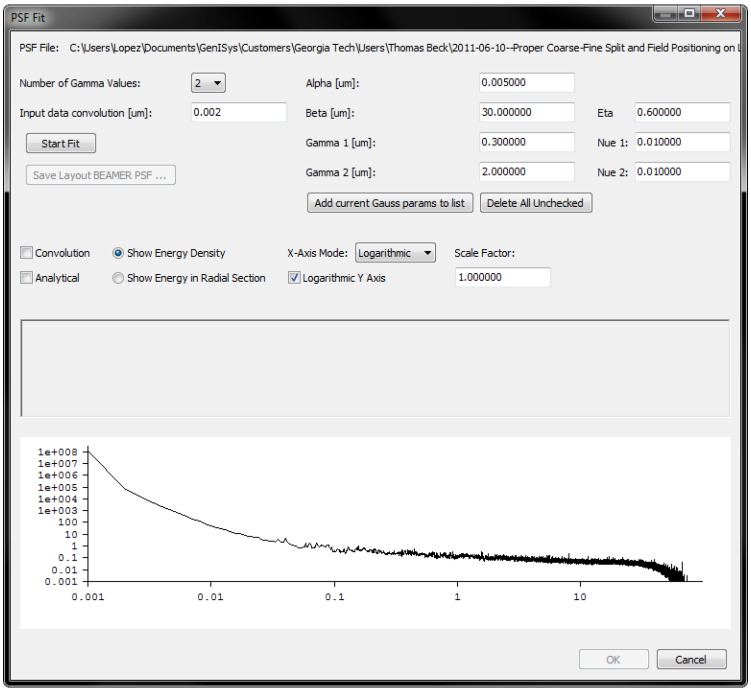
$$g_\sigma(x) = \frac{1}{\sqrt{\Pi} \cdot \sigma} \cdot e^{-\left(\frac{x}{\sigma}\right)^2} .$$

With that, the multi-Gaussian PSF can be written as

$$PSF(r) = \frac{1}{1+\eta+\nu_1+\nu_2}(g_\alpha(r)+\eta * g_\beta(r)+\nu_1 * g_{\gamma_1}(r)+\nu_2 * g_{\gamma_2}(r)) .$$

4.3.1.1.2 Fit Gaussian

The Fit Gaussian allows the graphical inspection of the loaded PSF data, and a conversion into a Gaussian PSF representation.



PSF Fit

PSF File: C:\Users\Lopez\Documents\GenISys\Customers\Georgia Tech\Users\Thomas Beck\2011-06-10--Proper Coarse-Fine Split and Field Positioning on L

Number of Gamma Values: 2

Input data convolution [um]: 0.002

Alpha [um]: 0.005000

Beta [um]: 30.000000

Eta 0.600000

Gamma 1 [um]: 0.300000

Nue 1: 0.010000

Gamma 2 [um]: 2.000000

Nue 2: 0.010000

□ Convolution

□ Analytical

● Show Energy Density

○ Show Energy in Radial Section

X-Axis Mode: Logarithmic

☑ Logarithmic Y Axis

Scale Factor: 1.000000

Number of Gamma Values

Sets the number of Gamma values to be used for the conversion. Setting this parameter to 0 will use the minimum of 2 Gaussians, since alpha and beta will always be used. There are up to 2 additional Gaussians possible to fit values for Gamma$_1$ and Gamma$_2$

Input data convolution [um]

A smoothing of the data values, that improves the numerical fit stability. A small value is typically sufficient, with the default of 2 nm small enough to not change the result significantly. The influence on the final result can be inspected visually by comparing the graphs of the input data with the convolved input data using the "Convolution" check box

Start Fit

Starts the conversion

Save BEAMER PSF

Saves the PSF and its characteristics as a BEAMER *.*lpsf* file

Alpha, Beta, Eta, ...

See PSF Parameters for details. These numbers are filled in once the conversion finishes, and are the used for drawing the fitted result in comparison to the input data

Add current Gauss params to list

Supports for a comparison view of different Gaussian parameters. Pushing this button adds the current set to a plot list showing all list entries and the original function one can determine which parameters are best suited.

Delete All Unchecked

Removes all unchecked entries from the plot list.

Convolution

Draws the numerical PSF data in comparison to the same function convolved with the value specified in *Input data convolution*.

Analytical

Draws the numerical PSF data in comparison to the multi-Gaussian PSF function, specified in the PSF parameter fields.

Show Energy Density

Draws the input data, and the optional graphs as a simple function of r (the functional form specified above).

Show Energy in Radial Section

Draws the input data and the optional graphs multiplied by r^2 in order to visualize the total energy deposited in a distance *r* from the point of exposure.

Logarithmic X-Axis

Scales the x axis logarithmically.

Quadratic X-Axis

The x axis is displayed as a x^2.

Logarithmic Y-Axis

Scales the y axis logarithmically.

Scale Factor

The factor is used to overlay the input data (that is specified in arbitrary units) and the fitted multi-Gaussian function (that is normalized). The factor is computed along with the PSF fit parameters but needs to be available as a user parameter since the user may chose to tune the fit parameters.

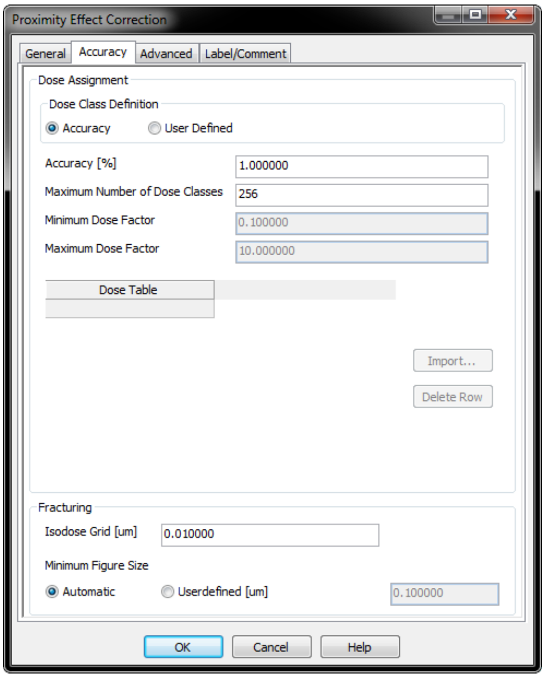
4.3.1.2 Accuracy Tab

Dose Class Definition

One of three modes are available to define the dose classes.

Accuracy

The number of dose classes will be determined using a specified relative dose accuracy. This mode is set as default since in practice there are many process variations (e.g. resist thickness variations, resist soft bake non-uniformities, gun current stability) limiting the usable dose accuracy to about 1%. The actual number of used dose classes is a function of the PEC dose range (that is a function of the layout coverage range) and the chosen accuracy. In other words, this mode creates a fracture only when the dose factor differs by more than the specified accuracy, and thereby avoids unnecessary fractures.



Dose Assignment

Dose Class Definition

● Accuracy ○ User Defined

Accuracy [%] 1.000000

Maximum Number of Dose Classes 256

Minimum Dose Factor 0.100000

Maximum Dose Factor 10.000000

Fracturing

Isodose Grid [um] 0.010000

Minimum Figure Size

● Automatic ○ Userdefined [um] 0.100000

User Defined

The BEAMER PEC will also accept a user specified dose classification table. This mode is used if there is a standardized process that always uses the same parameters (including a pre-defined dose table).

Accuracy Mode

Accuracy [%]
Specifies the relative dose accuracy to be used. The default is 1%, meaning that a geometry will only be fractured if the dose factors within this geometry differ by more than 1%

Maximum Number of Dose Classes
Specifies the maximum number of dose classes to be used. This is used, if the E-Beam tool is limited in the usable number of dose classes. In that case one can make sure that the dose classification does not exceed the available dose classes

Maximum Dose Value
Controls the highest possible value the PEC is allowed to assign to a feature. **Short Range Correction** must be turned on for this feature to be enabled.

Minimum Dose Value
Controls the lowest possible value the PEC is allowed to assign to a feature. **Short Range Correction** must be turned on for this feature to be enabled.

User Defined Mode

Dose Table
Allows the user to specify the dose table manually. With each entry entered, the table will add a new row for the next value

Import Table
Allows the user to import the dose classes from an external file

Delete Row
Allows the user to delete an entry in the table

Note that in *User* mode there is no maximum dose value as in the other dose class definition modes. The highest value in the user specified dose table will be the maximum usable dose value.

Fracturing

Isodose Grid
Describes the fracture grid or the possible locations for fractures so that the algorithm can assign different dose factors to a feature.

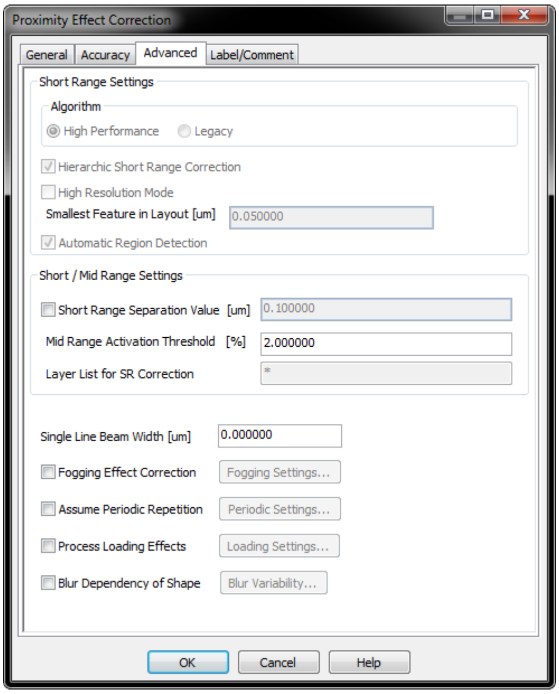
Minimum Figure Size
The size of features that will not be fractured. Features smaller then this value will not be fractured any further. LayoutBEAMER can compute a solution for the minimum figure size by taking the PSF and beam blur into account or the user can set the parameter.

Dose Table
This table defines the dose classes that are to be used by the dose assignment algorithms. A table can be read in from a file or manually being defined with this table.

4.3.1.3 Advanced Tab

Short Range Settings

When short range correction is selected in the General Tab, this section is active.



Short Range Settings

Algorithm

● High Performance ○ Legacy

☑ Hierarchic Short Range Correction

□ High Resolution Mode

Smallest Feature in Layout [um] 0.050000

☑ Automatic Region Detection

Short / Mid Range Settings

□ Short Range Separation Value [um] 0.100000

Mid Range Activation Threshold [%] 2.000000

Layer List for SR Correction *

Single Line Beam Width [um] 0.000000

□ Fogging Effect Correction — Fogging Settings...

□ Assume Periodic Repetition — Periodic Settings...

□ Process Loading Effects — Loading Settings...

□ Blur Dependency of Shape — Blur Variability...

Algorithm

The short range correction is based on one of two possible algorithmic implementations.

High Performance

Using sparse computation, the implementation is much faster and more accurate with respect to mid-range terms than the grid based algorithm offered in previous versions of BEAMER.

Legacy

Using a grid based computation, with the grid size related to the size of alpha. In case of the layout containing features that are smaller then alpha, setting the grid manually can improve accuracy of the correction.

Hierarchic Short Range Correction

When activated the short range PEC uses hierarchy; the center of an array is kept. This option is very effective for lines and spaces or contact holes where the pitch is larger than the SR influence range. Outside areas are flattened.

High Resolution Mode

This feature is only available for the legacy short range algorithm. In this case, the short range correction is using a grid based computation, with the grid size related to the size of alpha. In case of the layout containing features that are smaller then alpha, setting the grid manually can improve accuracy of the correction.

Smallest Feature in Layout

Enter the size of the smallest feature size within the layout.

Automatic Region Detection

BEAMER detects layout features that need a short range correction automatically. If desired this detection can be turned of and overwritten by defining layers which contain those elements that need to experience a short range correction.

Layer List for Correction

Defines the layer list for the short range correction.

Short / Mid Range Settings

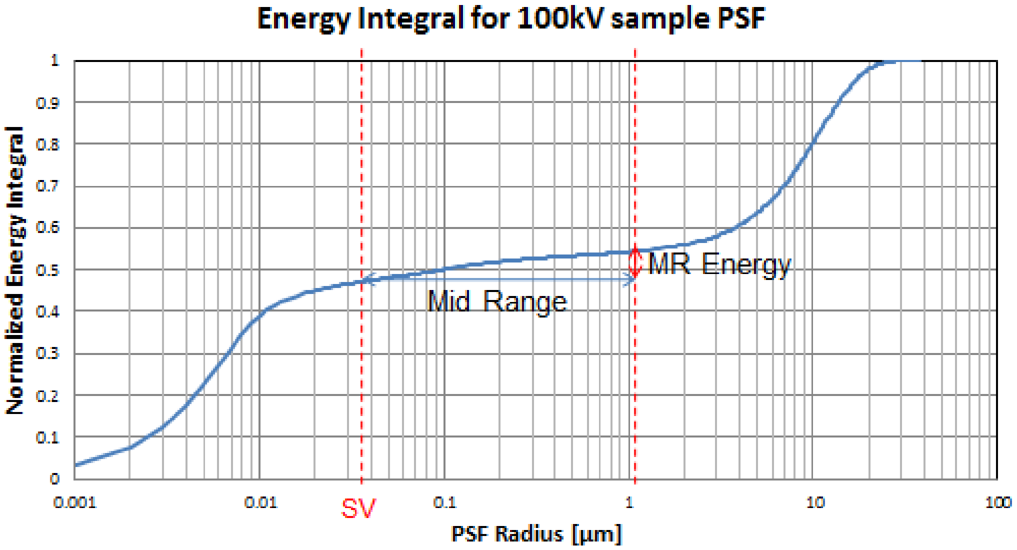
The correction of the mid range PSF energies can now be considered separately from the long or short range terms. Previously every energy contribution above 10% was accounted to the long range correction. Now with the possibility to enter a threshold it is possible to more precisely separate the long and mid range correction. This adds to the quality of the correction and better represents energy spread of the PSF.

Short Range Separation Value

Determines the radial cut location of the PSF into the short range and long range energy cushions. Any energies left of the cut will account to the short range energy and those right of it to the long range energy. With complex PSFs that have a strong mid range energy contribution this allows the precise separation for the correction. As the separation point is used for the energy integration you can observe different eta values.

Mid Range Activation Threshold

Determines the amount of energy necessary to trigger the mid range correction function. Should the energy in the mid range region of the PSF exceed the defined percentage of the long range energy the mid range correction is performed.



If MR_Energy > Mid Range Activation Threshold, the MR PEC is computed separately, otherwise its energy is considered in the LR PEC.

Assume Periodic Repetition

For large arrays of repeating structures the correction is much faster when only a single cell based is corrected and then repeated. The periodic repetition expects a single cell in the layout, and user input about the periodicity in x and y. The correction will then take these boundary conditions into account.

Periodic Settings

Opens the Periodic Settings dialog, that allows to specify the boundary conditions.

Pitch X, Y

Defines the pitch of the array.

Repetition X, Y

Repeats the corrected cell so the output of the PEC will become an array.

Process Loading Effects (Resist Effects)

Influence Range

When **Process Loading Effects** or **Resist Effects** is switched on, its influence range needs to be specified. A good first guess is the distance of resist residuals in large pads from the clear edge region.

Density Table

This is a table of local densities and the relative development rate. The local density is computed within the specified influence range, and the relative development rate can be adjusted to the observed process behavior. For example, if the loading effect requires a 20% increase in dose for large pads to be cleared completely, the relative development rate would be 1/1.2. Having no loading effects is equivalent to have a relative development rate of 1 for all local densities.

Delete Row

Deletes a row entry in the table.

4.3.2 Shape-PEC

The Shape PEC module allows the geometrical modulation of the layout to compensate for short and mid range effects in electron scattering and process behavior.

4.3.2.1 General Tab

See General Tab in PEC for parameter definitions in this tab.

Options

Dose Factor

controls the shape variation of the short range correction such that the under sizing with the over dosing will result in the target layout. The over dose factor is also taken into account for the long range PEC as it carries additional energy into the exposure.

4.3.2.2 Accuracy Tab

See Accuracy Tab in PEC.

Geometry

Shift Step Size [um]

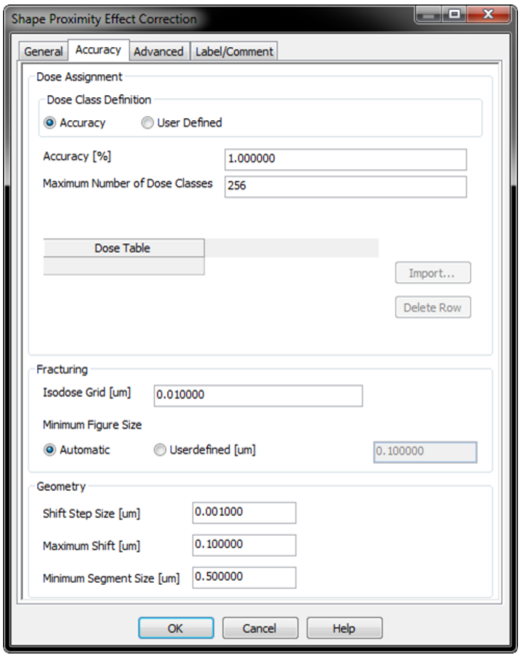
Defines the step size of the shape modulation. A value of 1nm will allow to move the edges in 1nm increments.

Maximum Shift [um]

Defines the maximum shift allowed for the shape modulation.

Minimum Segment Size [um]

Defines the minimum segment length of an edge segment that is moved in one piece.



Shape Proximity Effect Correction

Dose Assignment

Dose Class Definition

● Accuracy ○ User Defined

Accuracy [%] 1.000000

Maximum Number of Dose Classes 256

Fracturing

Isodose Grid [um] 0.010000

Minimum Figure Size

● Automatic ○ Userdefined [um] 0.100000

Geometry

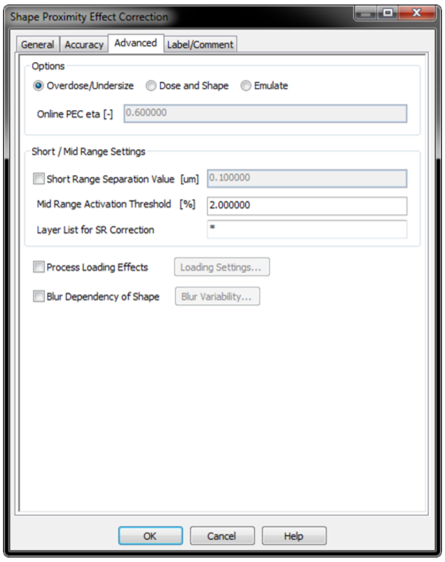
Shift Step Size [um] 0.001000

Maximum Shift [um] 0.100000

Minimum Segment Size [um] 0.500000

4.3.2.3 Advanced Tab

Options



Options

● Overdose/Undersize ○ Dose and Shape ○ Emulate

Online PEC eta [-] 0.600000

Short / Mid Range Settings

□ Short Range Separation Value [um] 0.100000

Mid Range Activation Threshold [%] 2.000000

Layer List for SR Correction *

□ Process Loading Effects

□ Blur Dependency of Shape

Undersize/Overdose

Performs an undersize/overdose shape correction based on the input at the general tab. This modulates the dose and the shape of the layout.

Dose and Shape

Performs the long range correction and computes the short range shape correction. The dose factor acts as an threshold indicator and creates the shape at a threshold of 0.5/ dose factor instead of the 0.5 as the default works, assuming a dose factor of 1.

Emulate

Emulates the long range correction and computes the short range shape correction as a correction for the effects that are not corrected on the tool. This mode is required for machines with an integrated long range PEC (such as the Nuflare tools).

Process Loading Effects (Resist Effects) & Short / Mid Range Settings

Please see the description in the PEC / Advanced tab here.

Shape Size Dependency of the Blur

Used for variable shaped beam (VSB) writers where the blur is changing with the shot size

ElementSize [um]

Size of the shape being written.

Blur Factor [-]

Multiplication to alpha based on the written shape.

Delete Row

Deletes a row entry in the table.

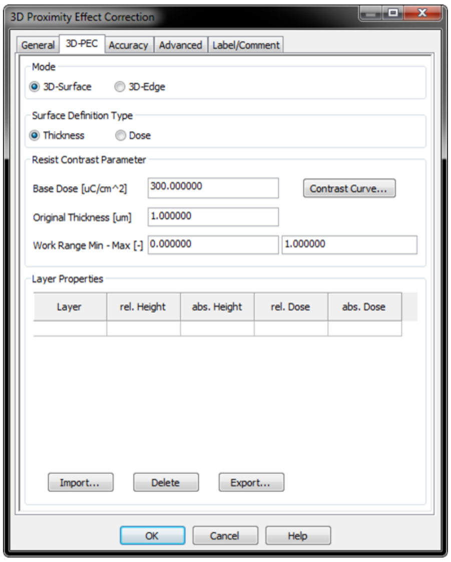
4.3.3 3D-PEC

The 3D PEC module calculates the dose corrections needed to compensate for proximity effects in the process of generating a three dimensional structure in resist.

4.3.3.1 General Tab

See General Tab in PEC.

4.3.3.2 3D-PEC Tab



Mode
● 3D-Surface ○ 3D-Edge

Surface Definition Type
● Thickness ○ Dose

Resist Contrast Parameter
Base Dose [uC/cm^2] 300.000000
Original Thickness [um] 1.000000
Work Range Min - Max [-] 0.000000 1.000000

Mode

3D Surface
Performs dose calculations that result in the resist having a specified thickness

3D Edge
Performs dose calculations to ensure that multi-layer resists are exposed such that the individual layers clear while maintaining accuracy of the lateral feature dimensions

3D Surface Mode

- Surface Definition Type

Thickness
Assigns layers certain relative height levels, the dose calculations are derived from the *Resist Contrast Parameter.*

Dose
Assigns layers certain dose levels that would develop the resist away to a certain height

- Resist Contrast Parameters (Thickness mode only)

Base Dose
Absolute value [µC/cm²] to define the relative 1.0 dose.

Original Thickness
Original pre-etch resist thickness required to calculate the absolute target thicknesses.

Work Range Min - Max[-]
Relative resist height (default 0.0, 1.0). Base Dose - defines the absolute exposure dose

- Layer Properties

Layer
Defines the layer that is assigned either a relative height or relative dose

rel. Height
Defines the relative height to be achieved by the dose modulation

rel. Dose
Defines the relative dose used for the exposure

abs. Height
Defines the absolute height to be achieved by the dose modulation

abs. Dose
Defines the absolute dose used for the exposure

The computation of the parameters is based upon a contrast curve that must be entered using the "Contrast Curve " button. There is an import option to use ASCII based files, which give dose [µC/cm²] and absolute height [um] such as:

```
100,      1.000
```

110,	1.000
120,	0.970
...	...

The layer properties can also be imported using an ASCII based table. Here layer name and target height are entered using the same naming convention:

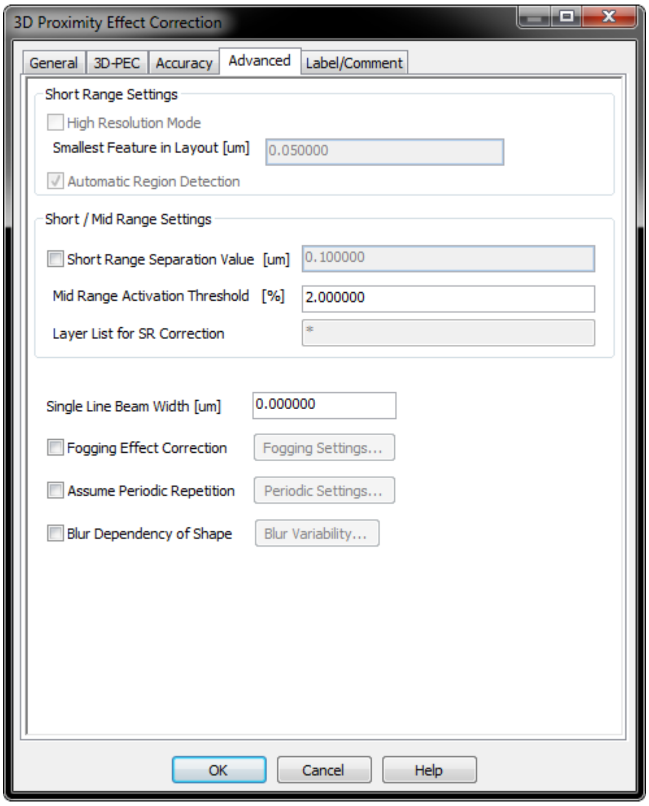
1,	0.800
2,	0.700
3,	0.600
...	...

4.3.3.3 Accuracy Tab

See Accuracy Tab in PEC.

4.3.3.4 Advanced Tab

See Advanced Tab in PEC for parameter definitions in this tab.



Short Range Settings

□ High Resolution Mode

Smallest Feature in Layout [um] 0.050000

☑ Automatic Region Detection

Short / Mid Range Settings

□ Short Range Separation Value [um] 0.100000

Mid Range Activation Threshold [%] 2.000000

Layer List for SR Correction *

Single Line Beam Width [um] 0.000000

□ Fogging Effect Correction

□ Assume Periodic Repetition

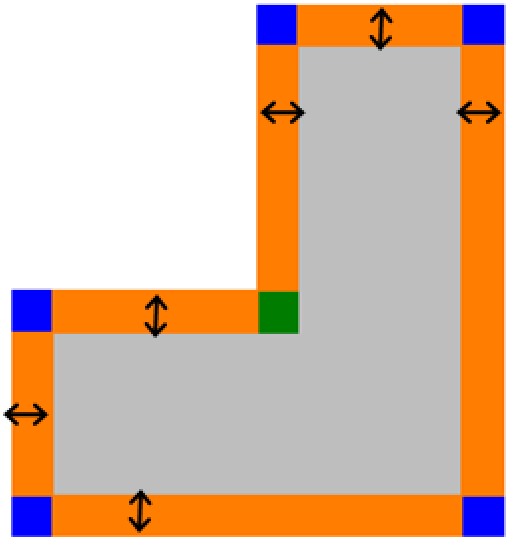
□ Blur Dependency of Shape

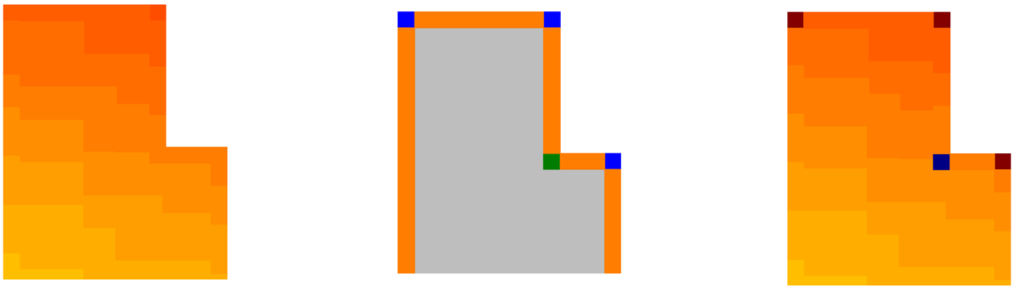
4.3.4 Corner-PEC

The Corner PEC module facilitates *corner sharpening* by rule based dose adjustments at feature edges and corners. The correction method uses the model based correction with Gaussian parameters and combines a rule based correction by breaking down the layout and giving dose multiplication factors. The breakdown of the layout is done using the *Size* parameter.

This *Size* parameter creates a bulk (grey), edges (orange) and corner elements (inner corner is green and outer corners are blue).The size of the corner elements depends on the corner angle. For each of these elements a dose factor is defined.

The layout is corrected as a whole, decomposed and then the dose factors are applied on those areas.

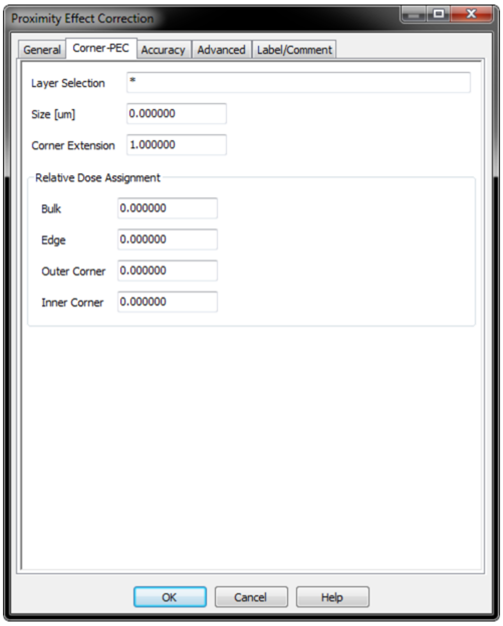




4.3.4.1 General Tab

See General Tab in PEC.

4.3.4.2 Corner-PEC Tab



Layer Selection *
Size [um] 0.000000
Corner Extension 1.000000
Bulk 0.000000
Edge 0.000000
Outer Corner 0.000000
Inner Corner 0.000000

Layer Selection

Defines the layers which are to be corrected by the Corner PEC module.

Size

Defines the border width (or size) in microns when applying the corner PEC algorithm.

Corner Extension

The Corner Extension parameter controls the distance where these extensions are cut off. A Corner Extension of 1.000 means that the cutoff is equal to the square root of two times the bias amount, from the original corner, orthogonal to a line bisecting the corner's angle.

Relative Dose Assignment

Bulk

Multiplication factor for the Bulk portion of the decomposed shape.

Edge

Multiplication factor for the Edge portion of the decomposed shape.

Outer Corner

Multiplication factor for the Outer Corner portion of the decomposed shape.

Inner Corner

Multiplication factor for the Inner Corner portion of the decomposed shape.

4.3.4.3 Accuracy Tab

See Accuracy Tab in PEC.

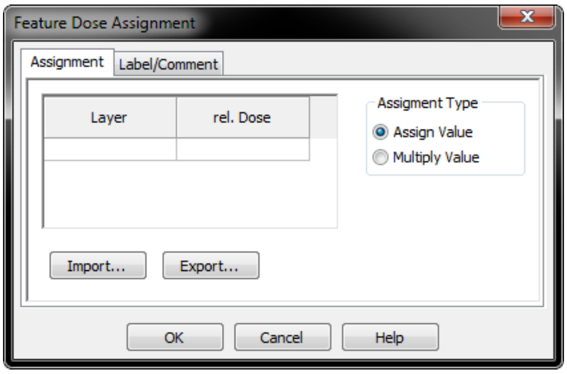
4.3.4.4 Advanced Tab

See Advanced Tab in PEC.

4.3.5 FDA

Feature Dose Assignment (FDA) enables the user to assign doses to features in the layout. Dose values can be assigned to layer(datatype) pairs. You can also assign new dose values to a proximity effect corrected layout by either a true value assignment or a multiplication of the existing dose values.

Please see the FDA section in the Advanced Application section of this manual.



Layer	rel. Dose

● Assign Value
○ Multiply Value

The **Layer / rel. Dose** table is used to manually assign a dose value to a particular layer(datatype) pair. When each entry is complete, a new row will be added automatically.

Import ...

Imports the dose table from an external file.

Export ...

Export the dose table to an external file.

Assignment Type

Assign Value

Assigns the this value to all geometries in the specified layer. Existing dose values will be overwritten.

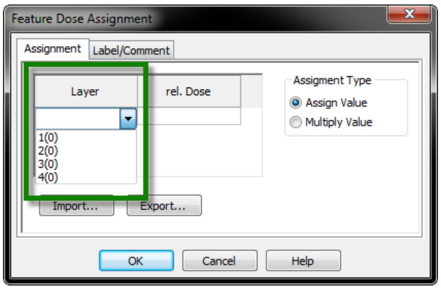
Multiply Value

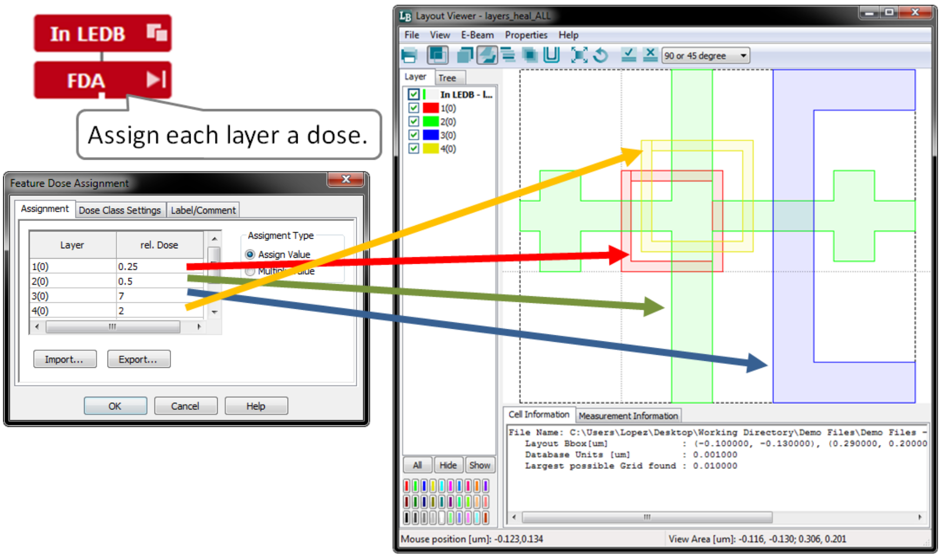
Multiplies the existing dose value by this value. In other words, pre-existing dose values will not be overwritten, but scaled with the specified value.

4.3.5.1 Assigning a Dose by Layer

In the example below, a layout in the LEDB format is shown in the VIEWER to the bottom right. The pattern consists of random shapes on layers 1(0), 2(0), 3(0), and 4(0). Using the FDA the user simply assigns a relative dose of 0.25 to layer 1(0), 0.5 to layer 2(0), 7 to layer 3(0) and 2 to layer 4(0).

If the module before the FDA has already been executed, layers from the pattern will be loaded automatically in a drop down in every table in the FDA dialog as shown to the right.





Layer	rel. Dose
1(0)	0.25
2(0)	0.5
3(0)	7
4(0)	2

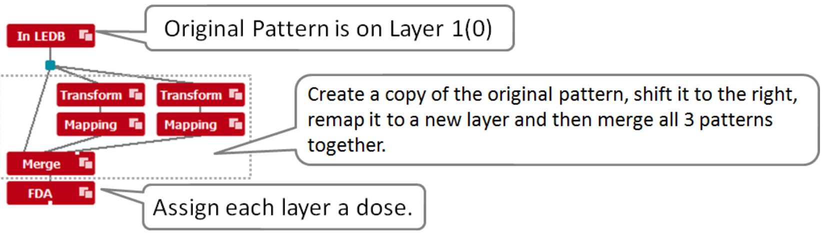
```
File Name: C:\Users\Lopez\Desktop\Working Directory\Demo Files\Demo Files -
   Layout Bbox[um]             : (-0.100000, -0.130000), (0.290000, 0.20000
   Database Units [um]         : 0.001000
   Largest possible Grid found : 0.010000
```

Mouse position [um]: -0.123,0.134

View Area [um]: -0.116, -0.130; 0.306, 0.201

4.3.5.2 Dose Matrix_Creation

In this example, we want to take a line space pattern on layer 1(0) and create copies on different layers and apply a different dose to each. The pattern has not been treated by PEC. In the flow below, the pattern is imported and then sent to a splitter to replicate the datapath. Each datapath from the splitter is treated differently. The edge to the far left simply enters the Merge module unaltered. This means that the original pattern in this datapath remains untouch. The middle edge is transformed.



Original Pattern is on Layer 1(0)

Create a copy of the original pattern, shift it to the right, remap it to a new layer and then merge all 3 patterns together.

Assign each layer a dose.

4.4 Verification

The Verification modules offer two modules for E-beam simulation and metrology functionality.

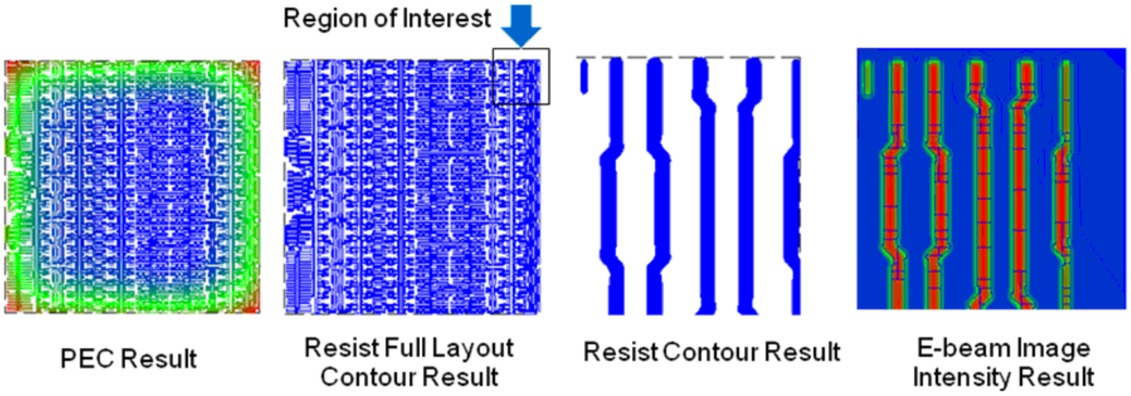
4.4.1 E-Beam

Description

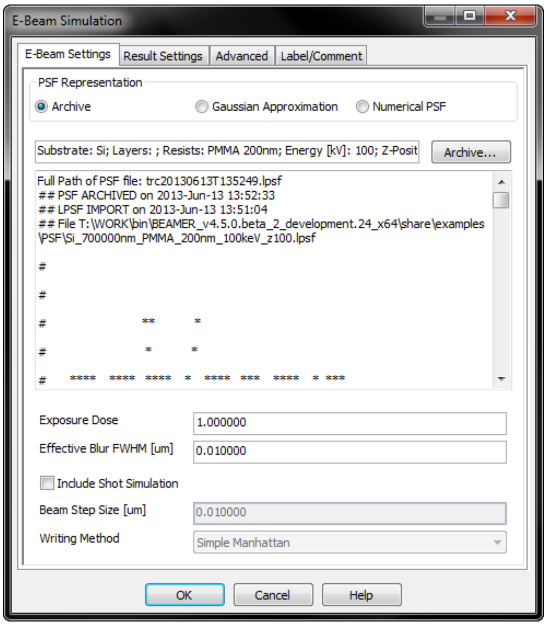
The E-Beam module simulates the energy density of the e-beam exposure in the resist. Simulation is very valuable for saving experiment time (e-beam, sample processing, SEM, etc.). Simulating a layout’s critical features while optimizing correction and writing parameters before writing can provide a clear understanding of effects and help in verification.

Function

The E-Beam module input is a non-corrected layout or a proximity effect corrected layout. In addition, a point spread function (PSF), describing the energy deposition in a radius around the beam, needs to be defined. The simulation is done in 2D (X-Y plane), at the Z-position where the PSF has been extracted. The PSF representation is equivalent to the settings in the PEC module. If the E-Beam module is used following a PEC module within the same flow, the default PSF settings will be automatically transferred from the PEC module to the E-Beam module.


Region of Interest
PEC Result
Resist Full Layout Contour Result
Resist Contour Result
E-beam Image Intensity Result

4.4.1.1 E-Beam Settings Tab



PSF Representation
● Archive ○ Gaussian Approximation ○ Numerical PSF

Substrate: Si; Layers: ; Resists: PMMA 200nm; Energy [kV]: 100; Z-Posit

```
Full Path of PSF file: trc20130613T135249.lpsf
## PSF ARCHIVED on 2013-Jun-13 13:52:33
## LPSF IMPORT on 2013-Jun-13 13:51:04
## File T:\WORK\bin\BEAMER_v4.5.0.beta_2_development.24_x64\share\examples
\PSF\Si_700000nm_PMMA_200nm_100keV_z100.lpsf

#

#

#             **     *

#              *     *

#   ****  ****  ****  *  ****  ***  ****  * ***
```

Exposure Dose: 1.000000
Effective Blur FWHM [um]: 0.010000
□ Include Shot Simulation
Beam Step Size [um]: 0.010000
Writing Method: Simple Manhattan

Exposure Dose

The exposure dose is the relative exposure dose value. The default is 1.0 but can be changed to simulate exposures at different dose values.

Include Shot Simulation

The checkbox controls whether the shots are simulated or the whole layout is done in a flood exposure.

Effective Short Range Blur FWHM [um]

The PSF described above models the material stack response to a point exposure (infinitely thin electron beam). In reality, no exposure is done using an infinitely thin electron beam - Gaussian Beam machines have a finite spot size, and Vector Shaped Beam machines have a beam blur that limit the edge acuity. The effective short range blur parameter allows users to specify the effective beam/process blur for exposure, which is of importance for short range correction. The effective beam size value is the Full Width Half Max (FWHM) of a Gaussian function. This is equivalent to the 12/88 width in the knife edge method.

Beam Step Size (BSS) or Shot Pitch

Shot pitch is the distance between shots when filling the shapes. The shot pitch is a multiple of the pattern units (writing grid resolution). It is advisable to keep the shot pitch equal to or smaller than the beam size. If you set the shot pitch larger than beam size, you will see roughness at geometry edges. This parameter is only available when **Include Shot Simulation** is on.

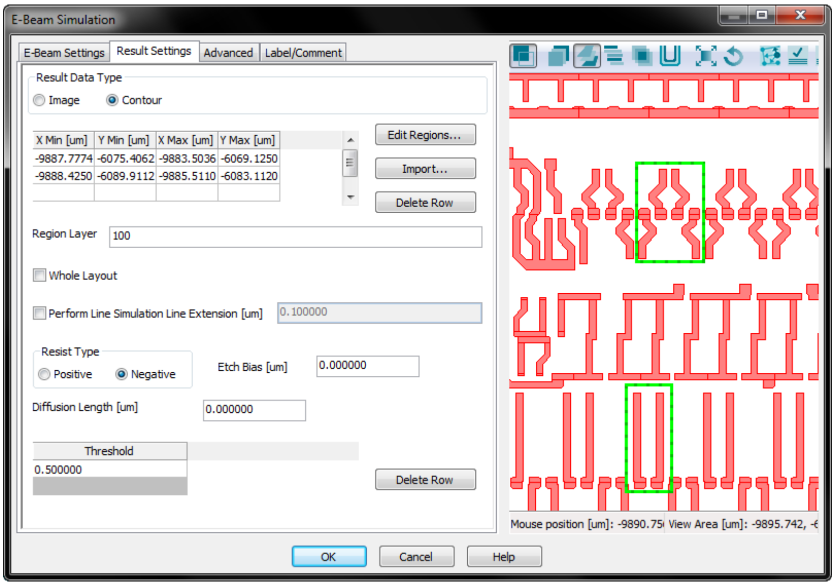
Writing Method

You can choose the available writing methods: Simple Manhattan or an e-beam machine format (if available). This option is only available when the checkbox **Include Shot Simulation** is active.

See General Tab in PEC for the rest of the parameter definitions in this tab.

4.4.1.2 Result Settings Tab

When doing an e-beam simulation, you have two modes for the type of result to be modeled: **Image** or **Contour** of the layout. Independent of which mode is used, **Image** or **Contour**, the result of an E-Beam simulation is available for subsequent modules like Merge, MINUS or Export.



Result Data Type
○ Image ● Contour

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
-9887.7774	-6075.4062	-9883.5036	-6069.1250
-9888.4250	-6089.9112	-9885.5110	-6083.1120

Region Layer 100

□ Whole Layout

□ Perform Line Simulation Line Extension [um] 0.100000

Resist Type
○ Positive ● Negative

Etch Bias [um] 0.000000

Diffusion Length [um] 0.000000

Threshold
0.500000

Mouse position [um]: -9890.75 View Area [um]: -9895.742, -

Image - Interactive Intensity Image

Interactive Intensity Image mode is used for inspecting a large layout in critical areas. Using the Edit Regions button, areas are defined for which the image intensities can be viewed later. The maximum area of the simulation = 1024 x (alpha*beam blur) / 2. After executing the module, the Viewer will display the defined regions as intensities views. This mode is useful to verify critical areas of the layout and get information on deposited energies. Process effects can be covered by using an etch bias and a resist diffusion length parameter. These are Gaussian functions that are convolved in addition to the PSF convolution.

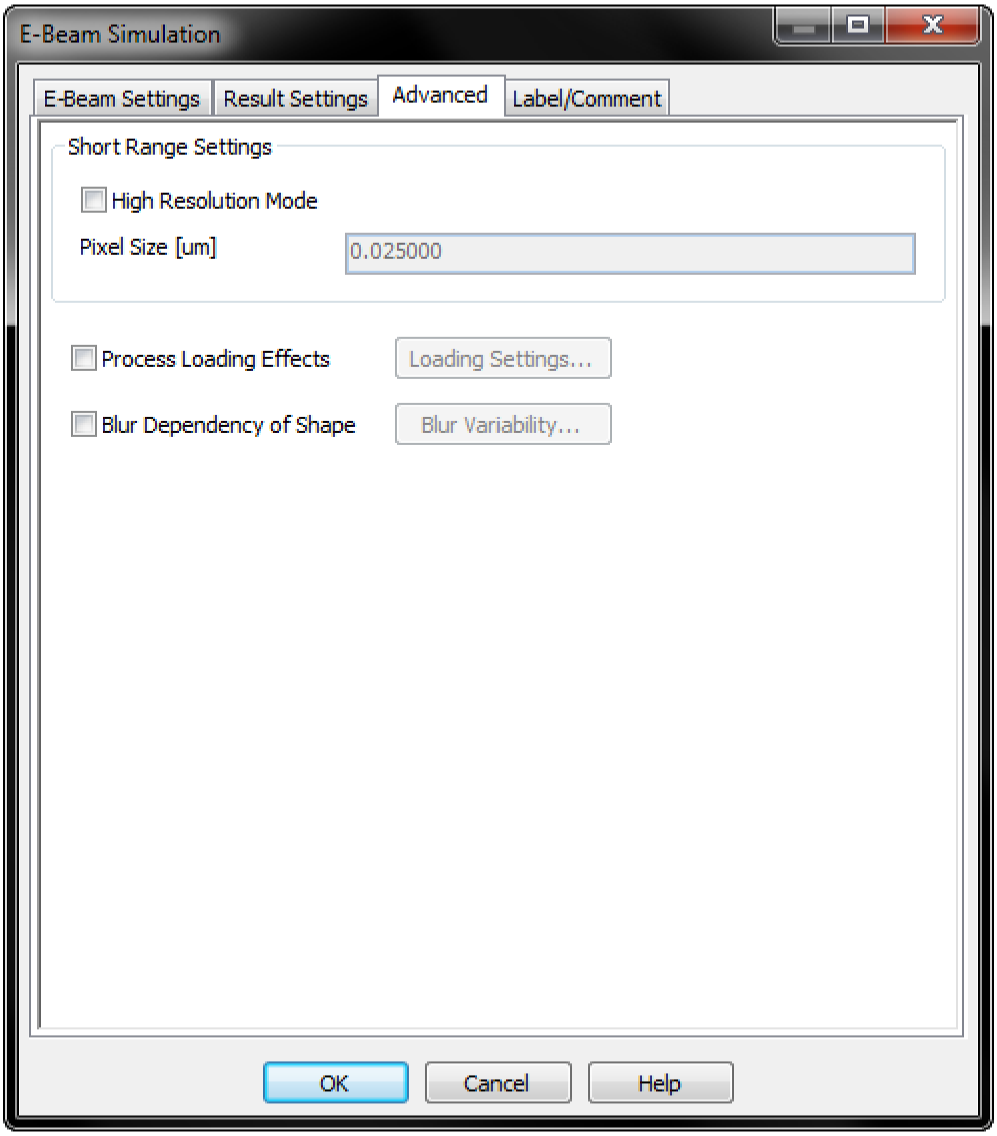
Contour - Full Layout Contour

Full Layout Contour mode simulates the full layout's resist contour at the defined thresholds and not be the intensity image. The E-Beam module also combines the functionality of the resist contour generation with **Resist Type**, **Diffusion Length** and **Threshold** parameters that are accessible in the same parameter window. The result of this mode can be processed like a layout. This enables applications like calculating the difference between the resist contours and original layout, which is useful for optimizing the correction and performing automated runs. The simulation time of the Full Layout Contour mode depends on the size of the layout and may take longer times. The result of this mode is the resist contours as polygons, not the intensity image.

Selected Simulation Regions

You can choose a Region Definition from a layout or a table. When choosing the Region Definition from a table, you can import a Measurement Result File (.txt). You can also check the optional box to perform a 1D simulation if possible and either specify a Region Layer or interactively add a Region Layer by choosing the Add Region button.

4.4.1.3 Advanced Tab



Short Range Settings

□ High Resolution Mode

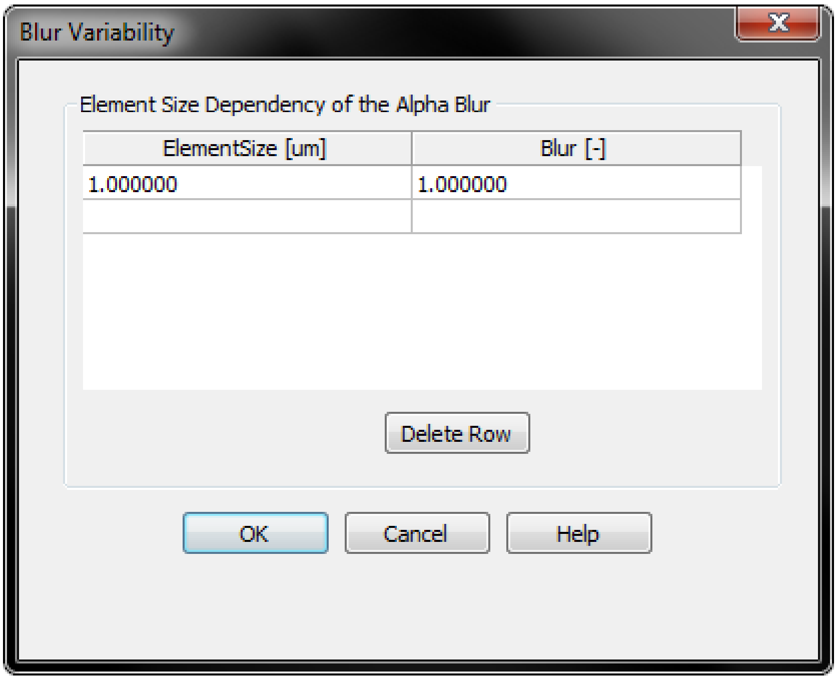
Pixel Size [um] 0.025000

□ Process Loading Effects

□ Blur Dependency of Shape

High Resolution Mode

These settings control the resolution used for the image simulation. Activating this checkbox will allow you to overwrite the internally computed pixel size with a finer value. Should the value you have entered be larger the algorithm will use the smaller internal value.



Blur Variability

Element Size Dependency of the Alpha Blur

ElementSize [um]	Blur [-]
1.000000	1.000000

Blur Variability ...

Element Size Dependency of the alpha blur - is used to take into account that for shaped beam machines forward scattering, alpha, is dependant on the size of the shape.

See [Advanced Tab]() in [PEC]() for the definitions of **Process Loading Effects**.

4.4.2 Metrology

Description

The Metrology Module is used to measure a CD at a defined position after the simulation module. It also allows importing measured data for optimization of the correction function.

Function

Metrology lines can be pre-defined in a layout file or defined in a table. You can choose a specific layer and/or datatype for the measurement. Measurements that have been pre-defined using the Layout VIEWER can be loaded into Metrology and specific coordinates can be entered or modified in the table for layer(s) that require measurements. Multiple cut lines on multiple layers can be set in a single file for measurement.

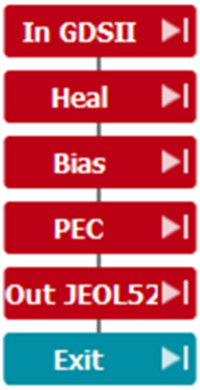
The Metrology module can be used to measure the CD after simulation at the defined metrology lines. The measured CD's can be viewed and saved in data sheets. The Analysis view displays the measured CD as a function of a varied parameter in a loop.

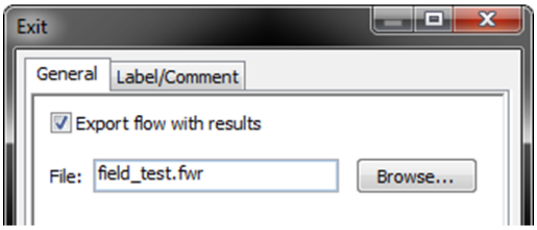
4.5 Control

Flow Control modules facilitate with the execution and flow of data in your flow(s). Each are described in greater detail in the following subsections.

4.5.1 Exit

The Exit module allows a flow to exit BEAMER automatically to prevent license server bottlenecks. It can be configured to save the results in the *.fwr format before closing BEAMER to analyze the completed flow at a later date when Export flow with results is checked and a file name is provided. The Browse button allows the user to select the directory to which the *.fwr file will be saved.





☑ Export flow with results

File: field_test.fwr

4.5.2 Loops

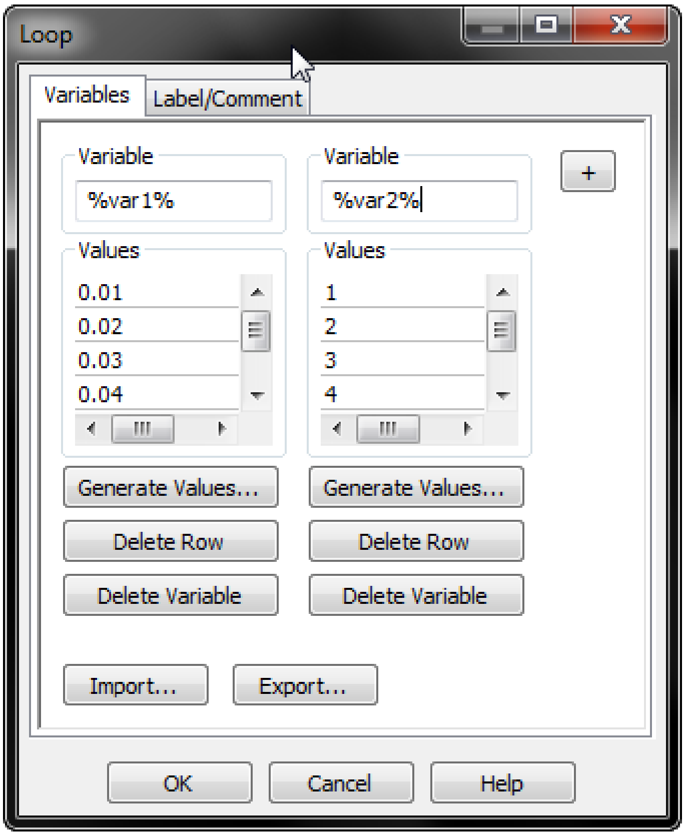
Description

The Loop module is used in combination with variables for optimization and automation. It provides flexible variable definition where values can be:

- Defined in a table (variable steps)
- Automatically generated
- Imported from a file

The module also provides the ability to loop over strings such as filenames. Finally, it can be also be used for multiple coupled variables, where every loop iteration will set corresponding variable values row by row.

A loop in BEAMER consists of two modules: the top Loop module, where parameters are defined, and the End Loop module, where the Loop operation finishes.
A Run-To button is located at the End Loop module. Clicking on this will execute the loop with all parameters. Clicking on any module within a loop will only run the loop with the first parameter set defined. There is no step wise execution of the loop available.



Variable: %var1%

Values: 0.01, 0.02, 0.03, 0.04

Variable: %var2%

Values: 1, 2, 3, 4

Function

Define the variables that are run inside the loop. The **+** button adds variables to this loop axis. Variable names must be declared with a percent symbol (%) at the start and end of the name as illustrated above.

Values

A list can be given to loop over the variables.

Generate Values...

A dialog used to generate a list of linear steps for the values list.

Delete Row

Deletes a value of the variable.

Delete Variable

Deletes the variable from the loop.

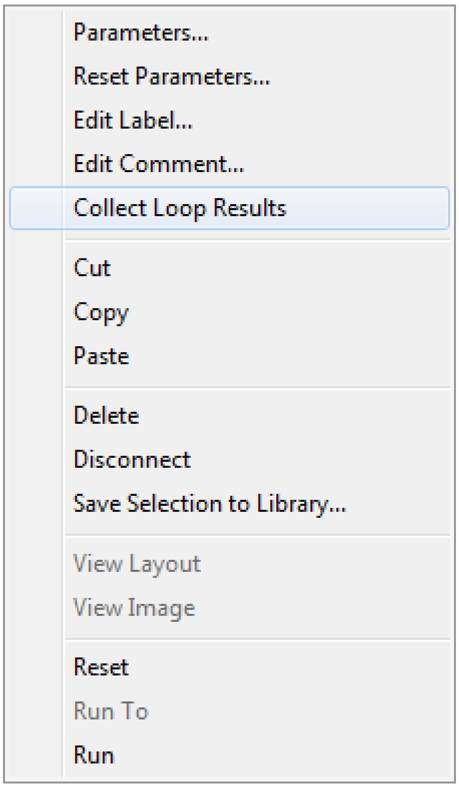
Import...

Loads the variables into the loop module.

Export...

Writes to a file on disk the variables of the loop module.

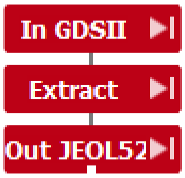
The option **Collect Loop Results** allows the visualization of each parameter variation done within a loop. The results of that visualization cannot be exported.



4.5.2.1 Batch File Conversion

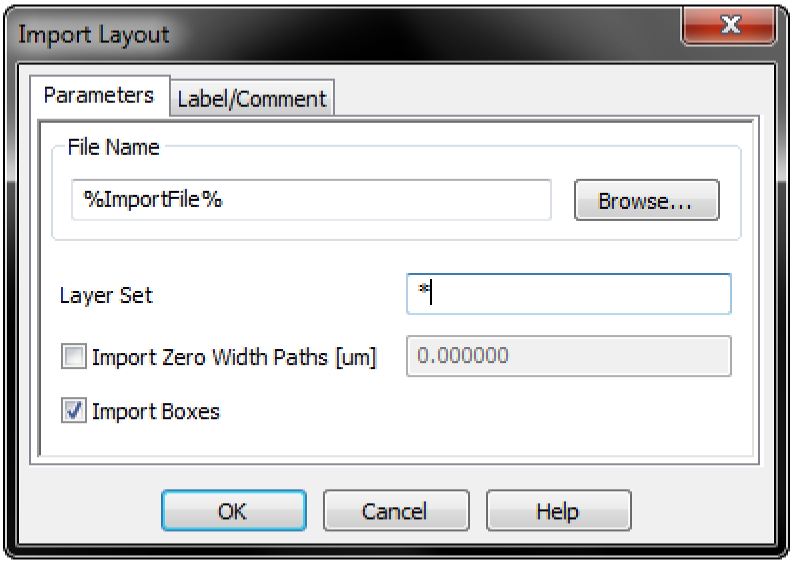
Loops can be particular helpful when larger amount of files must be processed. Here is an example of such a flow. We will load a specific file, extract a layer of interest and export to a v30 file.

In principle this flow looks like this.



Now to prepare this flow for a loop we must parametrize the Import name, the Extract layer and the Export file name.

Import:



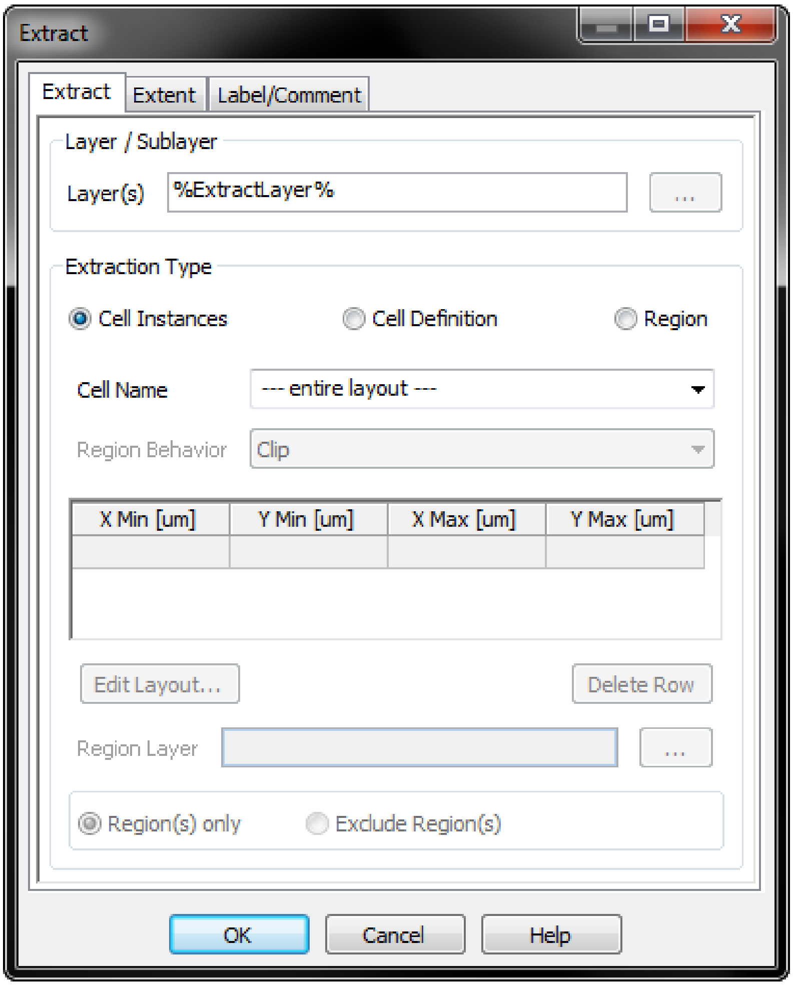
File Name: %ImportFile%

Layer Set: *

☐ Import Zero Width Paths [um]: 0.000000

☑ Import Boxes

Extract:



Layer / Sublayer

Layer(s): %ExtractLayer%

Extraction Type

● Cell Instances ○ Cell Definition ○ Region

Cell Name: --- entire layout ---

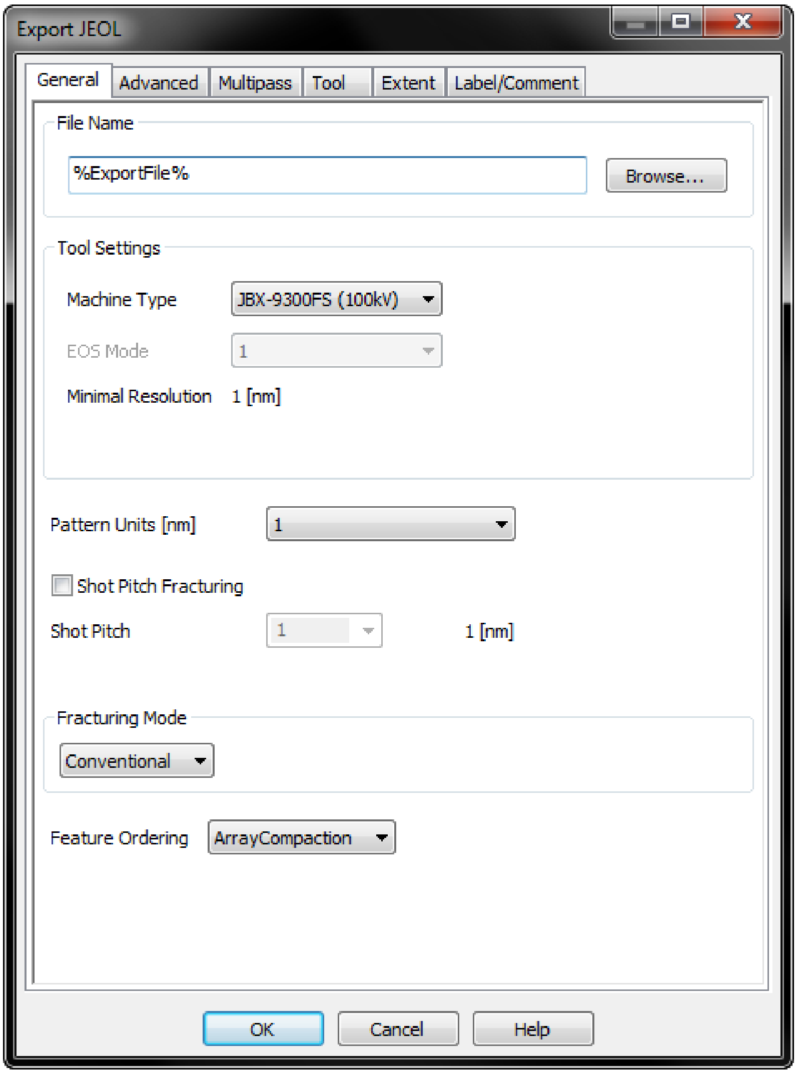
Region Behavior: Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

Region Layer

● Region(s) only ○ Exclude Region(s)

Export:



File Name

%ExportFile%

Tool Settings

Machine Type JBX-9300FS (100kV)

EOS Mode 1

Minimal Resolution 1 [nm]

Pattern Units [nm] 1

□ Shot Pitch Fracturing

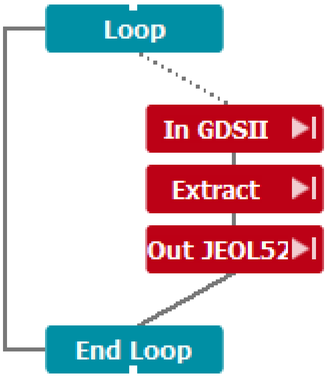
Shot Pitch 1 1 [nm]

Fracturing Mode

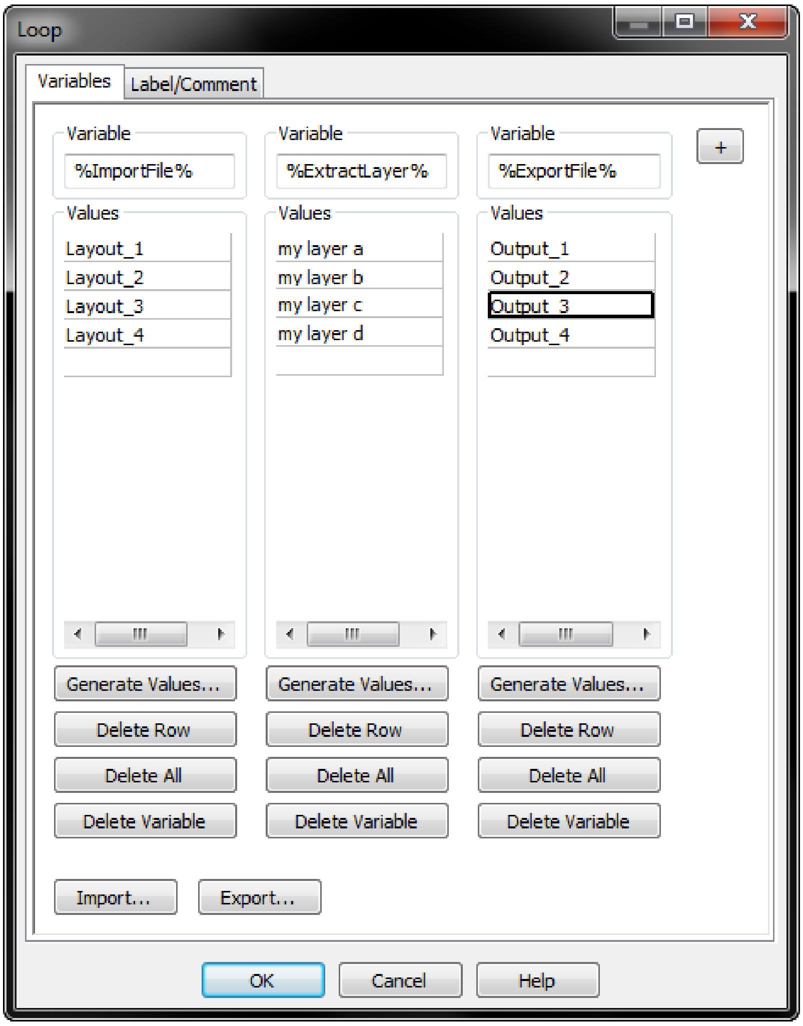
Conventional

Feature Ordering ArrayCompaction

This we wrap now in a LOOP:



The parameters of the LOOP shall be set such that we used parameters coupled. This means that each variables is stepped one by one through the index. So as the ImportFile variable is set, the Extract Layer is also set as well as the Export file name.



Variable	Variable	Variable
%ImportFile%	%ExtractLayer%	%ExportFile%
Layout_1	my layer a	Output_1
Layout_2	my layer b	Output_2
Layout_3	my layer c	Output_3
Layout_4	my layer d	Output_4

This is the scheme to process multiple files in a LOOP. These tables can be Export and prepared such a way to process large amounts of files. With the Import the Loop is than using the prepared variables.

4.5.3 Optimizer

Description

The optimizer module varies user defined variables and takes measurements at defined positions. The aim is to optimize the variable value such that parameter variations meet a target value. The algorithm being used is a differential evolution method which varies a defined space for the best value pairs to meet the target requirement.

Function

Optimizer Module

- Parameter tab
  - Variable - name of the variable
  - Start - initial value for the variable
  - Lower - lowest value this variable can reach
  - Upper - highest value this variable can reach

- Advanced tab
  - Population Size - maximum number of population within a generation is set

- Number of Generations - the maximum number of parameter variations

Metrology Module
This module should only be used in table mode. It allows the definition of the target values, typically CD values. The position and the length of the line define the target of optimization.

For more details, go here.

4.5.4 Script

Description

BEAMER has two script execution modes. The fist one is a trivial command line execution that allows you to execute a command within a shell. The other script capability is to launch a shell command from within BEAMER. With this you can use variables and other control modules available within BEAMER.

Function

Internal scripting allows the execution of shell commands from within BEAMER and this can be used to pass over parameters that are within loops to an external shell and process them.
To make handling easier a few predefined variables are provided:

- %IMPORT% ==> placeholder for the last imported file of the current process flow
- %EXPORT% ==> placeholder for the last exported file of the current process flow

4.5.5 Split

Description

The Splitter module connects to a preceding module and makes the output available to multiple successive modules.

Function

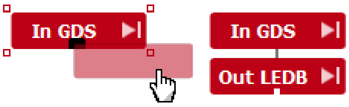
A Splitter module acts like any other module, e.g., right clicking opens a context menu where results can be viewed, etc.

5 Mouse and Keyboard Shortcuts

For quick reference, a few mouse and keyboard shortcuts that are used throughout BEAMER are outlined here for your convenience.

5.1 Visual Flow Editing

The Process Flow Design Window has a few keyboard and mouse combinations that are quickly summarized in the this table:

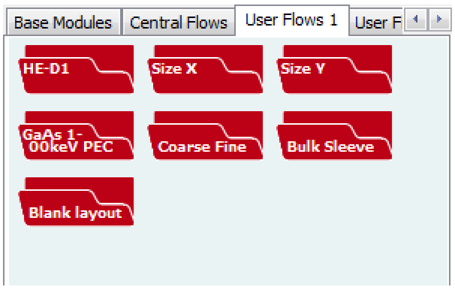
Action	Mouse/Keyboard Shortcut
Select Module(s)	Two options exist for selecting module(s) in the Visual Flow. • Left click on a module to select. To select additional modules, hold the SHIFT key while left clicking on other modules. The module(s) will be selected. • Hold down the left mouse button and drag the mouse. A black selection rectangle will appear as you drag the mouse. Allow the selection rectangle to partially touch the module(s) of interest and release the left mouse button.The module(s) will be selected.
Copy Selected Module(s)	CTRL + C
Cut Selected Module(s)	CTRL + X
Paste Copied or Cut Module(s)	CTRL + V
Inserting a Module into Process Flow Design Window	Two options exist for inserting module(s) in the Visual Flow. • Drag and drop the module from the Base Module library to the Process Flow Design Window. • Double-click the module in the Base Module library; it will automatically be added to the Process Flow Design Window.
Attaching a new Module to an Existing Module 	Three options exist: • Standard Method: Drag a Base Module into the Process Flow Design Window and touch the mouse cursor to the output port of the module to which to attach and release the left mouse button to attach. • Faster Method: This option is a two-step procedure: 1. Drag and drop a Base Module into the Process Flow Design Window. 2. To attach the output port of an existing module to the input port of the newly placed module: Using the right mouse button, click and hold anywhere on the existing module and drag from the existing module to the newly placed module, releasing the right mouse button when the cursor touches the new placed module. • Fastest Method: This option is a two-step procedure: 1. Select a module in the Process Flow Design Window to which to attach. 2. In the Base Module library, double-click with the left mouse button the desired module; it will be automatically be added to the Process Flow Design Window and attached to the selected module in step one.

5.2 Module/Operation

Module/Operation	Mouse/Keyboard Shortcut
Extract: Region Selection Export: Manual Field Placement E-Beam: Region Selection	This is a three-step procedure: 1. Set starting corner of bounding box: SHIFT + Left Mouse Click. Let go of the SHIFT key. 2. Move the mouse to set the size of the bounding box. 3. To confirm the size: SHIFT + Left Mouse Click. Let go of the SHIFT key when finished.
Metrology: Line Measurement Creation	1. While holding the SHIFT key click the Right Mouse button to set the starting point for the measurement line: SHIFT + Right Mouse Click. Let go of the SHIFT key. 2. Move the mouse to a desired location in the VIEWER. 3. To confirm the size: SHIFT + Right Mouse Click. Let go of the SHIFT key when finished.

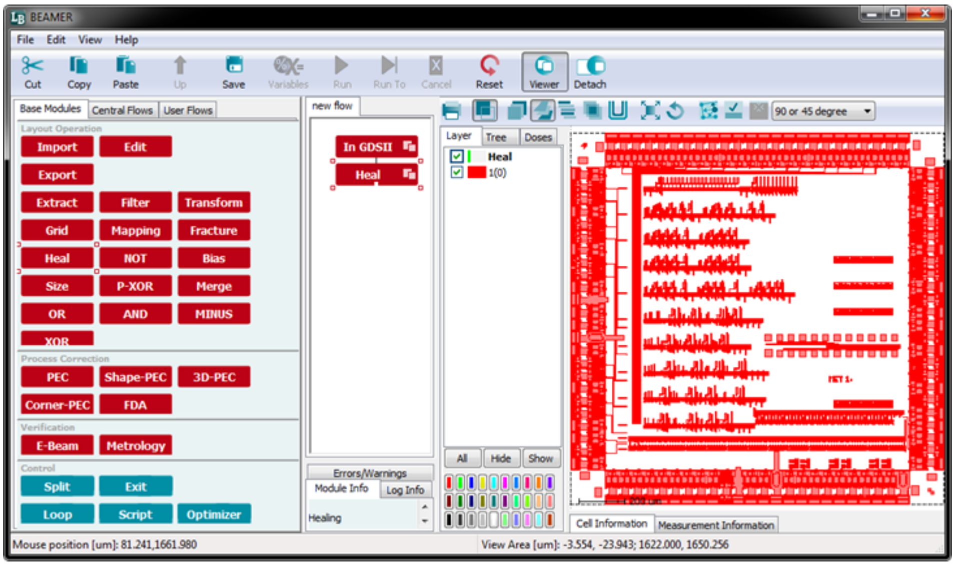
6 Database Modules

In addition to the Base Module Library, BEAMER has an area where the new functional modules can be saved in the BEAMER database, hence their name database modules. These database modules are single, user-defined modules that are an encapsulation of base modules that work together to create new and reusable functionality within BEAMER. Database modules behave like base modules. Depending on their creation, they can be ran individually or connected to other base/database modules.

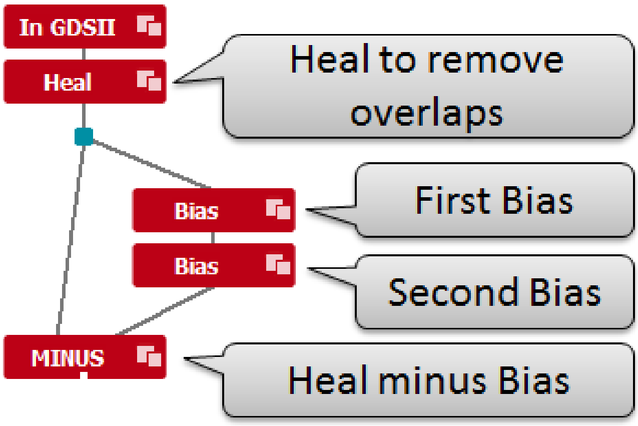


6.1 Creating a Database Module: Extending BEAMER's Functionality

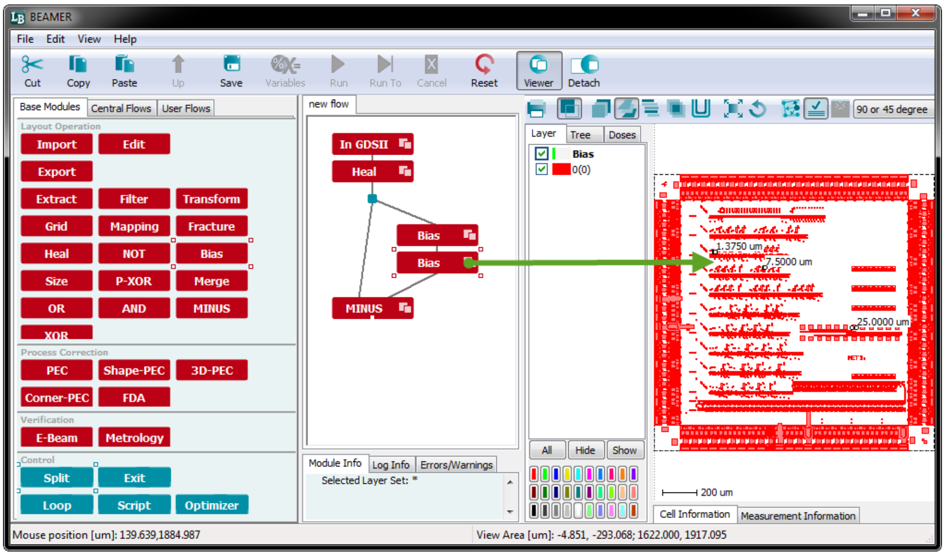
In this section we will generate a simple database module called a **Coarse-Fine Split**. For this example, we will import a pattern and heal out the overlaps. The goal is to separate any features that are larger than 1um from features that are equal to or smaller that 1um in size. Many applications make use of the Coarse-Fine Split. For example, the large (coarse) features could be exposed using a large current, while the small (fine) features are exposed using a lower current. The problem here is that since the pattern is already healed to remove overlaps (to avoid double exposure and poor shape fidelity in the printed image), one would think there is no way to separate the pattern without the use of a CAD program.


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer
new flow
In GDSII
Heal
Layer Tree Doses
Heal
1(0)
All Hide Show
90 or 45 degree
Errors/Warnings
Module Info Log Info
Healing
Cell Information Measurement Information
Mouse position [um]: 81.241,1661.980
View Area [um]: -3.554, -23.943; 1622.000, 1650.256

Instead, what we will do is use an advanced Boolean operation called Coarse-Fine Split. Here we want to produce the flow to the right. From the Heal module we attach a Split module. From the Split module we attach two Bias modules in succession. The first Bias module applies a -0.5um bias on every side of the shapes in the pattern. This means that any shapes smaller than 1um are erased. Any remaining patterns however are restored to size using the second Bias module with a 0.5um bias.



This means that the second Bias module only contains the features larger than 1um as illustrated below:

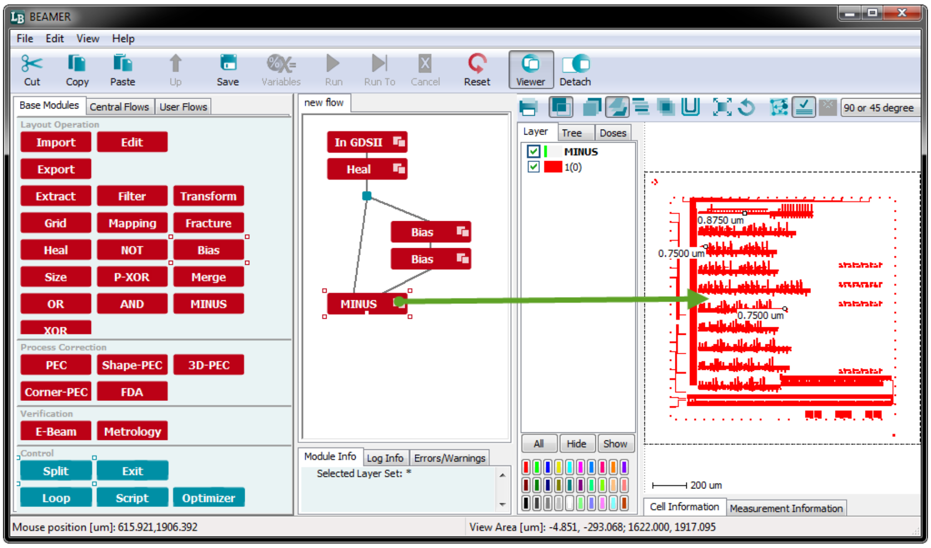

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer
new flow
In GDSII
Heal
Bias
Bias
MINUS
Module Info Log Info Errors/Warnings
Layer Tree Doses
Bias
0(0)
All Hide Show
90 or 45 degree
1.3750 um
7.5000 um
25.0000 um
200 um
Cell Information Measurement Information

Selected Layer Set: *

Mouse position [um]: 139.639,1884.987

View Area [um]: -4.851, -293.068; 1622.000, 1917.095

Remember that the goal is to separate the pattern into small and large features. Since we have the large features in the second Bias module as shown, to obtain the small features, we subtract the large features from the second Bias from the original pattern in the Heal module using the MINUS module. Therefore when the MINUS module executes, the shapes with a size equal to or less than 1um will be in the MINUS module. This is illustrated in the next image.

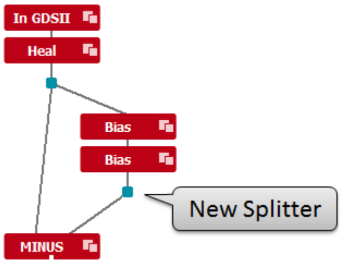

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer
new flow
In GDSII
Heal
Bias
Bias
MINUS
Module Info Log Info Errors/Warnings
Layer Tree Doses
MINUS
1(0)
All Hide Show
90 or 45 degree
0.8750 um
0.7500 um
0.7500 um
200 um
Cell Information Measurement Information

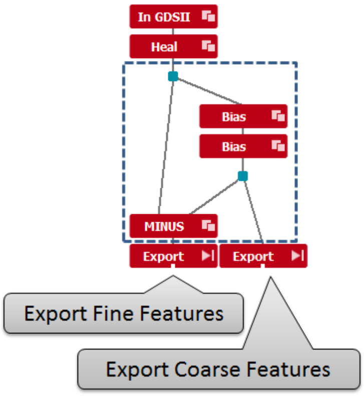
Selected Layer Set: *

Mouse position [um]: 615.921,1906.392

View Area [um]: -4.851, -293.068; 1622.000, 1917.095

At this point in time, we have no way of pulling out the coarse features and exporting them to their own file. To do this, we will add a splitter to the second Bias module. From here we can now attach Export modules to the flow to write the coarse and fine parts to a separate files. This is illustrated in the following image:

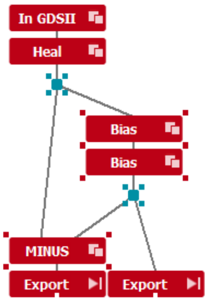




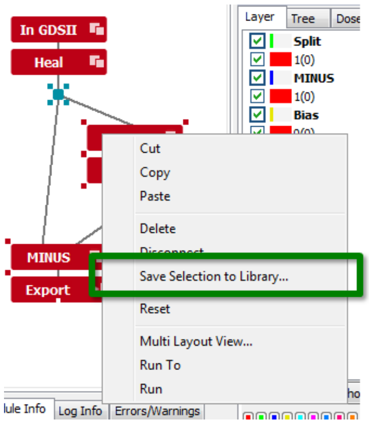
Interestingly enough, we have generated a new function in BEAMER called the Coarse-Fine Split. The modules needed for this operation are outlined in the dashed blue line in the left image. Since the Coarse-Fine Split operation may be used again, we could take these modules and save them as one database module.

To create a database module from follow these simple steps:

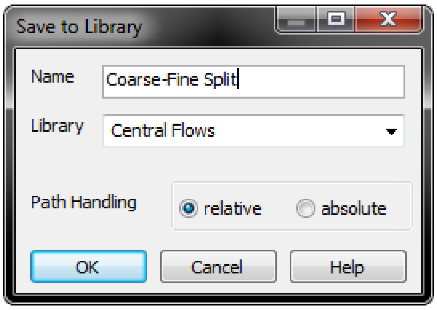
1. Select the modules that are needed for the new functionality. In the case of our example, select the modules in the blue rectangle. All the modules needed for the Coarse-Fine Split operation should look like the image on the right.



2. With the modules still selected, right-click on the label of one of the selected modules. A context-menu will appear. Click Save Selection to Library ... with the left mouse button.



3. A dialog should appear. This is where you can name the new database module. Here the name is Coarse-Fine Split. We are also saving it to the Central Flows Library and setting the Path Handling as relative. The difference between th Library options (Central Flows and User Flows) are explained in the section [Central Flows and Users Flows](). Path Handling is the how the database module should be managed internally. Relative implies that the relative path from the BEAMER binary to the database should be used. An absolute path implies that the direct path to the new database module should be used. Click OK to save this new database module to the Central Flows Library.

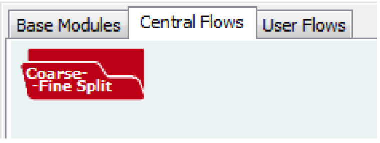

Save to Library

Name: Coarse-Fine Split

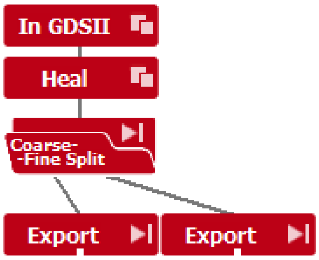
Library: Central Flows

Path Handling: ● relative ○ absolute

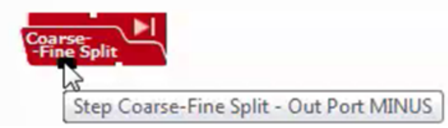
4. With the new database module saved, you should now find it in the Central Flows tab as shown to the right.



5. Here you can replace the original modules with this single database module by selecting and deleting the original modules and by "dragging-n-dropping" the new Coarse-Fine Split module to the workspace. Once you have hooked up the database module accordingly, the flow should function as you would expect: the coarse and fine features are separated with fine features being 1um or smaller in CD and coarse features being larger than 1um in CD.

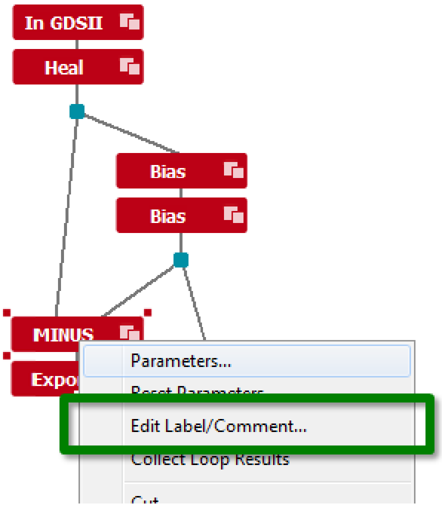


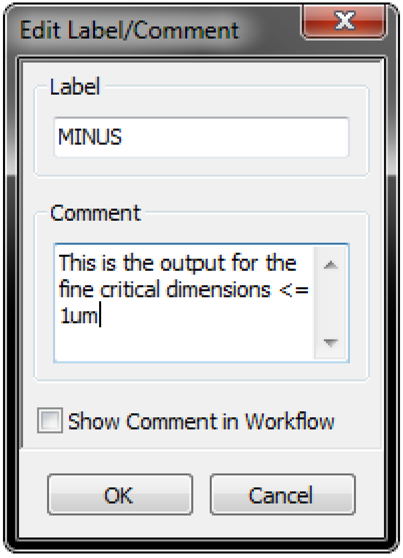
As an aside, after you drag-n-drop the database module into the workspace, you can hover the mouse over the ports. A pop-up will appear to indicate to what internal port the port of interest belongs. To the right, the out port on the left belongs to the MINUS module.



Step Coarse-Fine Split - Out Port MINUS

To makes this out port more meaningful than *Out Port Split,* before saving the base modules to the database, edit the comment field of the modules whose out ports will be exposed in the database module. For example, let's edit the MINUS module's comment field. To access the dialog on the right for the MINUS module, right-click the label of the module and left click **Edit Label/ Comment...** . You will then have access to the dialog to the bottom right.





Label

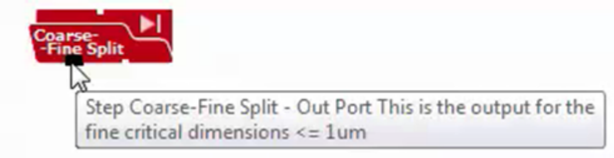
MINUS

Comment

This is the output for the fine critical dimensions <= 1um

□ Show Comment in Workflow

Click OK to save the comment. Now follow the same steps to save the modules as a Coarse-Fine Split database module and add the new module to the workspace. Hover over the left out port and you should see a more meaningful pop-up.



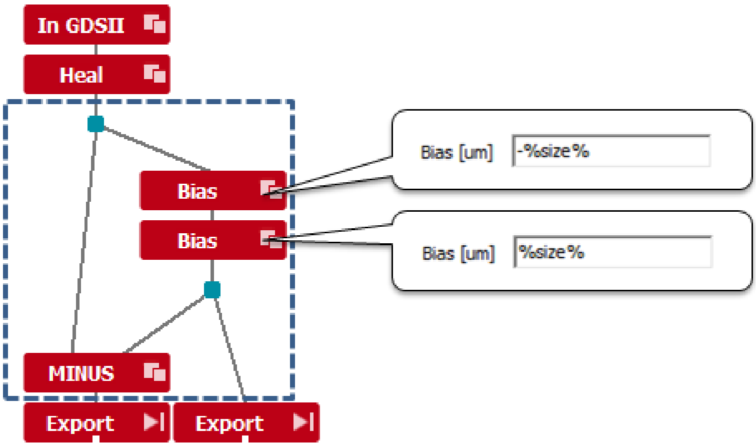
Step Coarse-Fine Split - Out Port This is the output for the fine critical dimensions <= 1um

Database modules are more powerful with variables. In the next section, variables and database modules are explained in detail.

6.2 Using Variables

Database modules can use variables. Variables in BEAMER are denoted by a text string surrounded by % symbols. For more information on variables in BEAMER, please refer to the section on Variables. In our previous example, we generated a Database module that implements a Coarse-Fine Split. However, the database module only works to separate coarse features that are larger than 1um from finer features that are 1um or smaller. Instead of always having a 1um Coarse-Fine Split operation, a variable in the Bias modules can allow us to dynamically change the separation value between coarse and fine features quickly and easily. Here's how it's done:

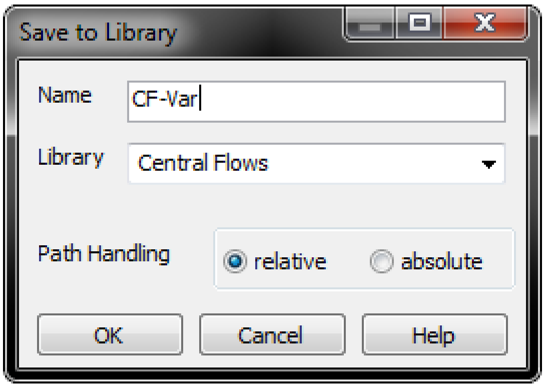
1. In the Bias modules, enter a variable of the same name as shown below. Notice how the first Bias has a negative sign in front of the variable while the second bias does not.



Bias [um] -%size%

Bias [um] %size%

2. Follow steps 1 through 5 in the previous section to save the set of modules as a Database module. Here we are saving the new module as **CF-Var**.

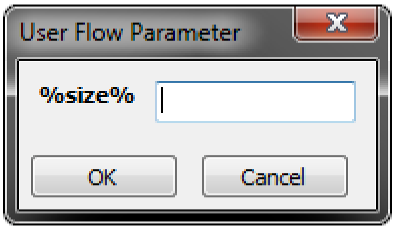


Name CF-Var

Library Central Flows

Path Handling ● relative ○ absolute

3. Now, when you insert this new variable into the workspace, a dialog will appear asking for the value of the %size% variable:



%size%

When a module within a database module contains a variable, a dialog box will open when the module is dragged into the workspace area like shown above. The user must then enter a value for the variable so that the module can properly execute. This offers a great flexibility when generating flows that are used frequently.

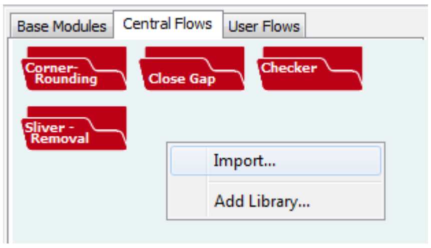
6.3 Central Flows and Users Flows

There are two locations for storing these customized modules, Central Flows and User Flows. The location does not change its behavior, but rather defines the availability to other users. A database module saved in the Central Flows library can be accessed by any user on the system, or on the network in cases where the system is configured as a server. Any module saved in the User Flows library is only available to the current user. If more than two User Flow libraries are created, you may need to use the navigation arrows to move between libraries.

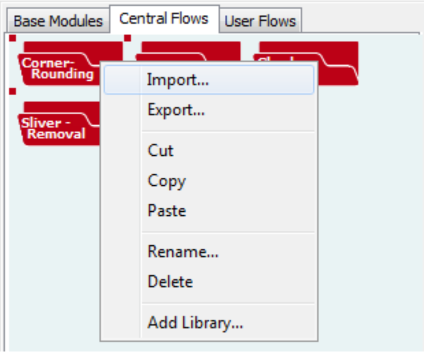
Central Flows

Save individual customized module settings or complete flows and makes them available to all users. There is only one Central Flows library and it cannot be deleted.

A right mouse click inside the Central Flows window pane opens a dialog where you can Import a .ftxt flow file. You can click **Add Library...** to create a new User Flows library tab.



Right clicking on a database module in the Central Flows window brings up a context menu that gives options for importing, exporting .ftxt flow files, cut, copy and paste to remove or duplicate the selected module, rename or delete the module or add a new User Flows library.



User Flows

By saving a flow into a User Flows database, only the current user of the system will have access to it. User Flows database modules can be added as necessary but the module is only available to the local user. To permit access to the module globally, save the flow into the Central Flows library. A right mouse click inside the User Flows window pane opens a dialog where you can Import a .ftxt flow file, rename, delete or add a library.

When you add a new library, a new database window pane is created where flows can be stored. You can give the new tab a unique name for managing your flows.

Creating a database module

When you have an individual module or complete process flow designed in the current Process Flow Design window, you can choose Library Save from the File menu and save the entire in the Process Flow Design window into your choice of database libraries. Optionally, if there is a specific module or flow you would like to save, first select it with the mouse and right click to bring up the context menu and choose Save Selection to Library.

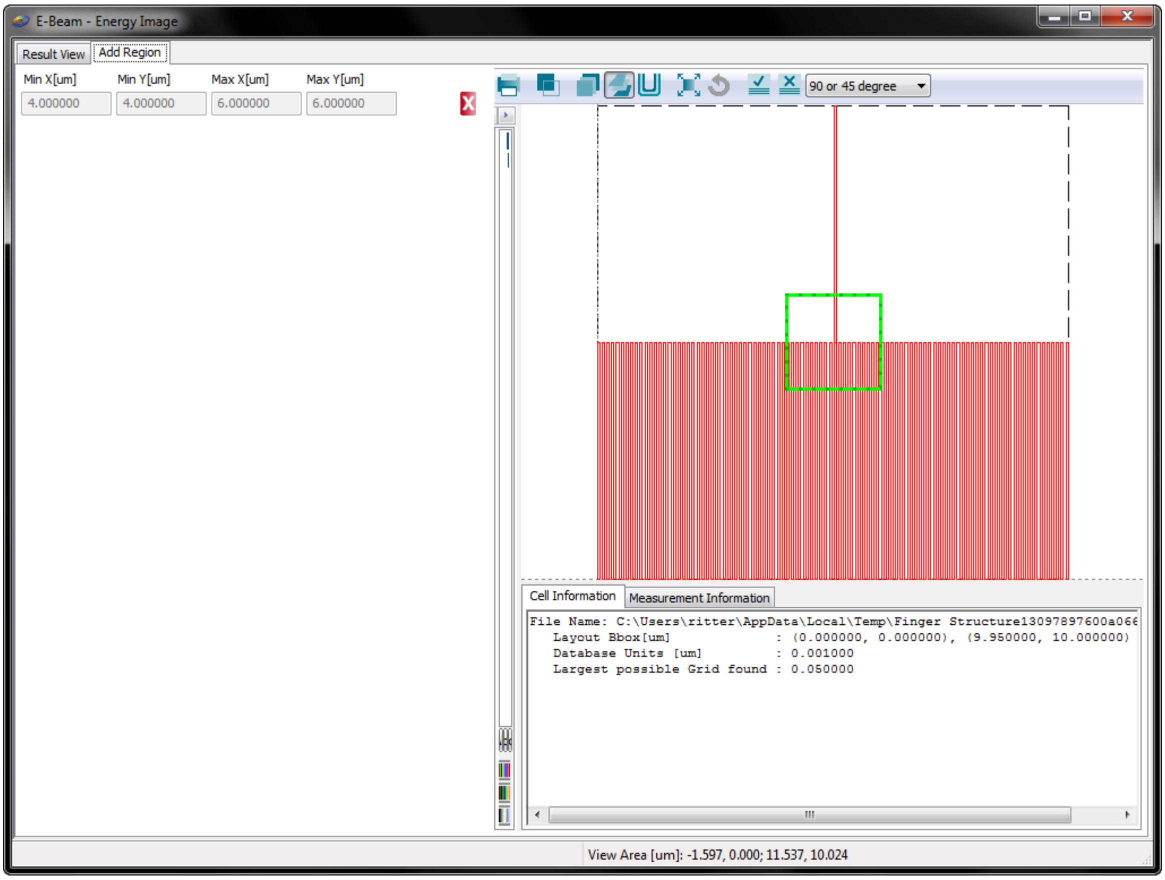
7 E-Beam Simulation

The Image Viewer provides enhanced viewing and evaluation capabilities for the E-Beam module. It combines a set of view options of the data to allow analysis of the generated data. Additionally, it can visualize simulation results of loops to compare the effects of changing parameters. Some extended e-beam simulation functions include:

- Intensity image simulation of multiple regions
- Extended viewing
  - 2D only, 1D only, 1D+2D views
  - 1D image at user defined cut-line
  - Multiple region selectable by drop-down
  - Easy simulation of additional regions
  - Matrix-view for loops
- Powerful Evaluation
  - Measure image intensity, slope, log slope in 1D view
  - Export data for external evaluation (e.g. Excel, MatLab)

7.1 Add Region

The Add Region tab defines the regions that are going to be simulated.



Min X[um]	Min Y[um]	Max X[um]	Max Y[um]
4.000000	4.000000	6.000000	6.000000

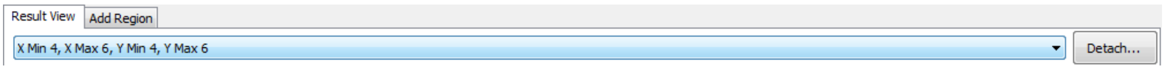
```
File Name: C:\Users\ritter\AppData\Local\Temp\Finger Structure13097897600a066
   Layout Bbox[um]            : (0.000000, 0.000000), (9.950000, 10.000000)
   Database Units [um]        : 0.001000
   Largest possible Grid found : 0.050000
```

View Area [um]: -1.597, 0.000; 11.537, 10.024

Regions can be inserted using the Shift key while dragging a frame with the mouse (start with first click and close with second click). The coordinates (to the left of the screen) can be adjusted afterwards if desired. The ☒ button can be used to delete the row of coordinates from the list.

7.2 Result View

The Result View tab visualizes the simulation results. You can choose which simulation region to analyze in the event multiple cases are defined.

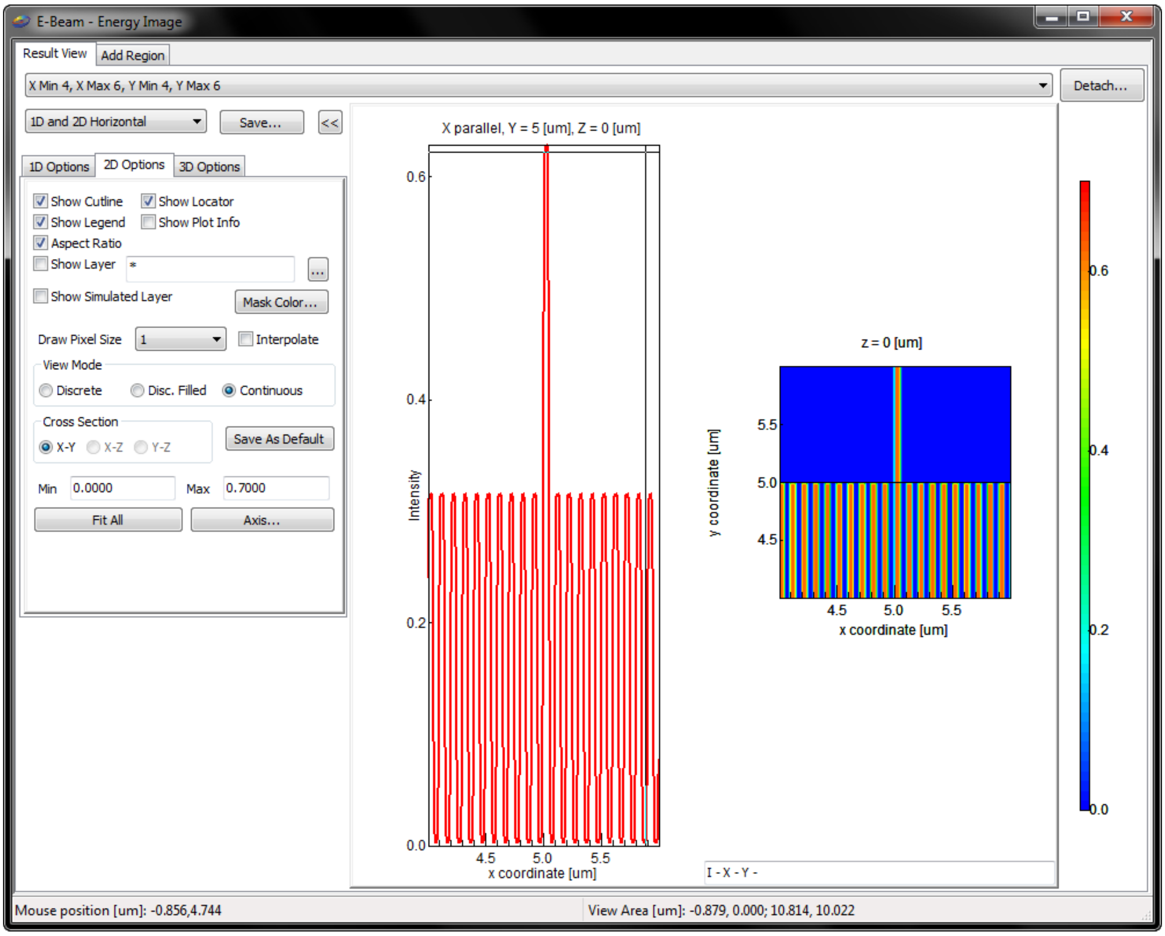


X Min 4, X Max 6, Y Min 4, Y Max 6

The *Detach...* button is to be used in case the plot from the analysis matrix is to be inspected individually.

7.3 1D and 2D View

The 1D and 2D View allows the definition of cutlines in the 2D data and displays the 1D data accordingly.



X Min 4, X Max 6, Y Min 4, Y Max 6

1D and 2D Horizontal

☑ Show Cutline ☑ Show Locator

☑ Show Legend □ Show Plot Info

☑ Aspect Ratio

□ Show Layer *

□ Show Simulated Layer

Draw Pixel Size 1 □ Interpolate

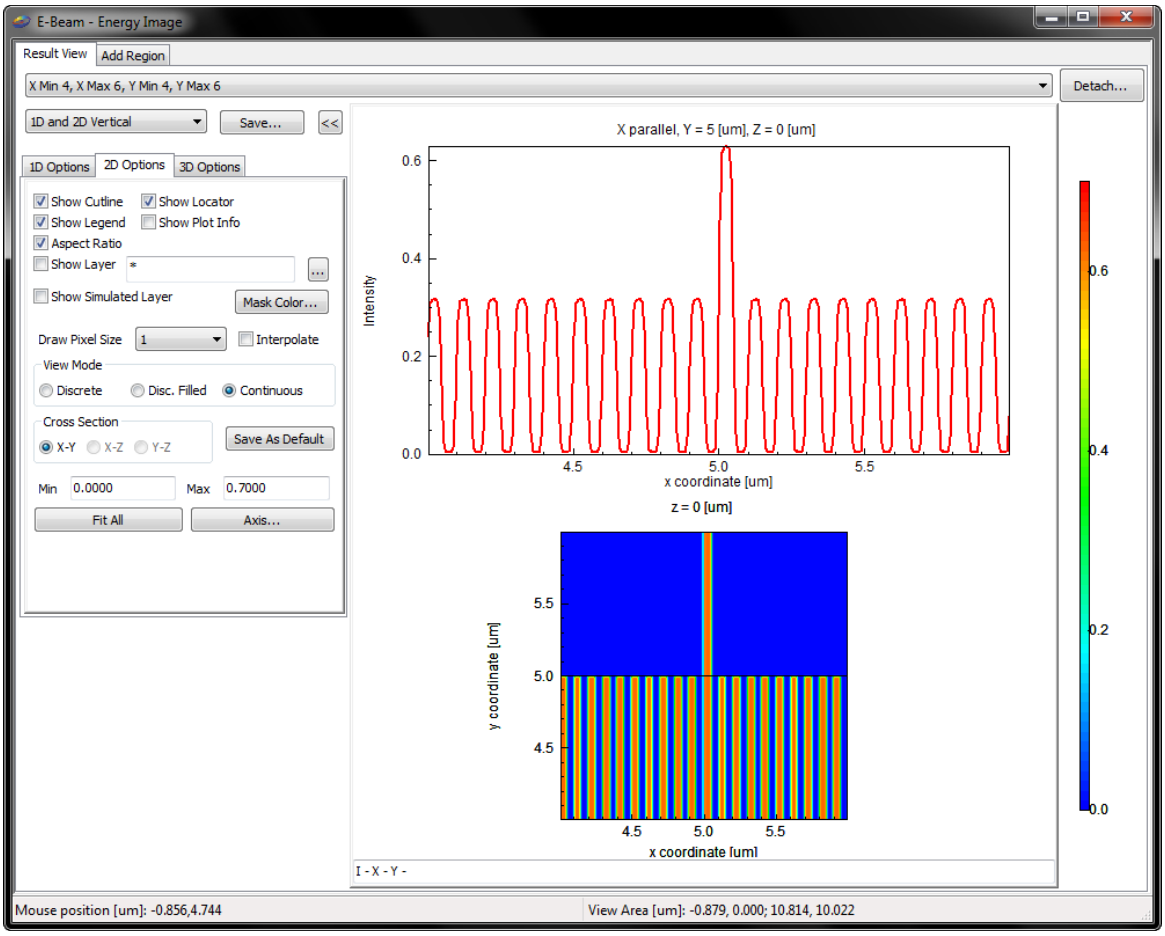
View Mode: ○ Discrete ○ Disc. Filled ● Continuous

Cross Section: ● X-Y ○ X-Z ○ Y-Z

Min 0.0000 Max 0.7000

Mouse position [um]: -0.856,4.744

View Area [um]: -0.879, 0.000; 10.814, 10.022



E-Beam - Energy Image

X Min 4, X Max 6, Y Min 4, Y Max 6

1D and 2D Vertical

☑ Show Cutline ☑ Show Locator
☑ Show Legend □ Show Plot Info
☑ Aspect Ratio
□ Show Layer *
□ Show Simulated Layer

Draw Pixel Size 1 □ Interpolate

View Mode: ○ Discrete ○ Disc. Filled ● Continuous

Cross Section: ● X-Y ○ X-Z ○ Y-Z

Min 0.0000 Max 0.7000

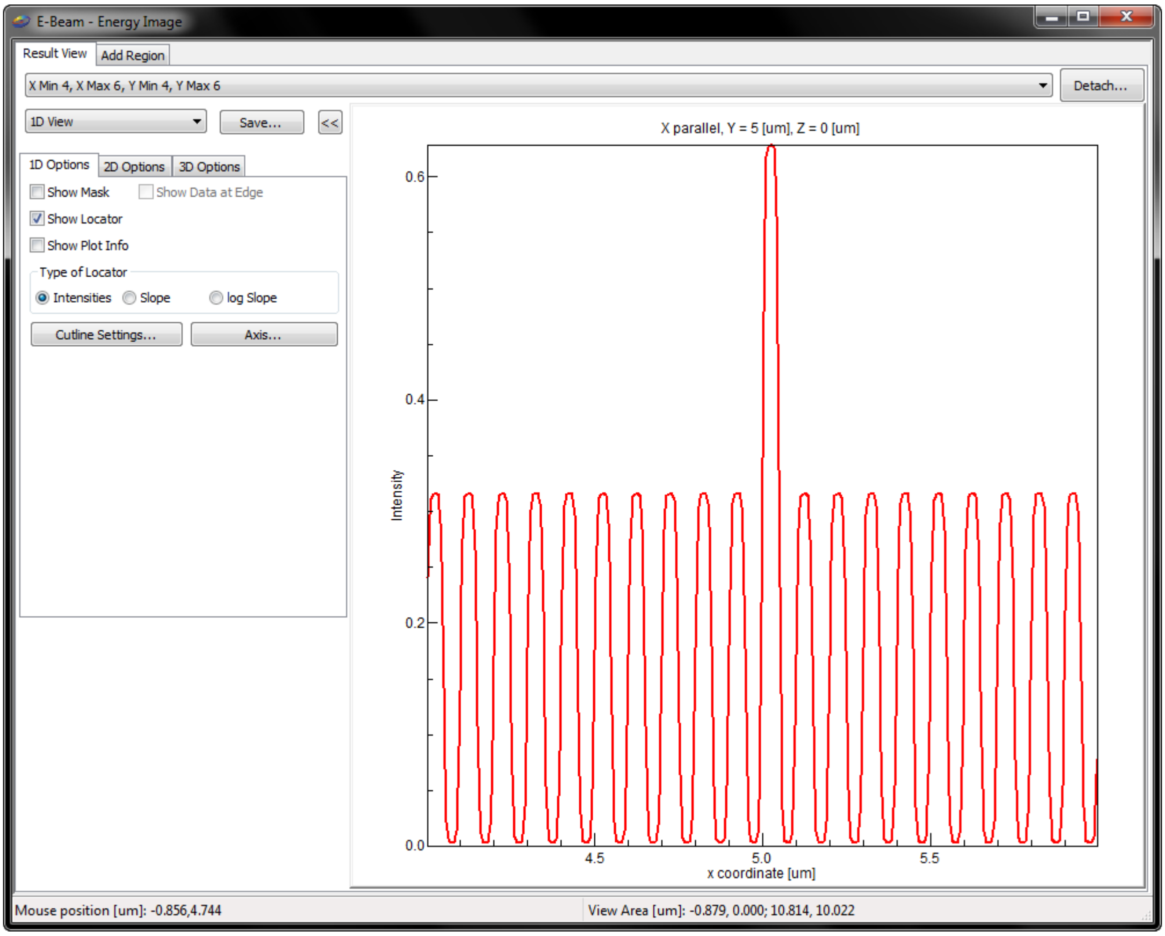
Mouse position [um]: -0.856,4.744

View Area [um]: -0.879, 0.000; 10.814, 10.022

The parameter settings can be looked up in the 1D and in the 2D sections below.

7.4 1D View

The 1D View displays the energy distribution below the cutline.



E-Beam - Energy Image

Result View | Add Region

X Min 4, X Max 6, Y Min 4, Y Max 6

1D View

1D Options | 2D Options | 3D Options

☐ Show Mask ☐ Show Data at Edge

☑ Show Locator

☐ Show Plot Info

Type of Locator

● Intensities ○ Slope ○ log Slope

Mouse position [um]: -0.856,4.744

View Area [um]: -0.879, 0.000; 10.814, 10.022

The view options available

Save

Saves the display to an image file or data file on the disk.

Show Mask

Toggles the display of the mask.

Show Locator

Toggles the display of the locator.

Show Plot Info

Toggles the display lower right supplemental plot information.

Type of Locator

Intensity

The crosshairs show the coordinate and intensity value on the axis. On the graph, it displays the coordinate where the intensity is present for the X-parallel crosshairs component and the intensity for the Y-parallel crosshairs component.

Slope and *log Slope*

The crosshairs display both the respective axis coordinates and the slope (or normalized log slope) for points of intersection on the graph.

Cutline Setting

Defines the position cut-line.

Axis

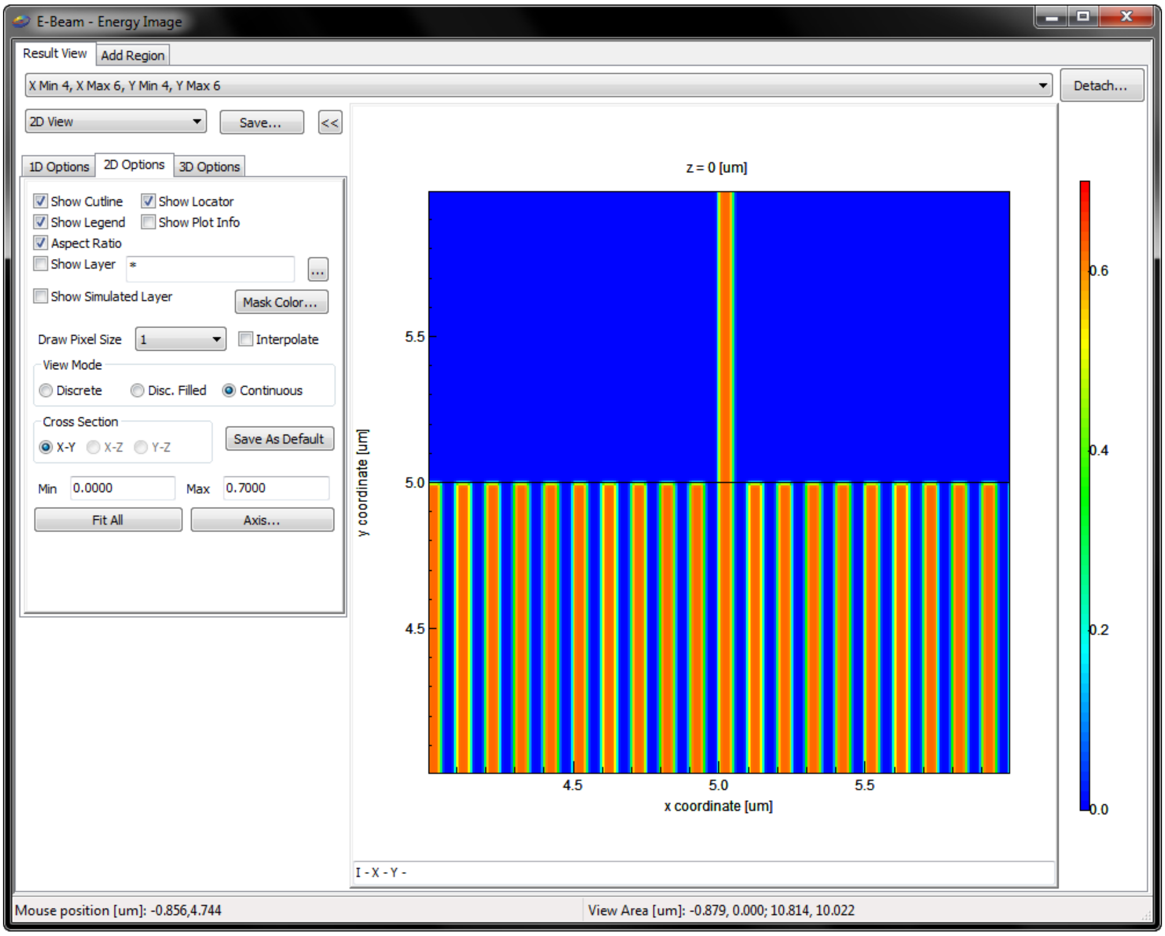
Sets the range of the axis that should be displayed.

Save 1D image

Stores the graph as image to the disk.

7.5 2D View

The 2D View displays the energy distribution as an X-Y cross-section view.



E-Beam - Energy Image

Result View | Add Region

X Min 4, X Max 6, Y Min 4, Y Max 6

2D View

1D Options | 2D Options | 3D Options

☑ Show Cutline ☑ Show Locator
☑ Show Legend □ Show Plot Info
☑ Aspect Ratio
□ Show Layer *
□ Show Simulated Layer

Draw Pixel Size 1 □ Interpolate

View Mode: ○ Discrete ○ Disc. Filled ● Continuous

Cross Section: ● X-Y ○ X-Z ○ Y-Z

Min 0.0000 Max 0.7000

I - X - Y -

Mouse position [um]: -0.856,4.744

View Area [um]: -0.879, 0.000; 10.814, 10.022

The initial display of the view will have strong pixelization. This is intentional to accelerate the drawing process. As you zoom into the areas the view will become more accurate.

Save

Saves the display to an image file or data file on the disk.

Show Cutline

Toggles the display of the cutline.

Show Locator

Toggles the display of the locator.

Show Legend

Toggles the display of the legend.

Show Plot Info
Toggles the display lower right supplemental plot information.

Aspect Ratio
Keeps the aspect ratio of the view.

Show Layer
Defines which layers of the mask should be displayed.

Show Simulated Layer
Toggles the display of the mask as an overlay view.

Mask color button
Toggles the display of the mask as an overlay view and defines the mask color.

Draw Pixel Size
Sets the size of an intensity pixel on the display. Lower values mean finer, but slower views.

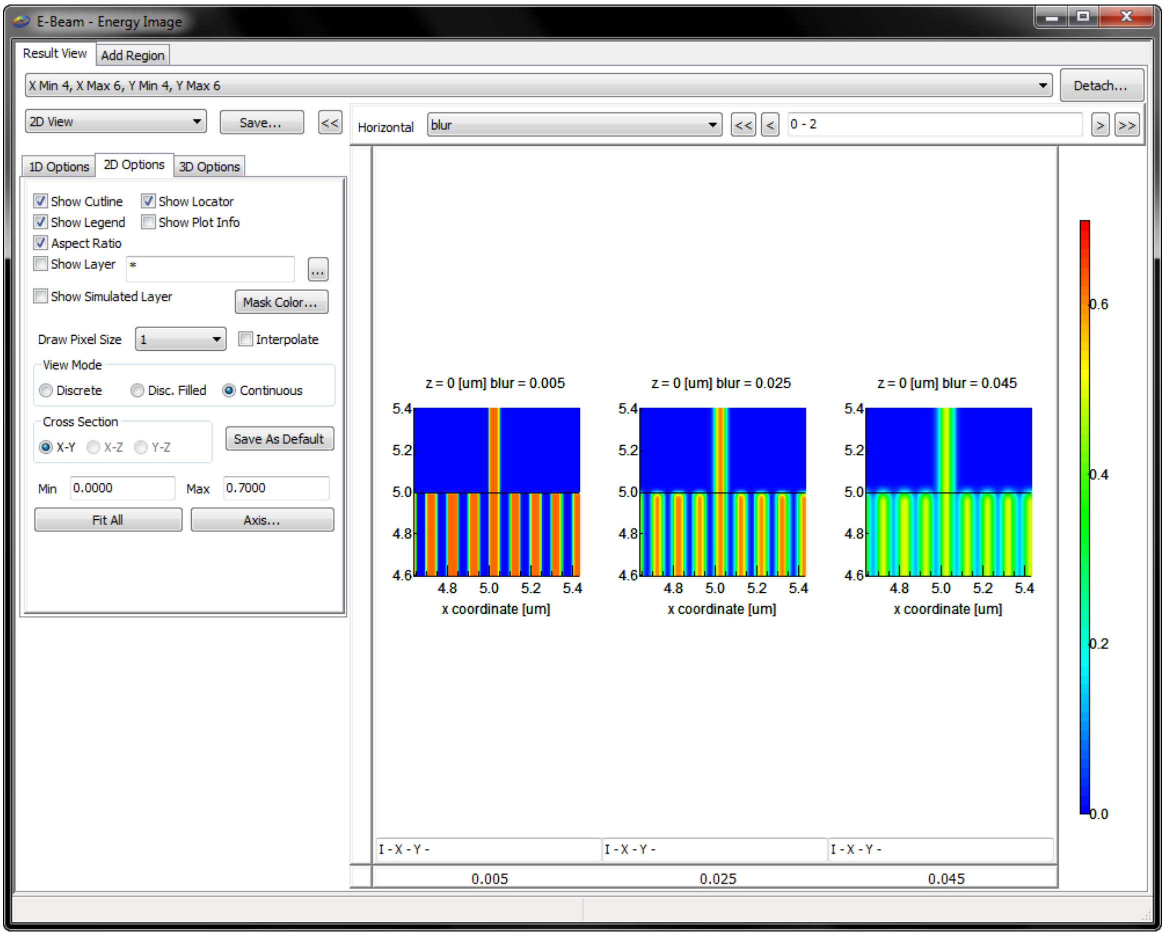
View Mode
Controls the visualization to be either discrete or continuous.

Fit all
Restores the default view.

Cross section
Sets the cut plane of the view.

7.6 Analysis

The Analysis visualization displays simulation runs in a matrix that looped over variable parameters. A control on top of the displayed results allows the sorting of the axis as desired. The matrix can take any desired form to be 1D, 2D or the 1D and 2D.



E-Beam - Energy Image

X Min 4, X Max 6, Y Min 4, Y Max 6

2D View

☑ Show Cutline ☑ Show Locator
☑ Show Legend □ Show Plot Info
☑ Aspect Ratio
□ Show Layer *
□ Show Simulated Layer

Draw Pixel Size 1 □ Interpolate

View Mode: ○ Discrete ○ Disc. Filled ● Continuous

Cross Section: ● X-Y ○ X-Z ○ Y-Z

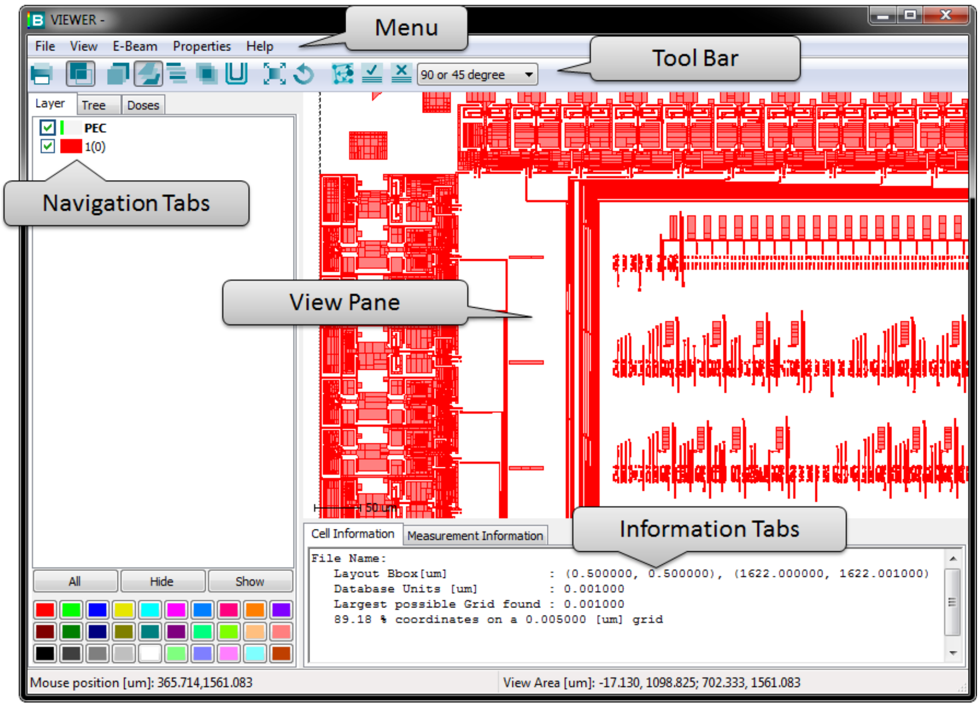
Min 0.0000 Max 0.7000

Horizontal blur 0 - 2

8 VIEWER

The VIEWER displays loaded layouts and intermediate results from a selected module or from multi-selected modules. To access the VIEWER click the of an executed module. If intermediate results are available, the VIEWER will open to show the result of the module's operation. Alternatively, right-click any executed module that contains intermediate results and click **View Layout** in the context menu to open the VIEWER.

Upon opening the VIEWER you will see the following interface. The main components are the Menu, Tool Bar, Navigation Tabs, View Pane and Information Tabs.

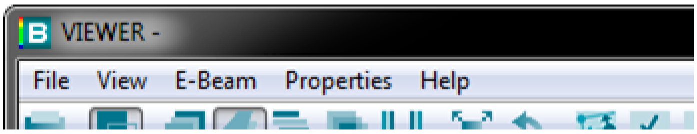


```
File Name:
   Layout Bbox[um]             : (0.500000, 0.500000), (1622.000000, 1622.001000)
   Database Units [um]         : 0.001000
   Largest possible Grid found : 0.001000
   89.18 % coordinates on a 0.005000 [um] grid
```

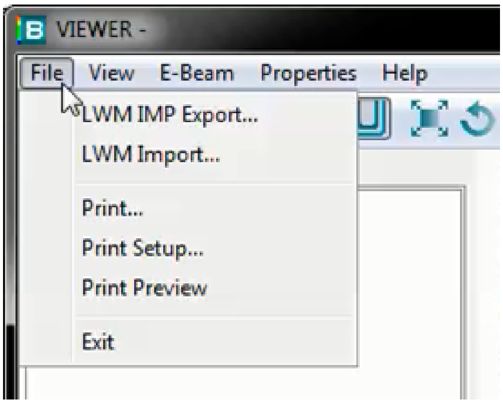
The VIEWER is also integrated into the dialogs of the Extract, Export (some machine formats), E-Beam and Metrology modules for region extraction, field control, region selection for simulation and line measurement, respectively. Please refer to these modules directly for more information on the VIEWER's specific use cases for each or contact support.

8.1 Menu

The menu bar gives access to the functionality of the VIEWER. The details of each menu item is explained below.



8.1.1 File



LWM IMP Export

The VIEWER can be used to introduce measurement lines into a loaded layout. These lines can be exported by this command into a separate file.

LWM Import

This entry allows the import of measurement lines.

Print

Prints the current view of VIEWER

Print Setup

Sets up the printer

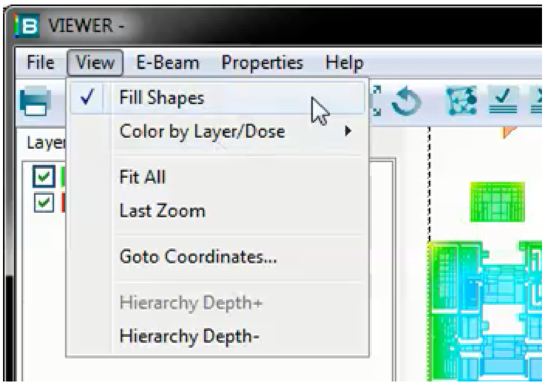
Print Preview

Preview of the print

Exit

Terminates the VIEWER

8.1.2 View



Fill Shapes

Displays the layout elements filled or framed

Color by Layer/Dose

Controls the color display by either Layer, Datatype or Dose

Fit All

Restores the view of the layout to show the entire layout in the given display window

Last Zoom

Restores the last zoom

Goto Coordinates

See Goto Coordinates

Hierarchy Depth +

Increases the hierarchy display depth

Hierarchy Depth -

Decreases the hierarchy display depth

8.1.2.1 Goto Coordinates

The Goto Coordinates dialog allows you to move the field of view to a specific point in the layout, by entering X and Y coordinates. The Width of the view can be specified. The height of the view is dependent on the size of the View Pane and is tied to the specified width.

Note that you can have the Goto Coordinates dialog open while navigating in the View Pane. The Goto Coordinates can be helpful when inspecting patterns with constants shifts on a larger scale.

Set Position

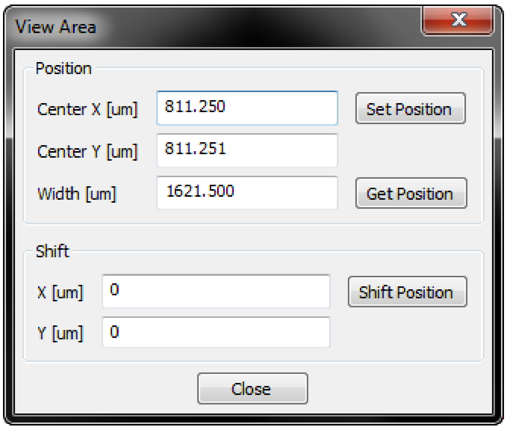
Moves the view to the currently defined coordinates

Get Position

Reads from the currently active view the coordinates

Shift Position

Moves the current view by the defined parameters



View Area

Position

Center X [um] 811.250

Center Y [um] 811.251

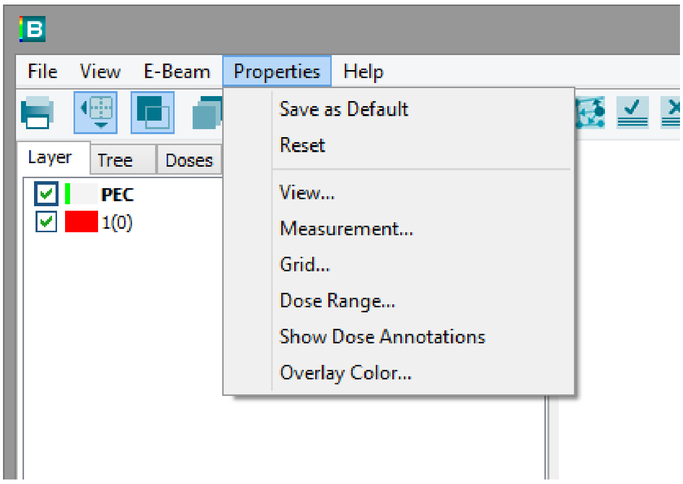
Width [um] 1621.500

Shift

X [um] 0

Y [um] 0

8.1.3 Properties



Save as Default

Saves the current VIEWER settings as default.

Reset

Resets the VIEWER settings to factory defaults.

View...

Opens the view properties dialog. See the View section below.

Measurement...

Opens the Measurement properties dialog. See the Measurement section below.

Grid...

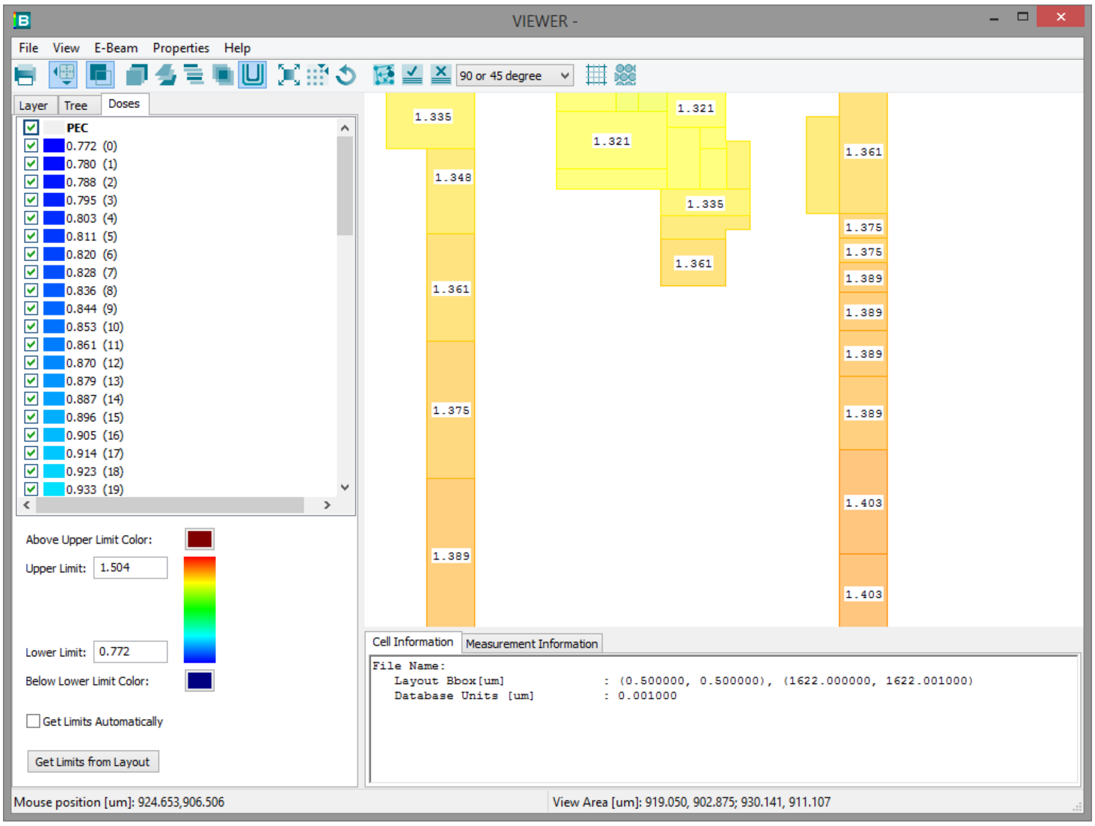
Opens the Grid properties dialog. See the Grid section below. Grid settings are also accessible via ▦ on the VIEWER toolbar.

Dose Range...

Opens the Dose Color properties dialog. See the Dose Color section below.

Show Dose Annotations

Displays assigned dose values (when in the *Color by Dose* view mode) within a shape when zoomed-in:



☑ PEC
☑ 0.772 (0)
☑ 0.780 (1)
☑ 0.788 (2)
☑ 0.795 (3)
☑ 0.803 (4)
☑ 0.811 (5)
☑ 0.820 (6)
☑ 0.828 (7)
☑ 0.836 (8)
☑ 0.844 (9)
☑ 0.853 (10)
☑ 0.861 (11)
☑ 0.870 (12)
☑ 0.879 (13)
☑ 0.887 (14)
☑ 0.896 (15)
☑ 0.905 (16)
☑ 0.914 (17)
☑ 0.923 (18)
☑ 0.933 (19)

Above Upper Limit Color:

Upper Limit: 1.504

Lower Limit: 0.772

Below Lower Limit Color:

□ Get Limits Automatically

```
File Name:
   Layout Bbox[um]             : (0.500000, 0.500000), (1622.000000, 1622.001000)
   Database Units [um]         : 0.001000
```

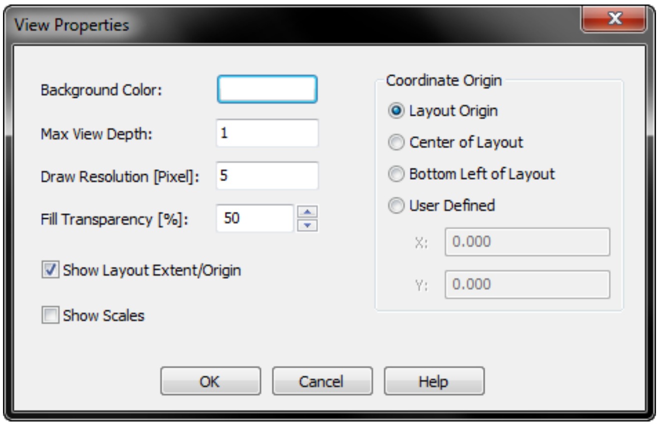
Mouse position [um]: 924.653,906.506

View Area [um]: 919.050, 902.875; 930.141, 911.107

Overlay Color...

Opens the Overlay Color properties dialog. See the Overlay Color section below.

8.1.3.1 View...



View Properties

Background Color:

Max View Depth: 1

Draw Resolution [Pixel]: 5

Fill Transparency [%]: 50

☑ Show Layout Extent/Origin

□ Show Scales

Coordinate Origin

● Layout Origin

○ Center of Layout

○ Bottom Left of Layout

○ User Defined

X: 0.000

Y: 0.000

Background Color
Sets the background color using a color palette.

Max View Depth
Sets the drawn depth of the layout.

Draw Resolution
Selects which layout elements are drawn. Should a layout element be smaller in size on the screen then this pixel value, then it is not drawn, to increase drawing speed.

Fill Transparency
Sets the transparency level for the filled view.

Show Layout Extent/Origin
Toggles the display of the layout extent and the origin.

Coordinate Origin
Selects the origin of the layout’s coordinate system.

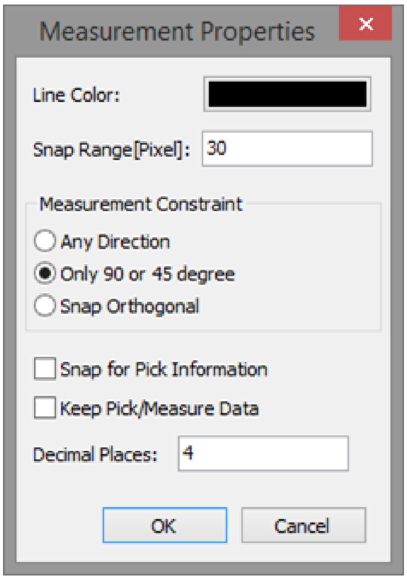
Layout Origin
Maintains original origin of the layout

Center of Layout
Places the origin of the layout coordinate system at the center of the layout.

Bottom Left of Layout
Places the origin at the lower left.

User Defined
Sets the coordinate origin to a user defined position.

8.1.3.2 Measurement...


Measurement Properties

Line Color: ■

Snap Range[Pixel]: 30

Measurement Constraint

○ Any Direction

● Only 90 or 45 degree

○ Snap Orthogonal

□ Snap for Pick Information

□ Keep Pick/Measure Data

Decimal Places: 4

Line Color

Sets the color of the measurement line.

Snap Range

Defines how close the end of the measurement line must be in order to snap to an existing border.

Measurement Constraint

Defines which angles are allowed for the measurement line

Any Direction

Any angle is allowed

Only 90 or 45 degree

Only angles that are a multiple of 45 degrees area allowed

Snap Orthogonal

Line is measuring orthogonal to the snapped order

Snap for Pick Information

When doing a pick, the measurement snaps to a border.

Keep Pick/Measurement Data

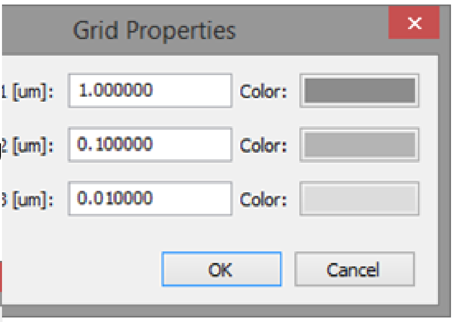
Keeps the measurement data saved during zooming.

8.1.3.3 Grid...

Grid settings are also accessible via ▦ on the VIEWER toolbar.

Show Grid 1 [um], Show Grid 2 [um], Show Grid 3 [um]

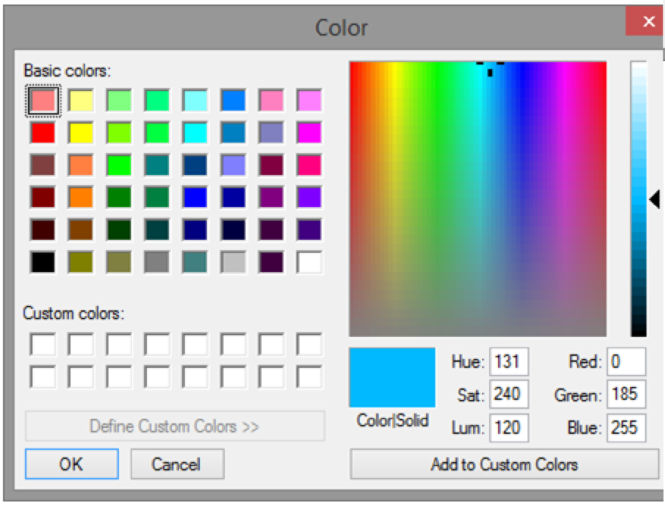
Each corresponding checkbox toggles and sets one of the three grids to be drawn over the layout. A grid is only drawn when a certain number of lines can be displayed. Each grid is assigned a different color. The corresponding grid colors can be changed by clicking the color and selecting a desired color from the palette shown below.


Grid Properties

1 [um]: 1.000000 Color:

2 [um]: 0.100000 Color:

3 [um]: 0.010000 Color:


Color

Basic colors:

Custom colors:

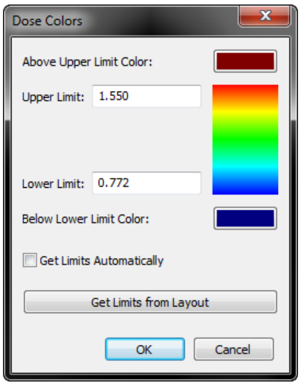
Hue: 131 Sat: 240 Lum: 120

Red: 0 Green: 185 Blue: 255

8.1.3.4 Dose Range...

The Dose Range dialog defines the color scale for drawing the dose information in the View Pane. A lower and upper limit can be set. The color scale is then adjusted accordingly. The corresponding upper limit color and lower limit color can be set by clicking the colored buttons and selecting a desired color from the palette.

Automatic limits can be obtained by enabling **Get Limits from Layout**.


Dose Colors

Above Upper Limit Color:

Upper Limit: 1.550

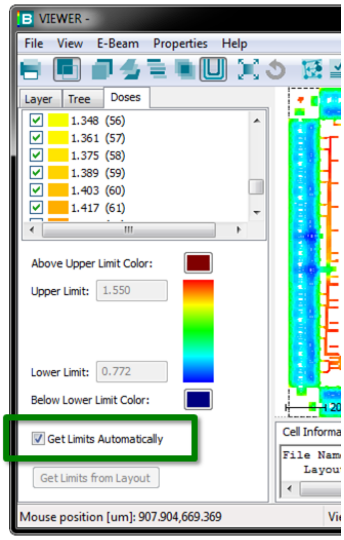
Lower Limit: 0.772

Below Lower Limit Color:

☐ Get Limits Automatically

To automatically obtain the upper and lower limits when clicking **Color-by-Dose**, enable the checkbox **Get Limits Automatically**. By default this is disabled. When enabled the Upper and Lower Limit text fields are disabled. To keep this feature enabled:

1. Click the checkbox labeled **Get Limits Automatically** to enable the feature via the Doses tab in the VIEWER (bottom left) or accessing the Dose Color dialog from the VIEWER Properties menu:


VIEWER -

	Dose
☑	1.348 (56)
☑	1.361 (57)
☑	1.375 (58)
☑	1.389 (59)
☑	1.403 (60)
☑	1.417 (61)

Above Upper Limit Color:

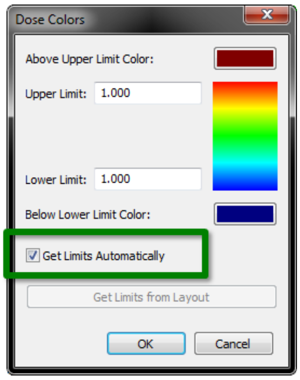
Upper Limit: 1.550

Lower Limit: 0.772

Below Lower Limit Color:

☑ Get Limits Automatically

Mouse position [um]: 907.904,669.369


Dose Colors

Above Upper Limit Color:

Upper Limit: 1.000

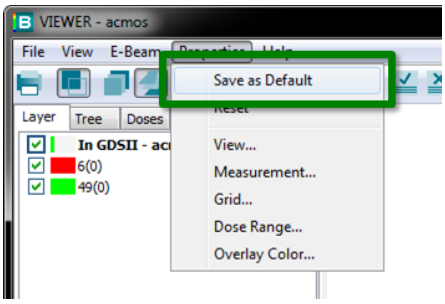
Lower Limit: 1.000

Below Lower Limit Color:

☑ Get Limits Automatically

2. Click **OK** to close the dialog

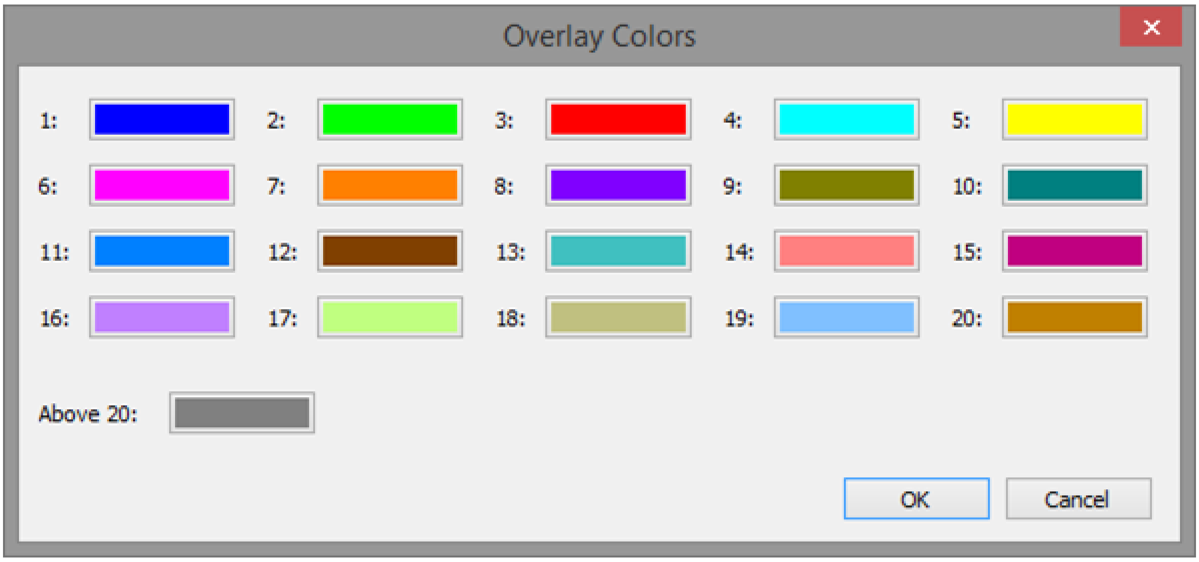
3. To save the setting, click **Save as Default** in the Properties menu



4. Close and re-open the VIEWER. Now whenever a corrected pattern is viewed and you click **Color-by-Dose**, the limits will automatically be set.

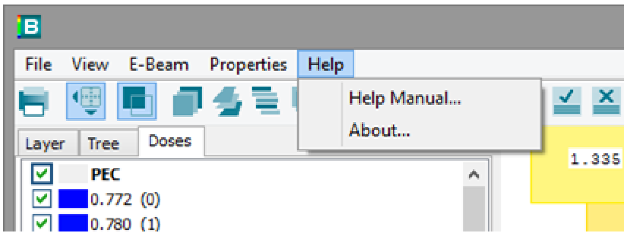
8.1.3.5 Overlay Color...

Defines the color used for drawing a specific numbers of overlaid elements. The number indicates the number of elements stacks at a pixel of the View Pane.



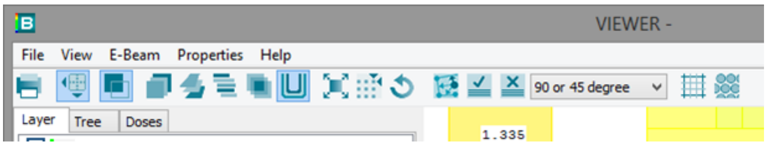
8.1.4 Help

The Help contains access to the Help Manual and About information regarding the VIEWER. The About dialog gives the version of VIEWER and the copyright statement.



8.2 Tool Bar

Commonly used VIEWER functions are quickly accessible from the tool bar. The name of a button of interest can be obtained by hovering the mouse over the button as shown to the right.



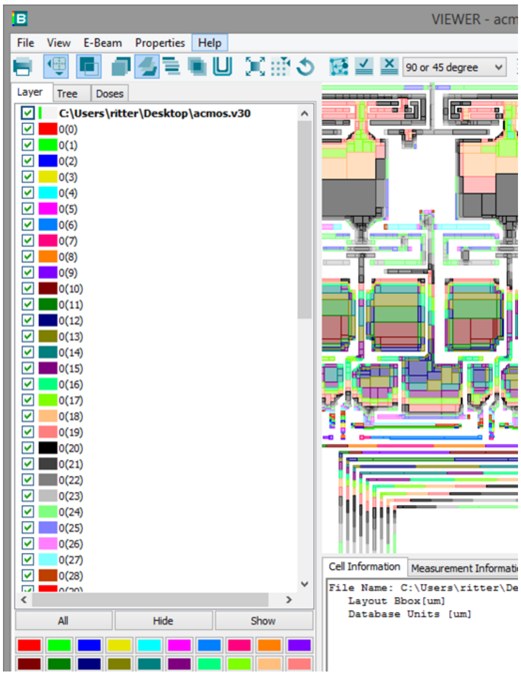
Tool Bar Component	Description
	Print
	Show Tree, Layer, Dose and Info
	Fill Shapes
	Color by Layout
	Color by Layer
	Color by Cell
	Color by Overlay
	Color by Dose
	Fit All
	Goto Coordinates
	Last Zoom
	Snap for Pick
✓	Keep Measurements
✕	Clear Measurements
90 or 45 degree ▾	Measurement line orientation behavior
	Grid Settings
	Shot Settings

8.3 Navigation Tabs

8.3.1 Layer

The Layer tab in the Navigation area allows to control which layers are shown in the View Pane. A check box toggles the display of individual layers. The name of a layer can be a true name, based on DXF files for example, or a number with parenthesis where the first number indicates the layer and the number in the parenthesis indicate the datatype.
The first line of the tree indicates the layout name along with the file location in a bold writing. The list afterward the available layers in the layout.

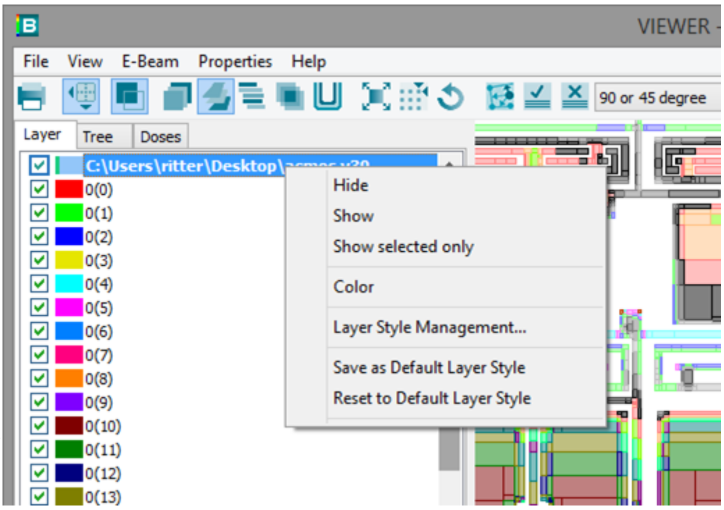
The color palette below the layer lists allows the assignment of another color to a selected layer.



```
File Name: C:\Users\ritter\De
   Layout Bbox[um]
   Database Units [um]
```

8.3.1.1 Layout Context Menu

The Layout Context Menu is available by right-clicking on the topmost node of the layer list for every layout displayed in the View Pane.


VIEWER -
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
C:\Users\ritter\Desktop\acmos.v30
0(0)
0(1)
0(2)
0(3)
0(4)
0(5)
0(6)
0(7)
0(8)
0(9)
0(10)
0(11)
0(12)
0(13)
Hide
Show
Show selected only
Color
Layer Style Management...
Save as Default Layer Style
Reset to Default Layer Style

Hide
Hides this layout from the View Pane

Show
Shows this layout in the View Pane

Show selected only
Shows only the layout in the View Pane

Color
Sets the color of this layout in the View Pane

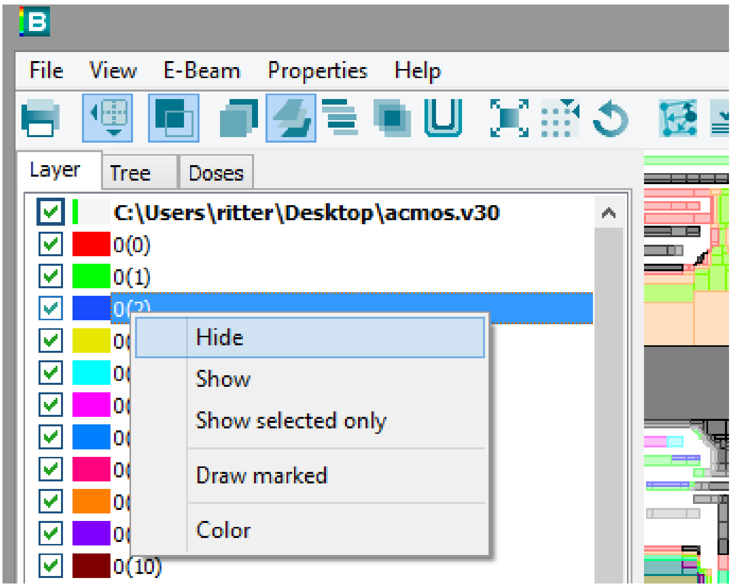
Layer Style management
Allows the save & load of specific color templates for a layout/layer setting

Save as Default Layer Style
Saves the current color map as Default

Reset to default Layer Style
Resets to the Default color style

8.3.1.2 Layer Context Menu

The Layer Context Menu is available by right-clicking on any layer in the layer list.



Hide
: Hides this layer from the View Pane

Show
: Shows this layer in the View Pane

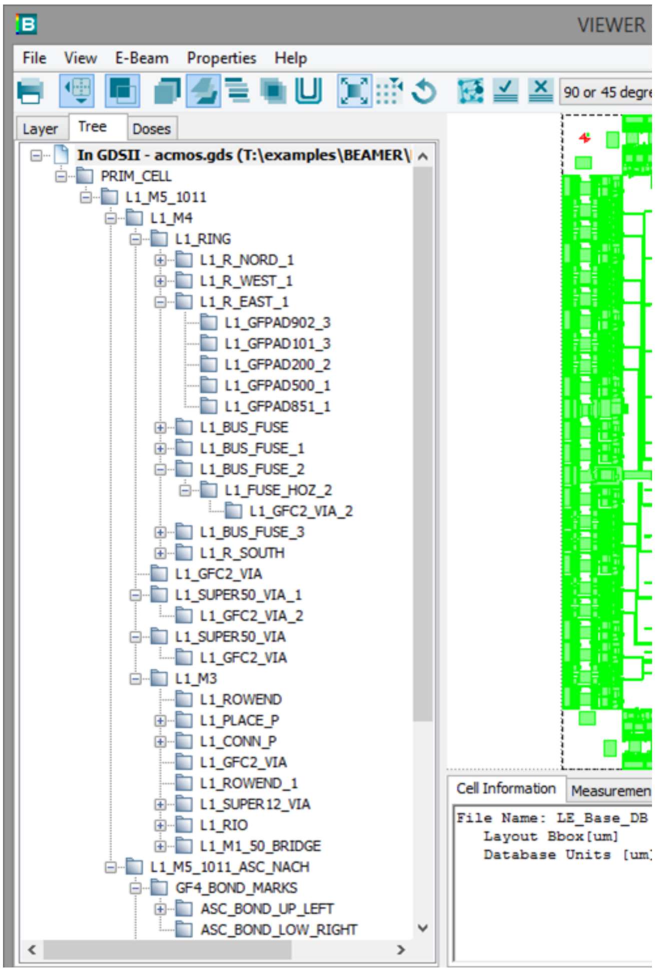
Show selected only
: Shows only the layer in the View Pane

Draw Marked
: Draw Marked prints a small cross on any element in this layer. Assists in finding very small elements in the layer since they might not be plotted otherwise unless zoomed in strongly.

Color
: Color sets the color of this layer in the View Pane

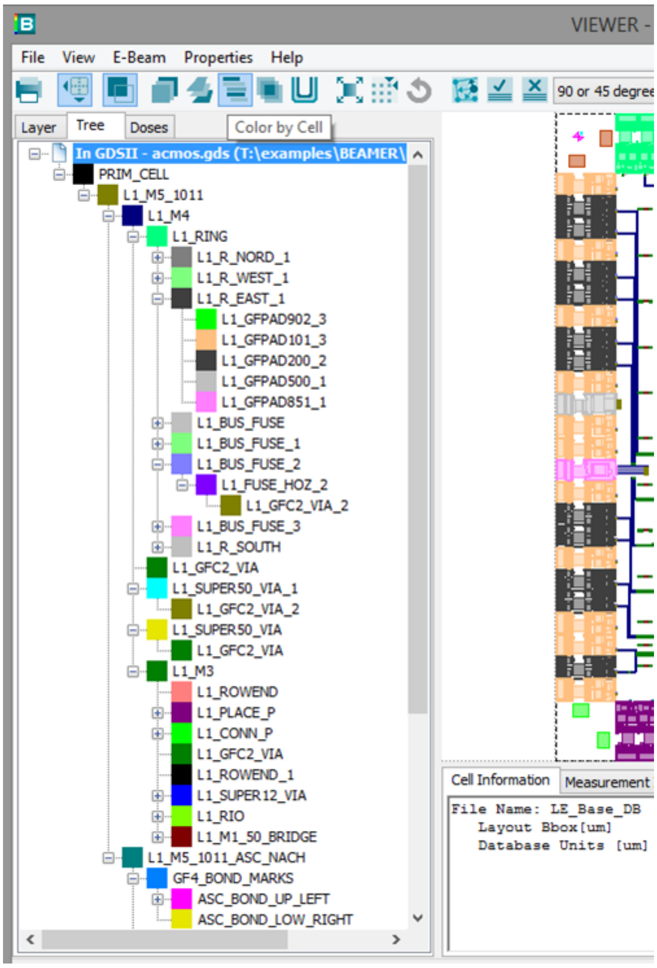
8.3.2 Tree

The Tree tab shows the pattern hierarchy. The small plus signs in front of a cell indicate a further depth to the hierarchy. To the bottom left, Color-by-Layer is selected. All cell names are in the tree have a folder icon. To the bottom right, Color-by-Cell is selected. Notice that the corresponding cell colors in the layout correspond to the colored squares (which were folder icons with Color-by-Layer) next to the cell name in the tree.



```
In GDSII - acmos.gds (T:\examples\BEAMER\
  PRIM_CELL
    L1_M5_1011
      L1_M4
        L1_RING
          L1_R_NORD_1
          L1_R_WEST_1
          L1_R_EAST_1
            L1_GFPAD902_3
            L1_GFPAD101_3
            L1_GFPAD200_2
            L1_GFPAD500_1
            L1_GFPAD851_1
          L1_BUS_FUSE
          L1_BUS_FUSE_1
          L1_BUS_FUSE_2
            L1_FUSE_HOZ_2
              L1_GFC2_VIA_2
          L1_BUS_FUSE_3
          L1_R_SOUTH
        L1_GFC2_VIA
        L1_SUPER50_VIA_1
          L1_GFC2_VIA_2
        L1_SUPER50_VIA
          L1_GFC2_VIA
        L1_M3
          L1_ROWEND
          L1_PLACE_P
          L1_CONN_P
          L1_GFC2_VIA
          L1_ROWEND_1
          L1_SUPER12_VIA
          L1_RIO
          L1_M1_50_BRIDGE
      L1_M5_1011_ASC_NACH
        GF4_BOND_MARKS
          ASC_BOND_UP_LEFT
          ASC_BOND_LOW_RIGHT
```

```
File Name: LE_Base_DB
   Layout Bbox[um]
   Database Units [um]
```



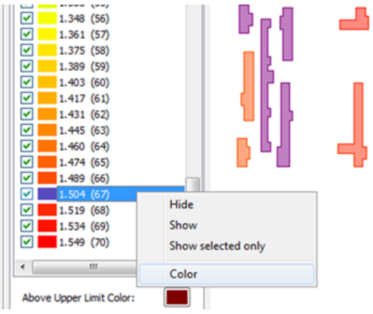
In GDSII - acmos.gds (T:\examples\BEAMER\
PRIM_CELL
L1_M5_1011
L1_M4
L1_RING
L1_R_NORD_1
L1_R_WEST_1
L1_R_EAST_1
L1_GFPAD902_3
L1_GFPAD101_3
L1_GFPAD200_2
L1_GFPAD500_1
L1_GFPAD851_1
L1_BUS_FUSE
L1_BUS_FUSE_1
L1_BUS_FUSE_2
L1_FUSE_HOZ_2
L1_GFC2_VIA_2
L1_BUS_FUSE_3
L1_R_SOUTH
L1_GFC2_VIA
L1_SUPER50_VIA_1
L1_GFC2_VIA_2
L1_SUPER50_VIA
L1_GFC2_VIA
L1_M3
L1_ROWEND
L1_PLACE_P
L1_CONN_P
L1_GFC2_VIA
L1_ROWEND_1
L1_SUPER12_VIA
L1_RIO
L1_M1_50_BRIDGE
L1_M5_1011_ASC_NACH
GF4_BOND_MARKS
ASC_BOND_UP_LEFT
ASC_BOND_LOW_RIGHT

```
File Name: LE_Base_DB
   Layout Bbox[um]
   Database Units [um]
```

8.3.3 Doses

The Doses tab allows the toggling of specific dose classes to be displayed in the View Pane. Like the Layer tab, you see first the dose value and in parenthesis, the dose class.

It is possible to apply custom coloring to a certain dose. This can be accomplished by right-clicking on a dose and clicking Color in the context menu as shown to the right

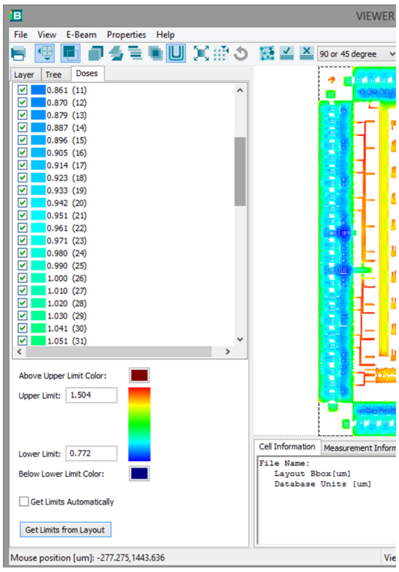


☑ 1.348 (56)
☑ 1.361 (57)
☑ 1.375 (58)
☑ 1.389 (59)
☑ 1.403 (60)
☑ 1.417 (61)
☑ 1.431 (62)
☑ 1.445 (63)
☑ 1.460 (64)
☑ 1.474 (65)
☑ 1.489 (66)
☑ 1.504 (67)
☑ 1.519 (68)
☑ 1.534 (69)
☑ 1.549 (70)

Above Upper Limit Color:

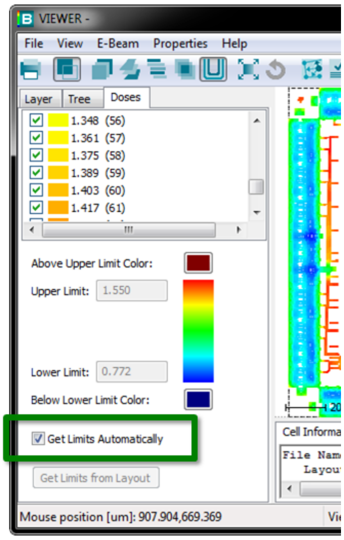
The bottom panel (highlighted in green to the right) is the same as the Dose Color dialog. This panel defines the color scale for drawing the dose information in the View Pane. A lower and upper limit can be set. The color scale is then adjusted accordingly. The corresponding upper limit color and lower limit color can be set by clicking the colored buttons and selecting a desired color from the palette.

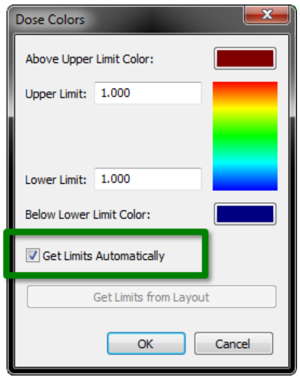
Automatic limits can be obtained by enabling **Get Limits from Layout**.



To automatically obtain the upper and lower limits when clicking **Color-by-Dose** , enable the checkbox **Get Limits Automatically**. By default this is disabled. When enabled the Upper and Lower Limit text fields are disabled. To keep this feature enabled:

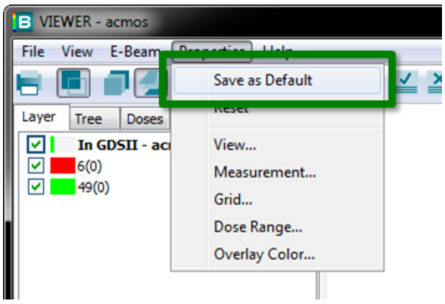
1. Click the checkbox labeled **Get Limits Automatically** to enable the feature via the Doses tab in the VIEWER (bottom left) or accessing the Dose Color dialog from the VIEWER Properties menu:





2. Click **OK** to close the dialog

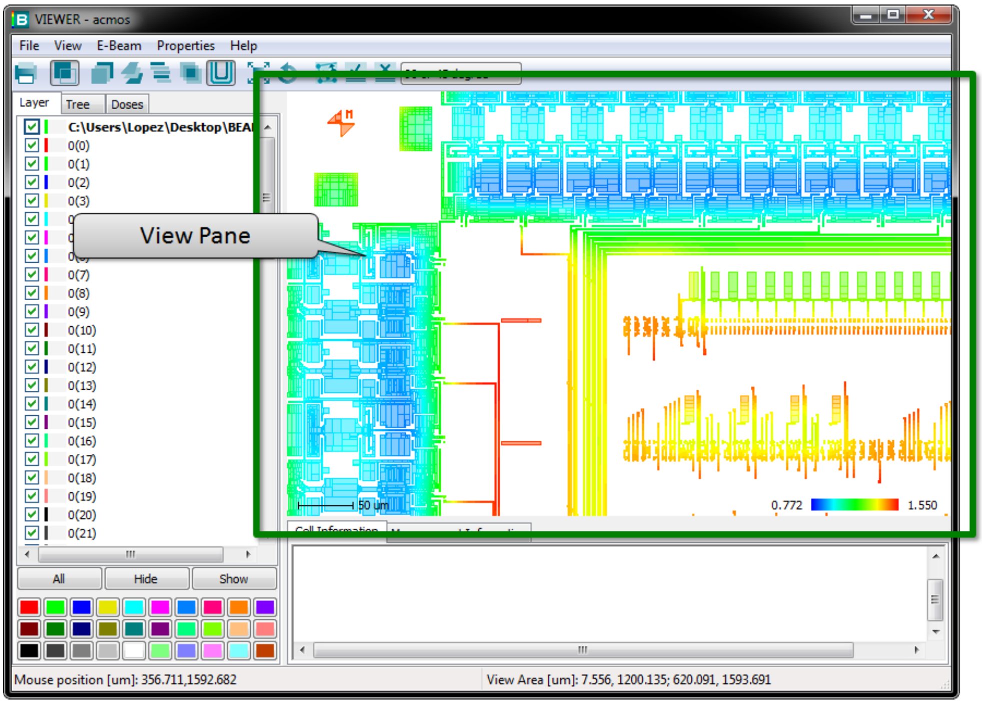
3. To save the setting, click **Save as Default** in the Properties menu



4. Close and re-open the VIEWER. Now whenever a corrected pattern is viewed and you click **Color-by-Dose**, the limits will automatically be set.

8.4 View Pane

The View Pane of the VIEWER displays the loaded layout.



Mouse position [um]: 356.711,1592.682

View Area [um]: 7.556, 1200.135; 620.091, 1593.691

The VIEWER window allows you to browse through loaded layouts with ease:

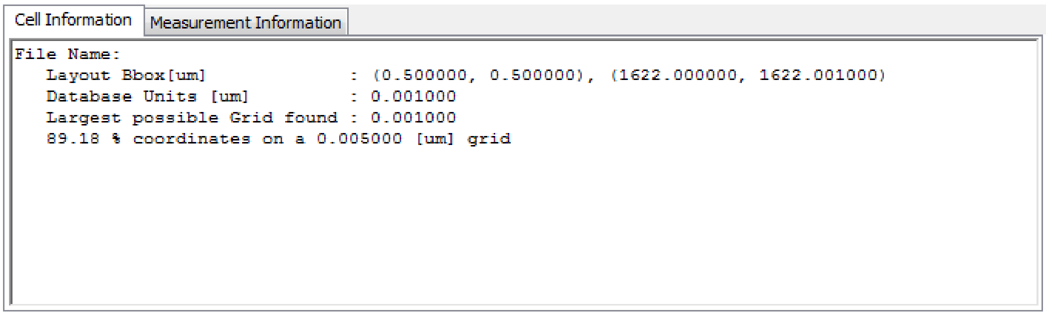
Action	Mouse/Keyboard Description
Zoom-In	• Left-click and drag the mouse from top-left to bottom-right on the viewable layout or • Turn the mouse wheel to the front
Zoom-Out	• Left-click and drag the mouse from bottom-right to top-left on the viewable layout • Turn the mouse wheel to the back
Centering	Left mouse click at a position to center the layout at that position
Panning	Press and hold the middle mouse button and move the mouse around. Releasing that button will then redraw the layout. Additionally the arrow keys can be used to navigate through the layout.
Measure	Right-click near one edge and right-click near another edge. A measurement will automatically be taken.
Snap to Corner or Edge for Pick Information	CTRL + double right-click near an edge or corner
Pick Information for a shape	Double right-click on a shape
Go to Coordinates	• CTRL + G • Also see Go to Coordinates

The VIEWER can also be found in the Extract, Export, E-Beam and Metrology modules. The same operations from above are available with some additional features as listed below:

Mouse and Keyboard Combination	Left	Middle	Right
Click	Pan to center view to click position	-	Measure start Measure end
Drag	• Zoom-in (drag a zoom box from top-left to bottom-right) • Zoom-out (drag a zoom box from bottom-right to top-left)	Pan	Measure start
Double Click	Pan to center view to click position	-	Pick corner or edge
Shift + Click	• Region start (Extract, Export, E-Beam) • Region end (Extract, Export, E-Beam)	-	Metrology line start Metrology line end
Shift + Drag	Region start	-	Metrology line start
Ctrl + Click	-	-	-
Ctrl + Drag	Pan	-	-

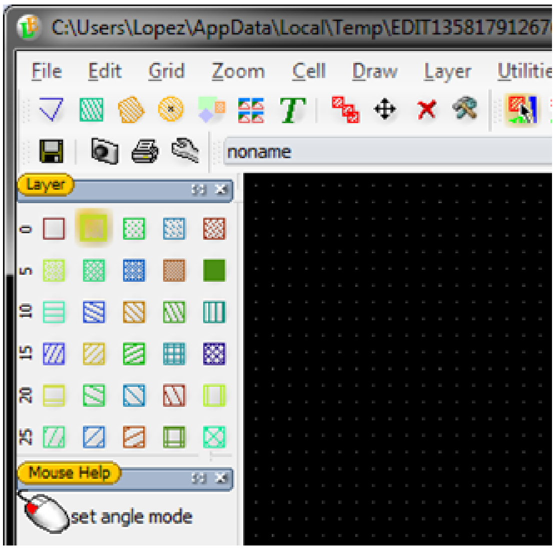
8.5 Information Tabs

The information area shows facts about the layout being viewed: area, layers, coverage or measurement information (i.e. positions of picks and distances between picks).



```
File Name:
   Layout Bbox[um]            : (0.500000, 0.500000), (1622.000000, 1622.001000)
   Database Units [um]        : 0.001000
   Largest possible Grid found : 0.001000
   89.18 % coordinates on a 0.005000 [um] grid
```

9 LayoutEditor


C:\Users\Lopez\AppData\Local\Temp\EDIT13581791267
File Edit Grid Zoom Cell Draw Layer Utilitie
noname
Layer
0
5
10
15
20
25
Mouse Help
set angle mode

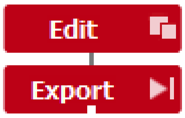
LayoutEditor, from Juspertor UG in Germany, is a fully functional layout editing program with sophisticated functions that is included in BEAMER via the Edit module.

It has a schematic capturing function where one can place a device in the schematic window and simply move the mouse to the layout window to get the corresponding device design. LayoutEditor is also equipped with a powerful macro functionality that allows for the programming of pattern generation. A full online manual and wiki is available at http://layouteditor.net/.

Adding the Edit module to a flow will open the LayoutEditor when the flow is executed which results in one of two operations:

1. It will load a layout from the previous module in the flow or
2. When there is no previous module, a blank layout will open, allowing for the creation of a new design.

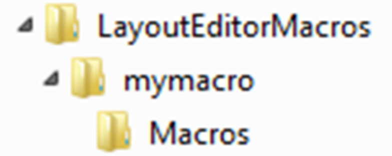
If a new pattern is generated, it is recommended that the pattern be saved immediately using the Export module. Be sure to immediately run the Export module to save pattern is written to file.



9.1 Configuring LayoutEditor to Use Macros

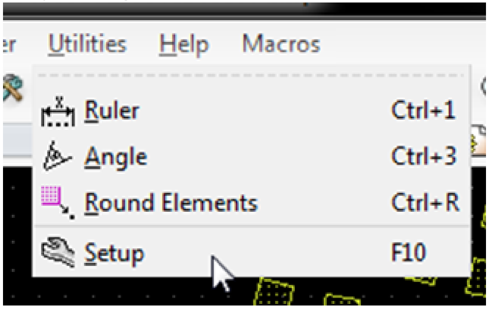
LayoutEditor has a powerful macro capability that can facilitate quick pattern generation with the use of hierarchy. To use macros in LayoutEditor, the program must be configured properly.

1. Create a directory for storing the LayoutEditor macros on your hard disk like below:

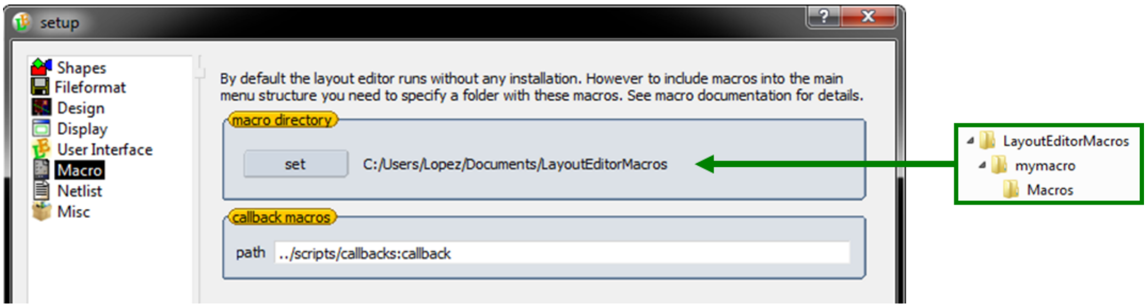


   a. The topmost directory, **LayoutEditorMacros**, can be any name.
   b. The subdirectory **mymacro** is mandatory.
   c. The directory under mymacro, **Macros**, will be read by LayoutEditor and placed in the LayoutEditor program menu.
   d. Move the script into the **Macros** directory.

2. In LayoutEditor, go to Setup by hitting **F10** or via the Utilities menu.



3. Under the Macro menu, set the macro directory and click **OK**.

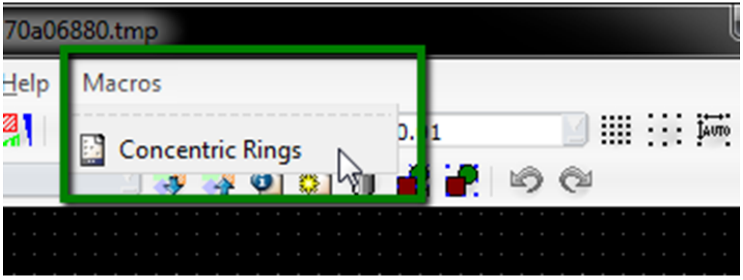


By default the layout editor runs without any installation. However to include macros into the main menu structure you need to specify a folder with these macros. See macro documentation for details.

set C:/Users/Lopez/Documents/LayoutEditorMacros

path ../scripts/callbacks:callback

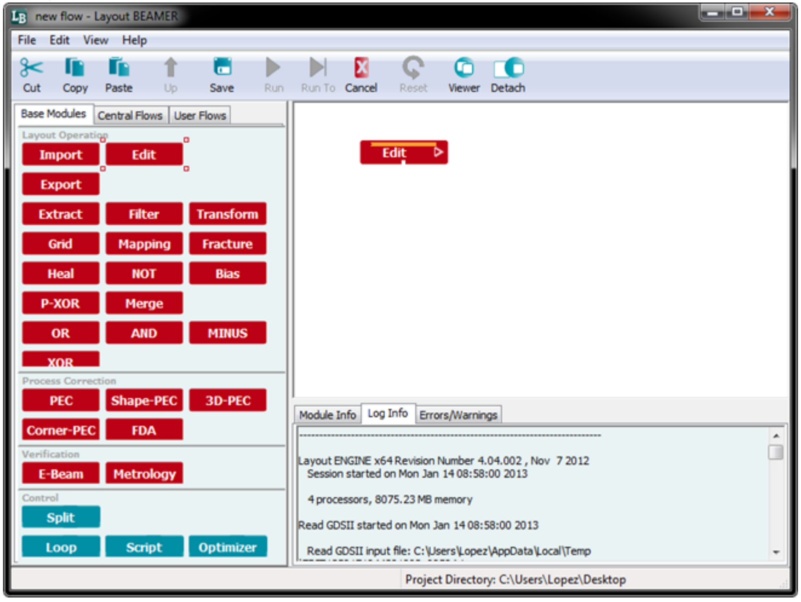
4. Exit out of LayoutEditor and reset the Edit module to run LayoutEditor again. You should now find Macros on the program menu. When you click on Macros, you should see the macro name.



9.2 Executing a Macro

The following steps show you how to execute a macro in LayoutEditor from BEAMER.

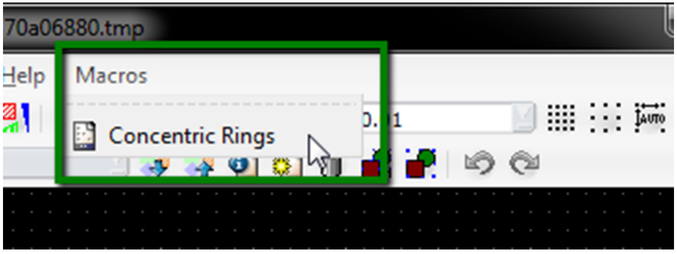
1. In BEAMER, drag the Edit module into the work space and run the module



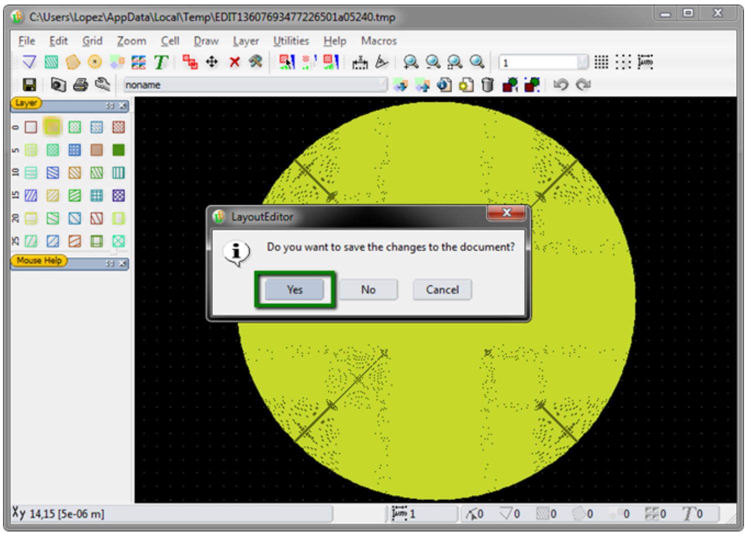
```
Layout ENGINE x64 Revision Number 4.04.002 , Nov  7 2012
  Session started on Mon Jan 14 08:58:00 2013

  4 processors, 8075.23 MB memory

Read GDSII started on Mon Jan 14 08:58:00 2013

  Read GDSII input file: C:\Users\Lopez\AppData\Local\Temp
```

Project Directory: C:\Users\Lopez\Desktop

2. In LayoutEditor, we need to run the macro. When LayoutEditor is configured correctly, the macro is readily available under the program menu. Just select the macro you want to run and it will execute. In this example, the user has setup a macro called **Concentric Rings**.



3. When the macro is finished, exit LayoutEditor and click **Yes** to return to BEAMER.



C:\Users\Lopez\AppData\Local\Temp\EDIT136076934772265O1a05240.tmp

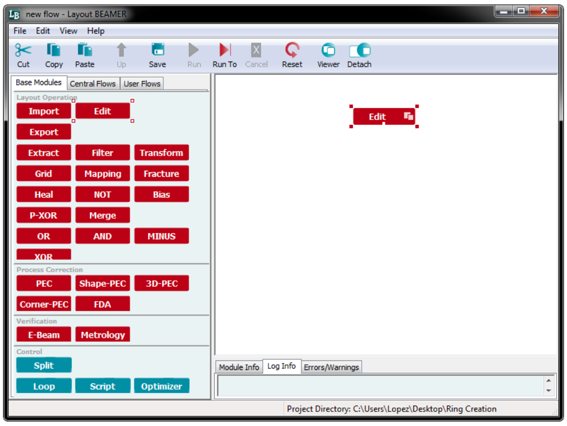
LayoutEditor

Do you want to save the changes to the document?

9.3 Saving a Pattern from LayoutEditor

To save a pattern from LayoutEditor:

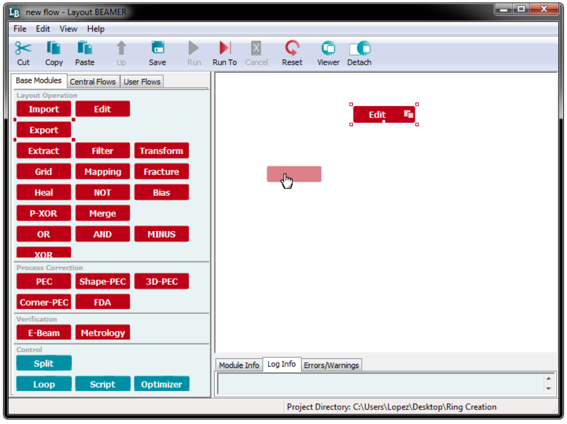
1. Exit LayoutEditor and click **Yes** to return to BEAMER.



new flow - Layout BEAMER

Project Directory: C:\Users\Lopez\Desktop\Ring Creation

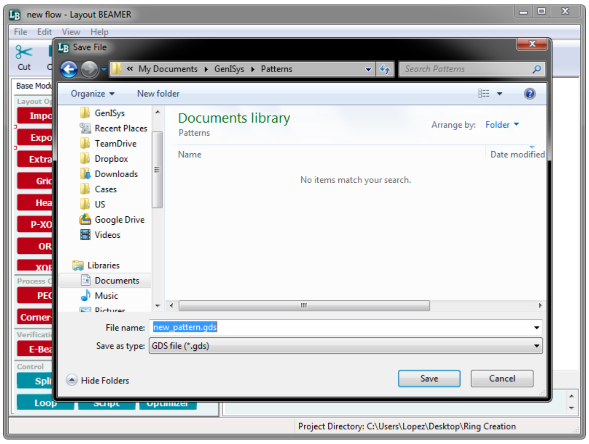
2. Drag an Export module and attach it to the out port of the Edit module.



new flow - Layout BEAMER

Project Directory: C:\Users\Lopez\Desktop\Ring Creation

3. Type the name of the file and select the format you wish to save the pattern.



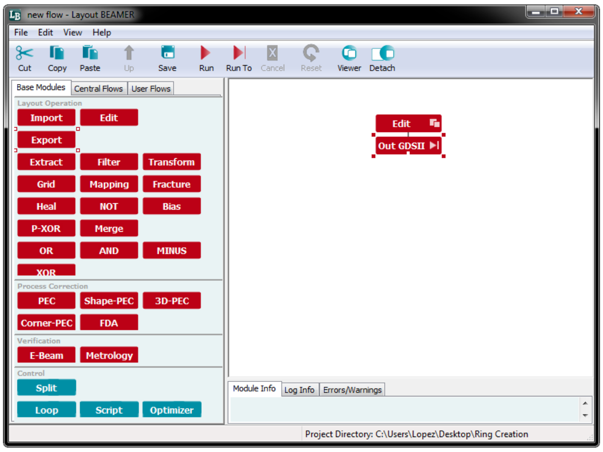
No items match your search.

File name: new_pattern.gds

Save as type: GDS file (*.gds)

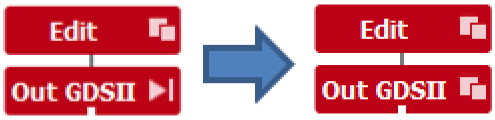
Project Directory: C:\Users\Lopez\Desktop\Ring Creation

4. Once you have entered the name and configured the output format, you will see the following screen.



Project Directory: C:\Users\Lopez\Desktop\Ring Creation

5. Click **Run** to execute the flow which will save your pattern. This will change the ▶| icon on the Export module to ⧉. Your pattern created in LayoutEditor from the Edit module is now saved to disk.



10 Frequently Asked Questions (FAQ)

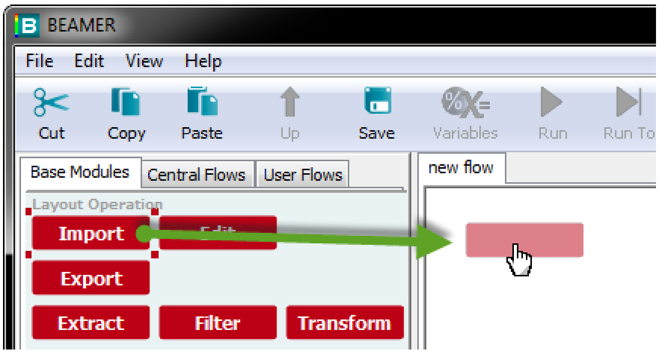
BEAMER has unlimited applications for preparing data, performing corrections and doing verification using the integrated simulation and modeling capabilities. The GenISys application group works closely with customers to develop new use cases that we make available to users from time to time. The applications listed below are a great resource for becoming familiar with using some of the basic and advanced features of the software. We encourage BEAMER users worldwide to submit their applications for inclusion into our documentation so that it can be shared with others. In this section, frequently asked questions (FAQs) and application cases are answered and explained.

10.1 Data Preparation and Inspection FAQ

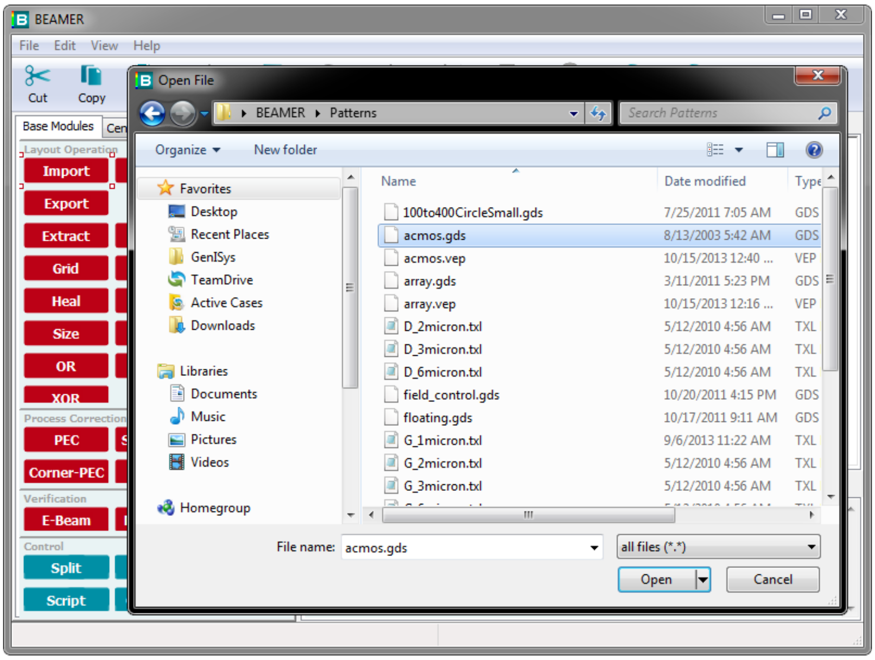
Data preparation and inspection are essential to prepare good data for optimized exposures. In this section, the most commonly asked questions are addressed for data preparation.

10.1.1 How can I import layout and export it to a specific format?

To import a layout, drag and drop the Import module into the workspace.



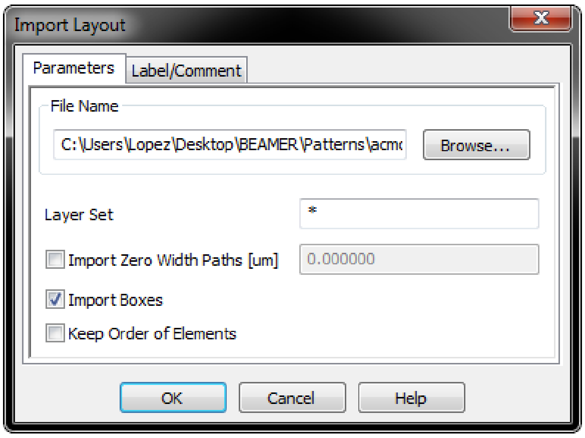
When released, it will fall into place and the Open File dialog will appear. Select a file to open and click the Open button.



Name	Date modified	Type
100to400CircleSmall.gds	7/25/2011 7:05 AM	GDS
acmos.gds	8/13/2003 5:42 AM	GDS
acmos.vep	10/15/2013 12:40 ...	VEP
array.gds	3/11/2011 5:23 PM	GDS
array.vep	10/15/2013 12:16 ...	VEP
D_2micron.txl	5/12/2010 4:56 AM	TXL
D_3micron.txl	5/12/2010 4:56 AM	TXL
D_6micron.txl	5/12/2010 4:56 AM	TXL
field_control.gds	10/20/2011 4:15 PM	GDS
floating.gds	10/17/2011 9:11 AM	GDS
G_1micron.txl	9/6/2013 11:22 AM	TXL
G_2micron.txl	5/12/2010 4:56 AM	TXL
G_3micron.txl	5/12/2010 4:56 AM	TXL

File name: acmos.gds
all files (*.*)

The Import Layout dialog will appear. Accept the default import settings by clicking OK.



Import Layout

Parameters | Label/Comment

File Name: C:\Users\Lopez\Desktop\BEAMER\Patterns\acmc

Browse...

Layer Set: *

□ Import Zero Width Paths [um]: 0.000000

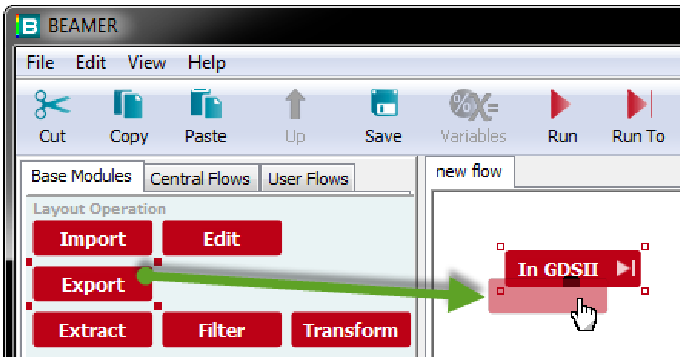
☑ Import Boxes

□ Keep Order of Elements

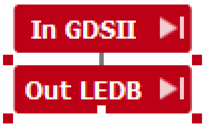
OK | Cancel | Help

The Import module label changes to indicate what will be imported. In this example, a GDS pattern will be imported into BEAMER. The module however has not been ran as indicated by the icon ▶| on the module.

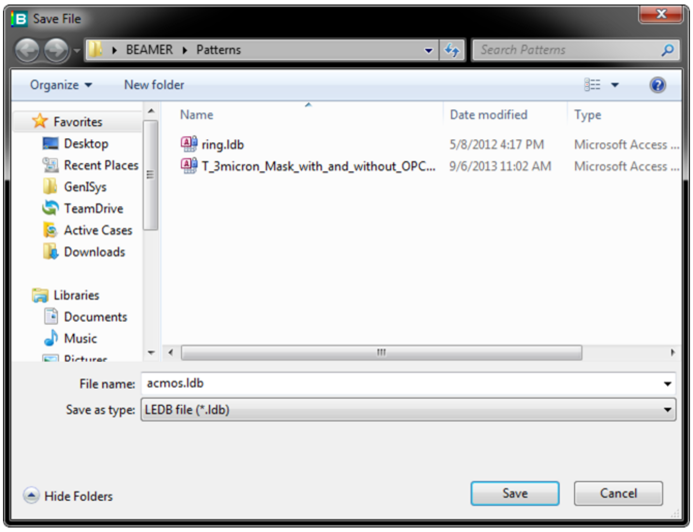
To import the pattern, you can click Run on the toolbar ▶ Run or the "Run To" icon on the module ▶|. When complete the icon will change to ⧉. Now drag an Export module into the work pane and place it on top of the Import module, until the small white square (middle, bottom of Import icon) turns black.



When released the Export module will attach itself to the Import module, and the Save File dialog will open.



Choose a format to save as, provide a file name and click Save.



Name	Date modified	Type
ring.ldb	5/8/2012 4:17 PM	Microsoft Access ...
T_3micron_Mask_with_and_without_OPC...	9/6/2013 11:02 AM	Microsoft Access ...

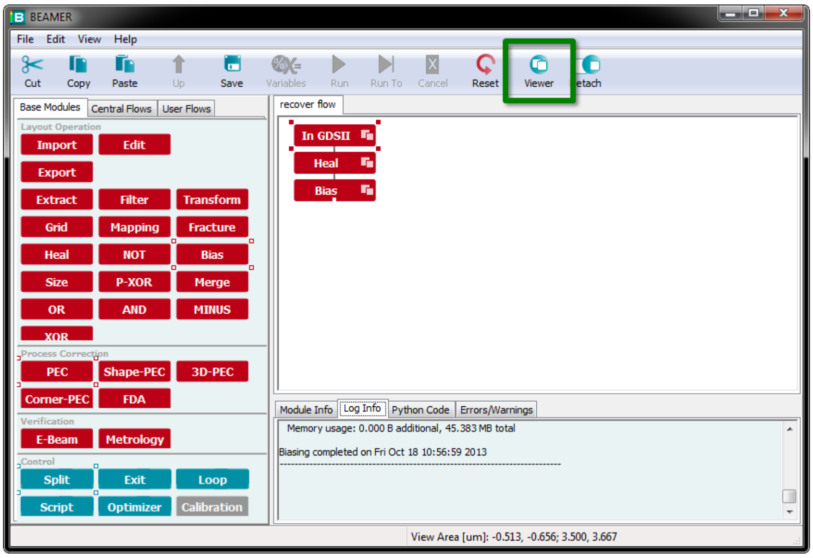
File name: acmos.ldb

Save as type: LEDB file (*.ldb)

Depending on the export format, a dialog will open to allow customization of the export settings. The details of the export formats can be looked up here. Click OK to confirm and run the Export module by clicking Run on the toolbar Run or the "Run To" icon on the module ⏭.

10.1.2 How can I quickly view and compare the result of executed modules?

To quickly view the result of every executed module in a flow, use the inline Global VIEWER. The Global VIEWER is activated by clicking the Viewer button in the toolbar:

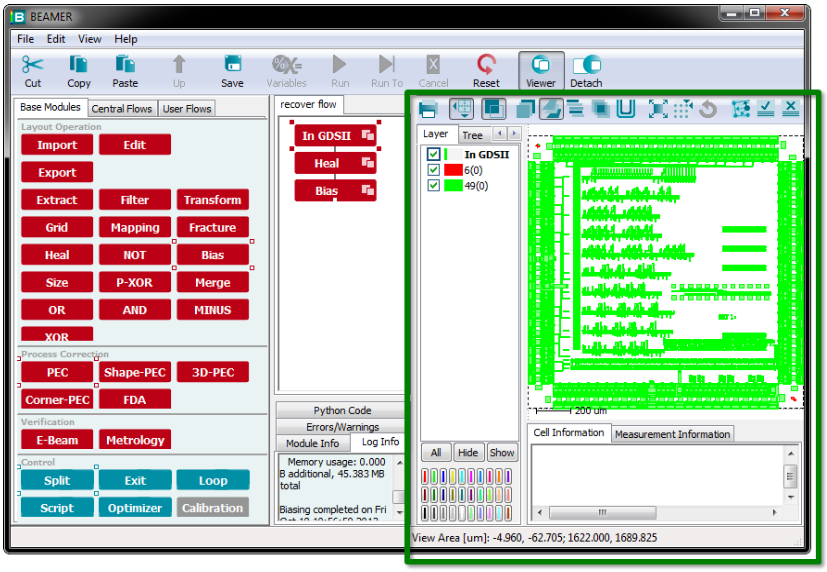

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII Heal Bias
Module Info Log Info Python Code Errors/Warnings

Memory usage: 0.000 B additional, 45.383 MB total

Biasing completed on Fri Oct 18 10:56:59 2013

View Area [um]: -0.513, -0.656; 3.500, 3.667

By clicking the Viewer button, the inline Global VIEWER will appear as highlighted below:


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
recover flow
In GDSII Heal Bias
Layer Tree
In GDSII
6(0)
49(0)
200 um
All Hide Show
Cell Information Measurement Information
Python Code Errors/Warnings Module Info Log Info

Memory usage: 0.000 B additional, 45.383 MB total

Biasing completed on Fri

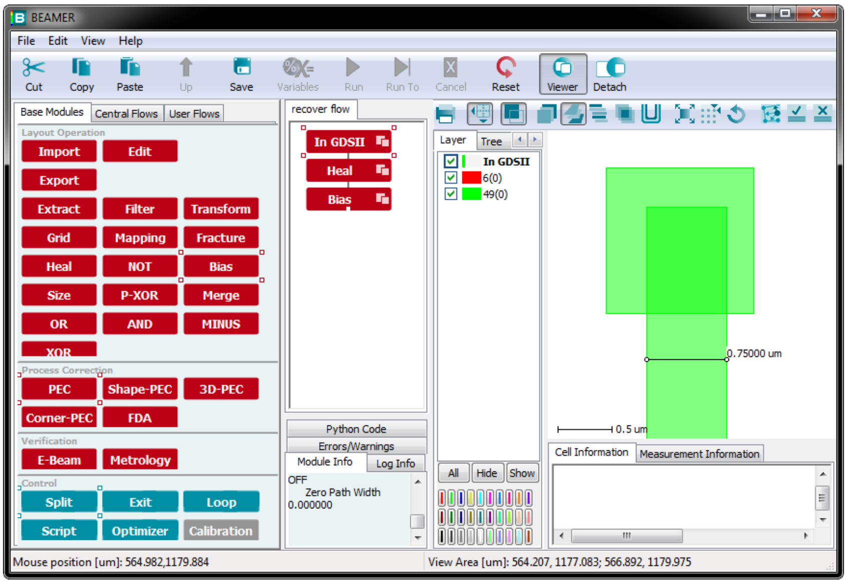
View Area [um]: -4.960, -62.705; 1622.000, 1689.825

In this example, we have imported the acmos.gds pattern. This pattern contains overlaps, so we are removing the overlaps using the Heal module. Moreover, in the pattern there are 750nm wide interconnects (lines or wires) that are coming out as 800nm wide features when we develop the resist. This tells us that we have a process bias of 50nm. To compensate for the process bias, we are negatively biasing the pattern by -25nm on every side. This means that the pattern we want to print should have 700nm drawn interconnects, but with a process bias of 50nm, we should get 750nm wide lines.

The inline Global VIEWER can be used to quickly inspect the imported GDSII file by selecting only the Import module named "In GDSII." Here we can see an overlap between two shapes when zoomed-in. We can also measure the width of the interconnect to see that it is 750nm.

The VIEWER window allows you to browse through loaded layouts with ease:

Action	Mouse/Keyboard Description
Zoom-In	• Left-click and drag the mouse from top-left to bottom-right on the viewable layout or • Turn the mouse wheel to the front
Zoom-Out	• Left-click and drag the mouse from bottom-right to top-left on the viewable layout • Turn the mouse wheel to the back
Centering	Left mouse click at a position to center the layout at that position
Panning	Press and hold the middle mouse button and move the mouse around. Releasing that button will then redraw the layout. Additionally the arrow keys can be used to navigate through the layout.
Measure	Right-click near one edge and right-click near another edge. A measurement will automatically be taken.
Snap to Corner or Edge for Pick Information	CTRL + double right-click near an edge or corner
Pick Information for a shape	Double right-click on a shape
Go to Coordinates	• CTRL + G • Also see Go to Coordinates



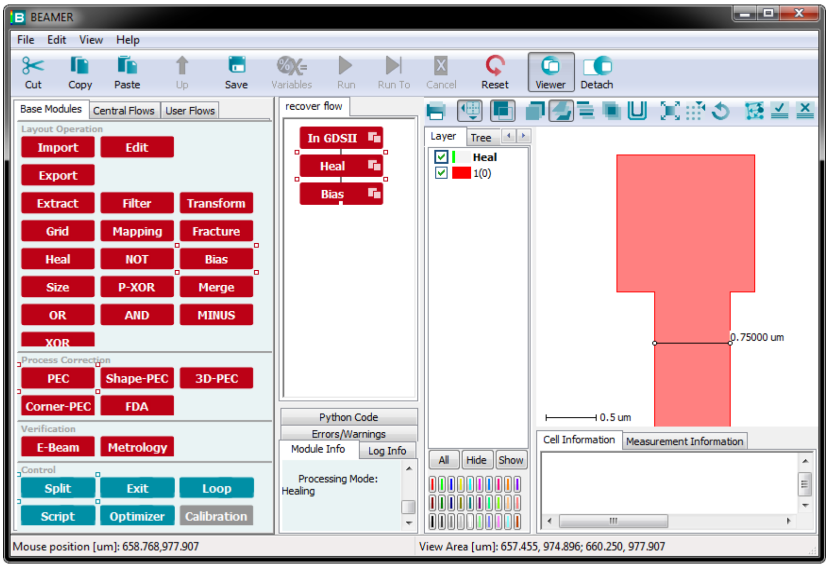
OFF
Zero Path Width
0.000000

☑ In GDSII
☑ 🟥 6(0)
☑ 🟩 49(0)

Mouse position [um]: 564.982,1179.884

View Area [um]: 564.207, 1177.083; 566.892, 1179.975

By clicking the Heal module to select it, the resulting pattern from the executed Heal module is shown. You can immediately notice that the pattern has been placed on layer 1(0) as it has been essentially flattened moved to layer 1(0) by default. Zooming into the pattern, you can see that the overlaps are removed and the interconnect is still 750nm wide:

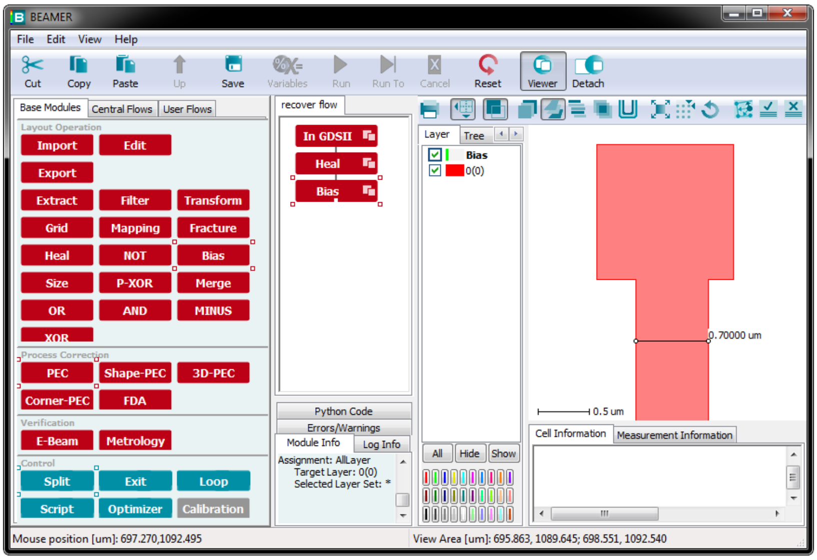

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
recover flow
In GDSII
Heal
Bias
Layer Tree
Heal
1(0)
0.75000 um
0.5 um
Python Code
Errors/Warnings
Module Info Log Info
All Hide Show
Cell Information Measurement Information

Processing Mode:
Healing

Mouse position [um]: 658.768,977.907

View Area [um]: 657.455, 974.896; 660.250, 977.907

By selecting the Bias module, the pattern can now be seen on layer 0(0). This is because the target layer for the Bias module is set to 0(0) by default. Zooming in and measuring shows that the interconnect is indeed 700nm since a -25nm bias was applied to every side.

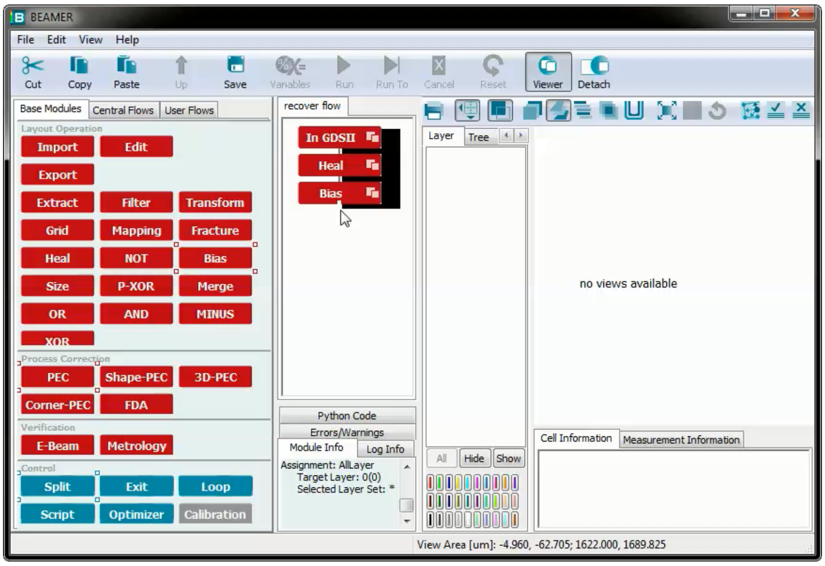

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
recover flow
In GDSII
Heal
Bias
Layer Tree
Bias
0(0)
0.70000 um
0.5 um
Python Code
Errors/Warnings
Module Info Log Info
All Hide Show
Cell Information Measurement Information

Assignment: AllLayer
Target Layer: 0(0)
Selected Layer Set: *

Mouse position [um]: 697.270,1092.495

View Area [um]: 695.863, 1089.645; 698.551, 1092.540

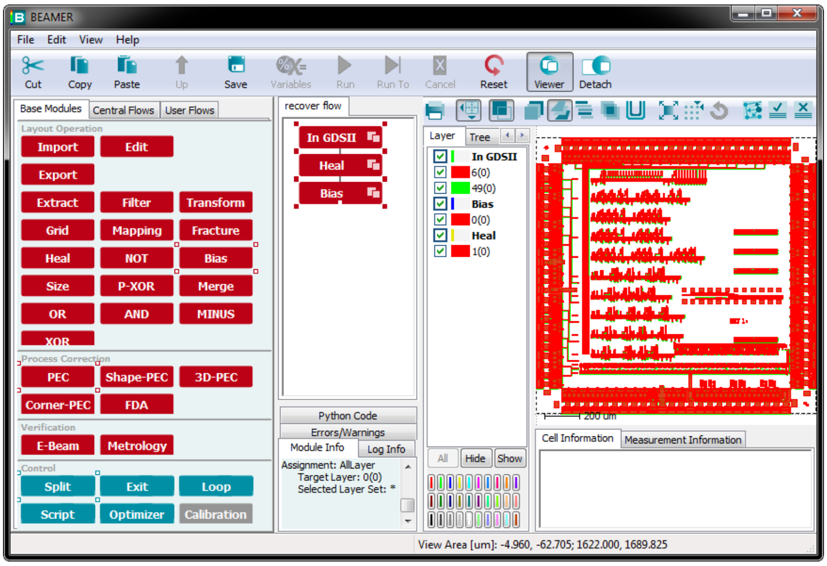
So far the advantage to using the inline Global VIEWER is the quick inspection to executed modules in the BEAMER process flow by simply selecting the executed module of interest. A more powerful application is to compare multiple execute modules in a flow. This will place the resulting layout of each module atop of one another. To select multiple modules, click in the white area of the workspace and drag the mouse to create a black selection region that touches all three modules as shown below:


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII
Heal
Bias
Python Code
Errors/Warnings
Module Info Log Info
Layer Tree
All Hide Show
no views available
Cell Information Measurement Information

Assignment: AllLayer
Target Layer: 0(0)
Selected Layer Set: *

View Area [um]: -4.960, -62.705; 1622.000, 1689.825

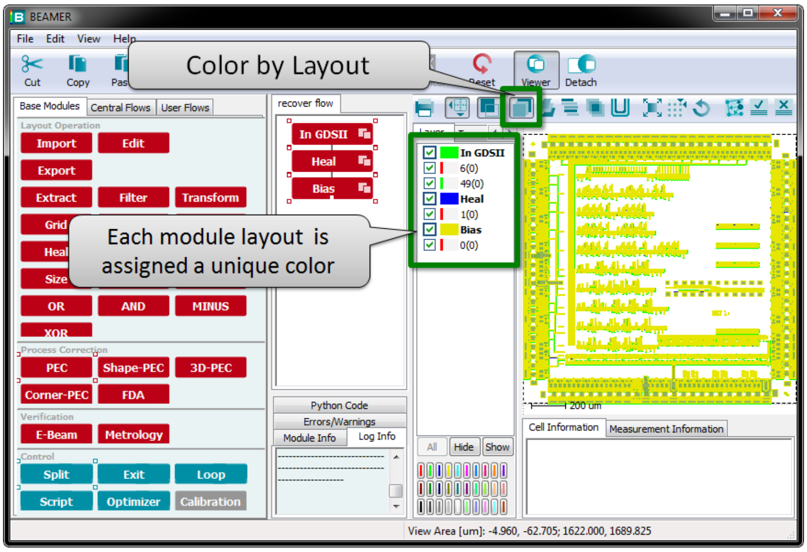
Once all three modules are selected, the resulting patterns from each module will be placed atop of one another:


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII
Heal
Bias
Python Code
Errors/Warnings
Module Info Log Info
Layer Tree
In GDSII
6(0)
49(0)
Bias
0(0)
Heal
1(0)
All Hide Show
200 um
Cell Information Measurement Information

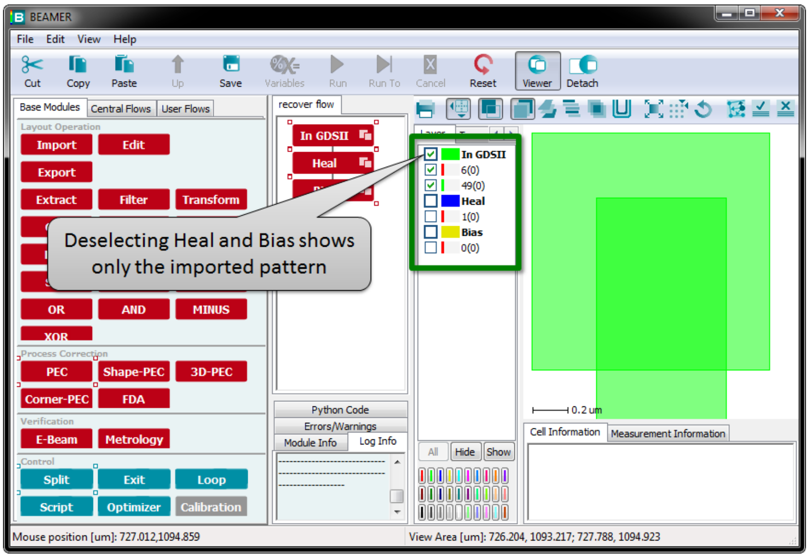
Assignment: AllLayer
Target Layer: 0(0)
Selected Layer Set: *

View Area [um]: -4.960, -62.705; 1622.000, 1689.825

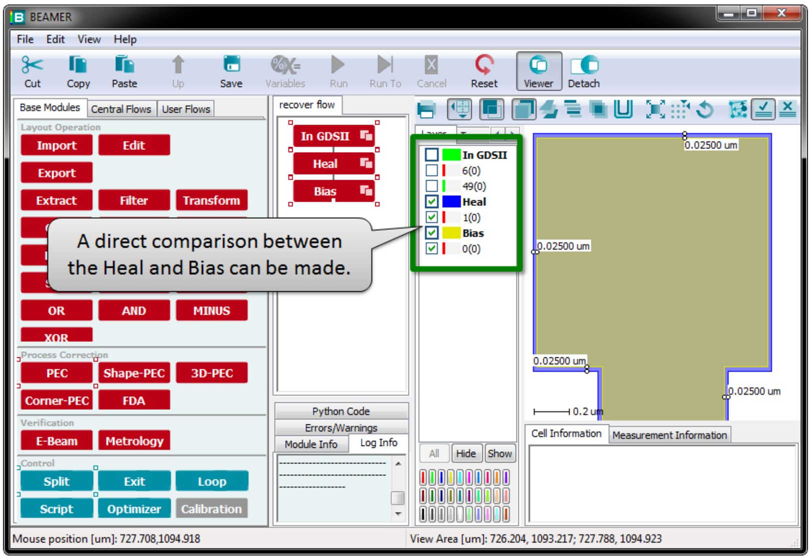
To facilitate viewing, click on the Color by Layout button found in the inline Global VIEWER. Immediately, each module layout is assigned a unique color:



You can select only one module layout to be visible by deselecting the Heal and Bias layouts in the VIEWER:

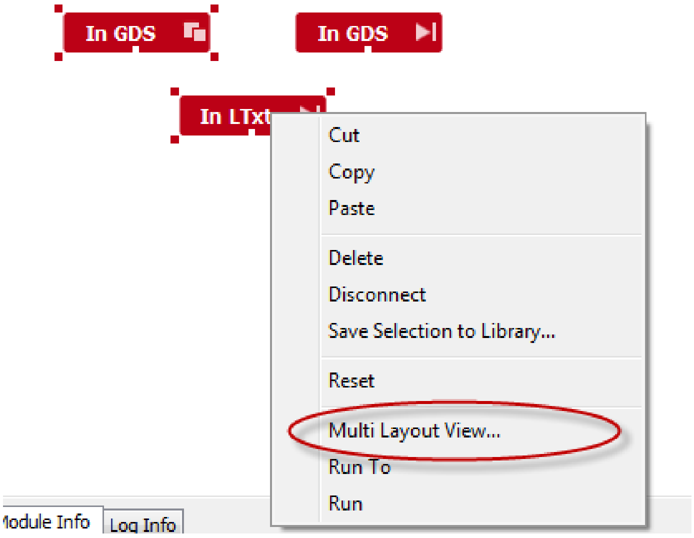


Or, you can check the Heal and Bias layouts only (and deselecting "In GDSII") to compare the result of the two operations. Here it is apparent that the Bias negatively biases the pattern by 25nm on every side:


A direct comparison between the Heal and Bias can be made.
In GDSII
Heal
Bias
Mouse position [um]: 727.708,1094.918
View Area [um]: 726.204, 1093.217; 727.788, 1094.923

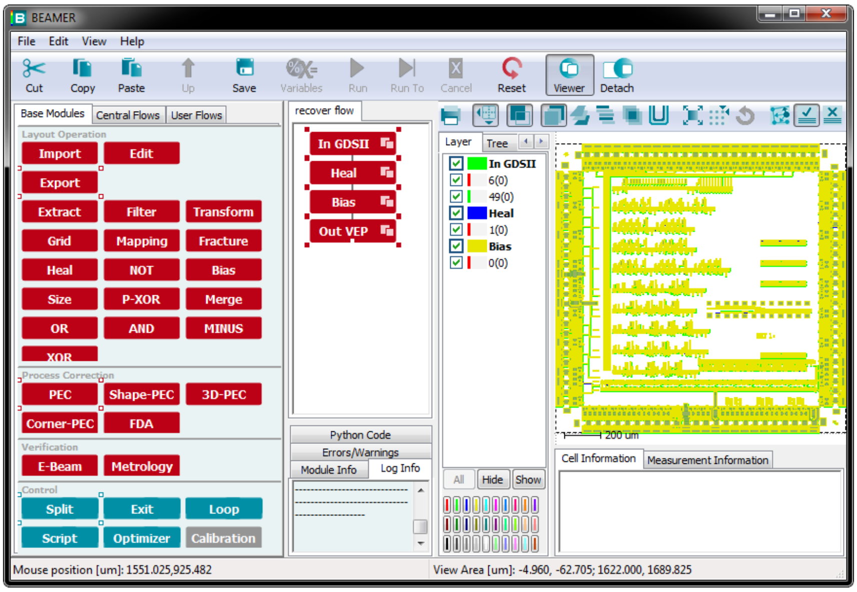
The Global VIEWER can be used to quickly inspect and compare the data in a flow. As shown, above, the Global VIEWER provides immediate inspection of multiple modules at the same time.

Alternatively, Multiple layouts can be viewed at the same time using the normal VIEWER. To select multiple modules that contain viewable layouts, draw a box around the modules to be viewed with the left mouse button or holding down the Control Key and then click on the modules of interest. Finally, right click on the label on any of these modules to bring up a context menu where you select "Multi Layout View..." to open all selected layouts in a separate VIEWER.



10.1.3 Why can't I view the Export module in the Global VIEWER?

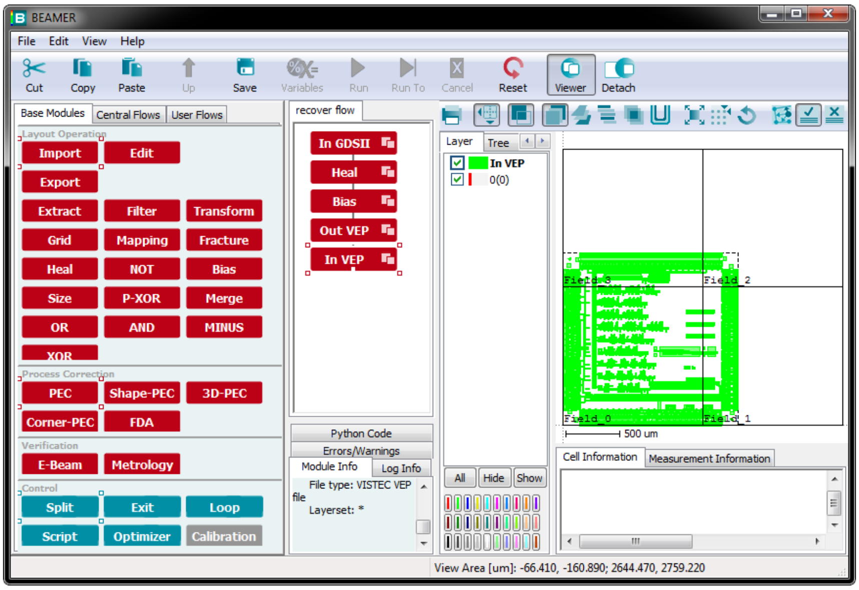
The Global VIEWER can only show the layouts that have been loaded into memory. For example, given the following configuration below, all modules have been selected, but only the Import, Heal and Bias layouts are visible for comparison in the Global VIEWER:


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII Heal Bias Out VEP
Python Code Errors/Warnings Module Info Log Info
Layer Tree
In GDSII 6(0) 49(0) Heal 1(0) Bias 0(0)
All Hide Show
200 um
Cell Information Measurement Information

Mouse position [um]: 1551.025,925.482

View Area [um]: -4.960, -62.705; 1622.000, 1689.825

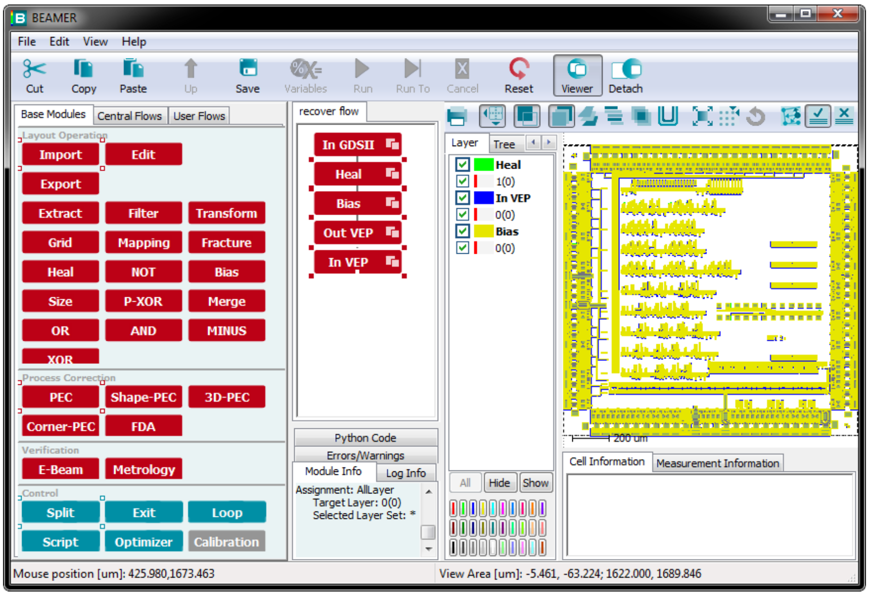
To view the contents of the Exported file, simply re-import the file by adding an Import module to the Export module. When attaching the Import module to the bottom of Export module, the Import module will automatically inherit the file name of the exported file. Run the Import module and select it (and other modules if you desire) to view the pattern in the Global VIEWER. Below, only the Import module is selected for the exported machine format. If the machine format supports fields, the main field boundaries are visible in the VIEWER like below:


BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII Heal Bias Out VEP In VEP
Python Code Errors/Warnings Module Info Log Info
Layer Tree
In VEP 0(0)
All Hide Show
Field_3 Field_2 Field_0 Field_1
500 um
Cell Information Measurement Information

File type: VISTEC VEP file
Layerset: *

View Area [um]: -66.410, -160.890; 2644.470, 2759.220

With the Import module executed, you should now be able to compare the exported pattern with the other modules directly. Below, we are able to compare the Heal, Bias and Exported machine file to one another in the Global VIEWER:



Assignment: AllLayer
Target Layer: 0(0)
Selected Layer Set: *

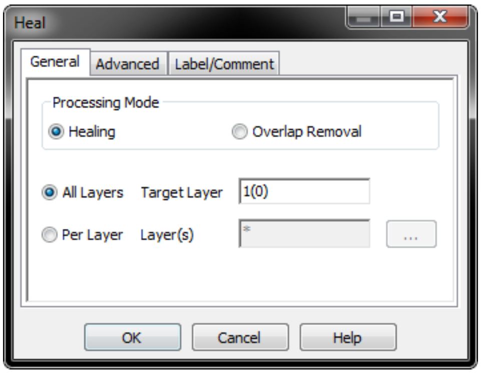
Mouse position [um]: 425.980,1673.463

View Area [um]: -5.461, -63.224; 1622.000, 1689.846

10.1.4 How can I remove overlaps within a layout?

Sometimes, layout data may have overlapping polygons. This condition can lead to double exposure and result in poor shape fidelity.

The Heal module removes overlaps in the layout and joins abutting polygons into one. Overlaps can cause unwanted double exposure during writing resulting in poor shape fidelity. The default Heal settings will remove overlaps and merge the remaining elements into a single polygon onto a single layer. The default target layer is 1(0). If the Overlap Removal (OLR) radio button is selected only overlaps are removed with the pattern placed on layer 1(0). With OLR, polygon merging will not take place.

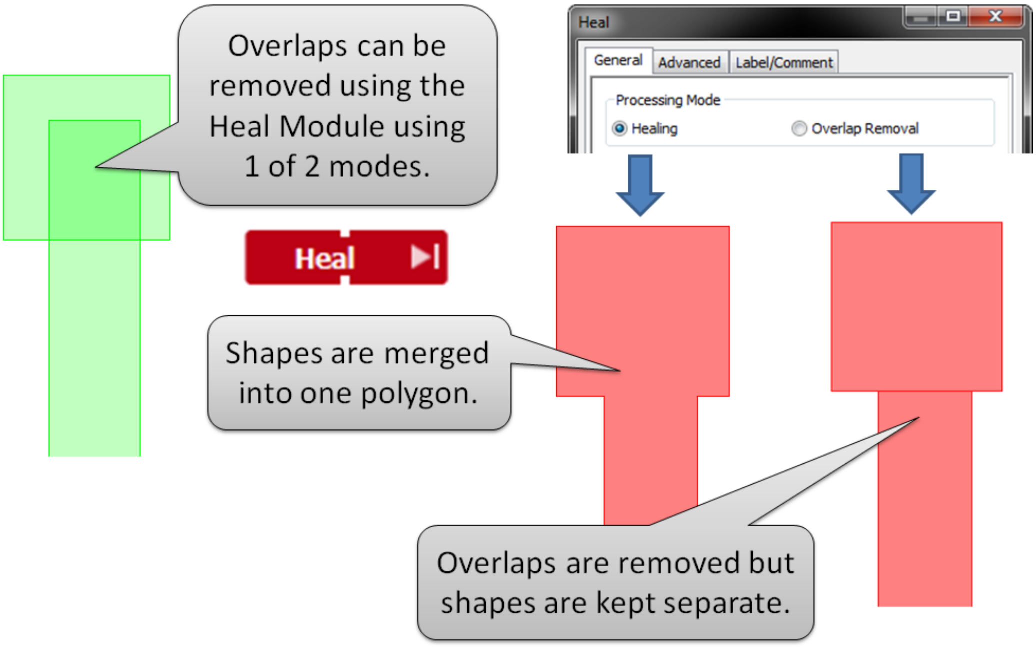


Processing Mode

● Healing ○ Overlap Removal

● All Layers Target Layer 1(0)

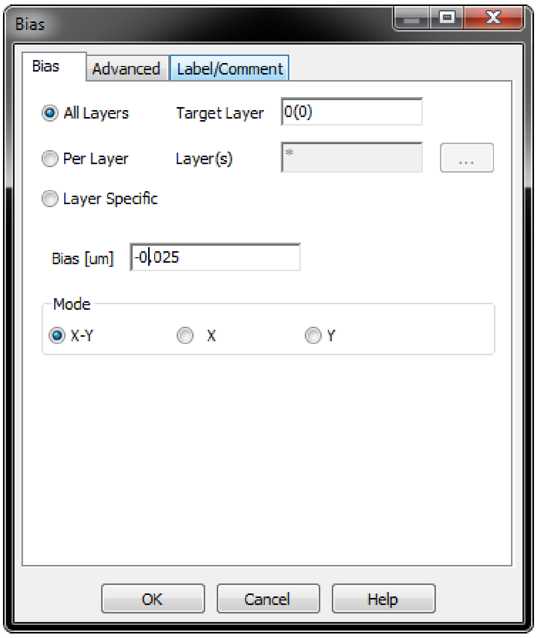
○ Per Layer Layer(s) *


Overlaps can be removed using the Heal Module using 1 of 2 modes.
Heal
Heal
General
Advanced
Label/Comment
Processing Mode
Healing
Overlap Removal
Shapes are merged into one polygon.
Overlaps are removed but shapes are kept separate.

10.1.5 How can I (positively or negatively) bias my layout?

The Bias module is used to re-size layout elements, without scaling which would also scale the spaces. This sizing can be either positive or negative. When the module is dropped into the flow design pane a Parameters dialog box will open.

The Bias module is explained in greater detail in the Layout Operation Module section called Bias.



● All Layers Target Layer 0(0)

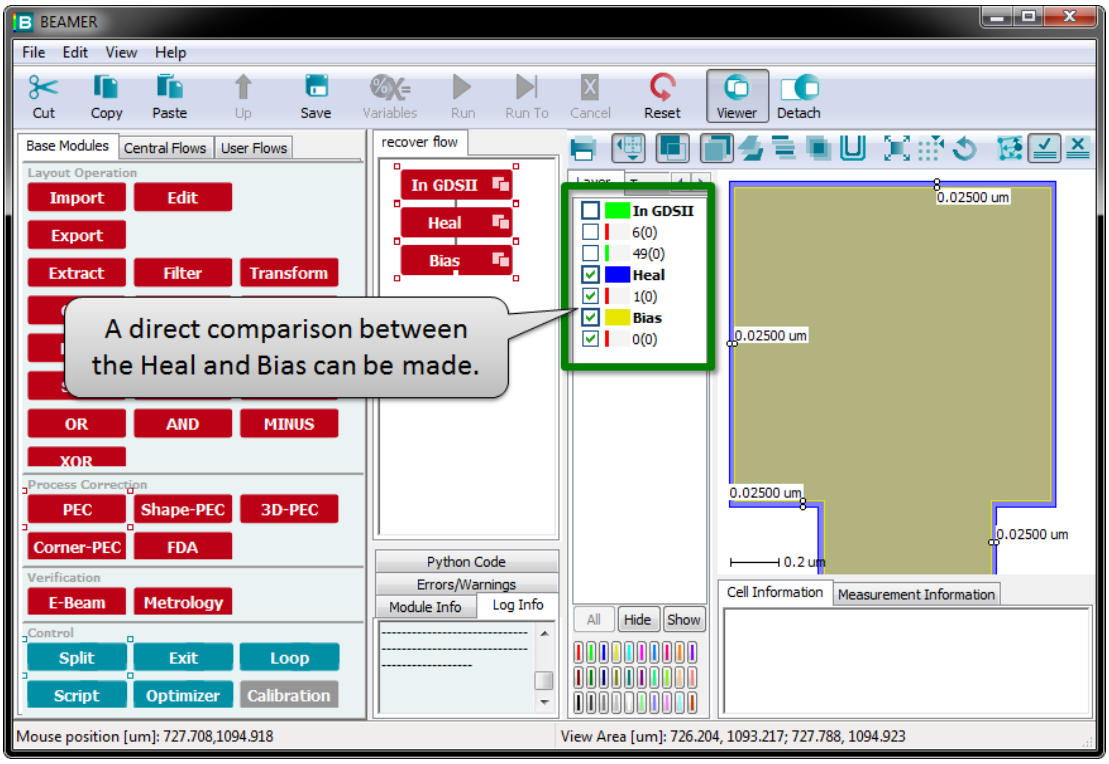
○ Per Layer Layer(s) *

○ Layer Specific

Bias [um] -0.025

Mode: ● X-Y ○ X ○ Y

Below is an example from How can I quickly view and compare the result of executed modules? This shows an example of a Bias in use after a Heal operation. A quick comparison in the Global VIEWER shows the result of the -25nm bias in yellow versus the Heal layout in blue:



A direct comparison between the Heal and Bias can be made.

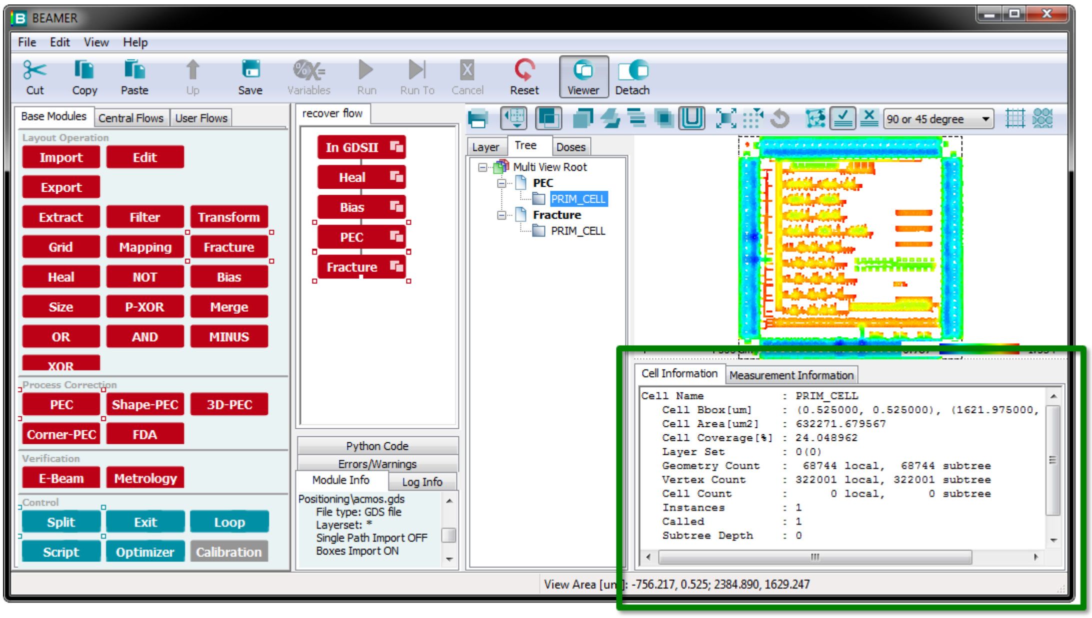
□ 🟩	In GDSII
□	6(0)
□	49(0)
☑ 🟦	Heal
☑	1(0)
☑ 🟨	Bias
☑	0(0)

Mouse position [um]: 727.708,1094.918

View Area [um]: 726.204, 1093.217; 727.788, 1094.923

10.1.6 Where can I find the area, number of shapes, vertices, etc. for a pattern?

The area, number of shapes and number of vertices within a pattern are readily available in the VIEWER. Using the Global VIEWER, select any number of modules in the flow to quickly enable a multi-view and click on the Tree tab in the VIEWER. From there, expand the tree per module and select the topmost cell within module. In the example below, the PEC module top cell called PRIM_CELL has been selected. In the Cell Information panel (highlighted in green), all pertinent information regarding the cell is obtained. The area of the layout is given as the Cell Area; the number of shapes is given as the Geometry count; and the number of vertices are given as Vertex Count below:



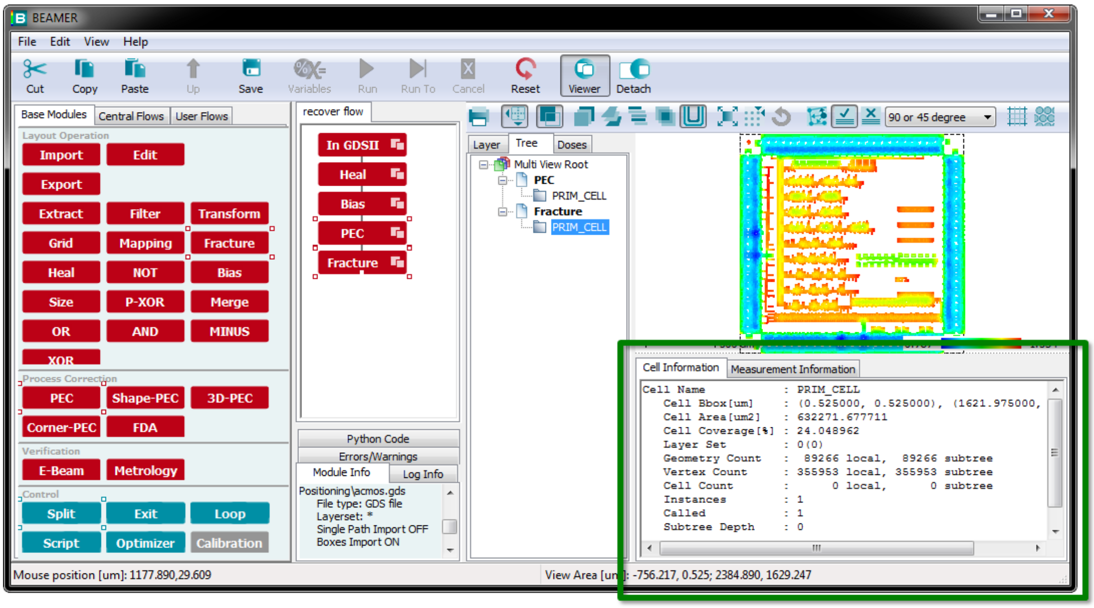
```
Positioning\acmos.gds
   File type: GDS file
   Layerset: *
   Single Path Import OFF
   Boxes Import ON
```

```
Cell Name           : PRIM_CELL
   Cell Bbox[um]    : (0.525000, 0.525000), (1621.975000,
   Cell Area[um2]   : 632271.679567
   Cell Coverage[%] : 24.048962
   Layer Set        : 0(0)
   Geometry Count   :  68744 local,  68744 subtree
   Vertex Count     : 322001 local, 322001 subtree
   Cell Count       :      0 local,      0 subtree
   Instances        : 1
   Called           : 1
   Subtree Depth    : 0
```

View Area [um]: -756.217, 0.525; 2384.890, 1629.247

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII Heal Bias PEC Fracture
Python Code Errors/Warnings Module Info Log Info
Layer Tree Doses
Multi View Root PEC PRIM_CELL Fracture PRIM_CELL
90 or 45 degree
Cell Information Measurement Information

In the view above, we see that the PEC module produces 68,744 shapes total. However since many of these shapes are still polygons, an additional fracture is needed to reduce the polygons into primitive shapes (trapezoids and rectangles) so that the data can be understood by the e-beam's pattern generator. In lieu of a machine export, a Fracture module is used to machine independently fracture the corrected pattern. As a result, the number of shapes is 89,266:



```
Positioning\acmos.gds
   File type: GDS file
   Layerset: *
   Single Path Import OFF
   Boxes Import ON
```

```
Cell Name           : PRIM_CELL
   Cell Bbox[um]    : (0.525000, 0.525000), (1621.975000,
   Cell Area[um2]   : 632271.677711
   Cell Coverage[%] : 24.048962
   Layer Set        : 0(0)
   Geometry Count   :  89266 local,  89266 subtree
   Vertex Count     : 355953 local, 355953 subtree
   Cell Count       :      0 local,      0 subtree
   Instances        : 1
   Called           : 1
   Subtree Depth    : 0
```

Mouse position [um]: 1177.890,29.609

View Area [um]: -756.217, 0.525; 2384.890, 1629.247

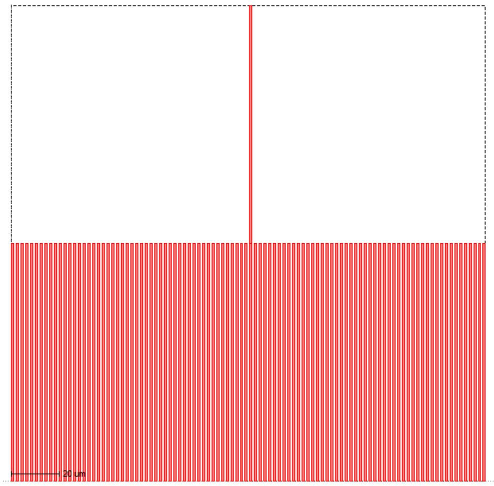
BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII Heal Bias PEC Fracture
Python Code Errors/Warnings Module Info Log Info
Layer Tree Doses
Multi View Root PEC PRIM_CELL Fracture PRIM_CELL
90 or 45 degree
Cell Information Measurement Information

The VIEWER provides a quick way to determine the area, number of shapes and the number of vertices within a pattern while comparing multiple modules at the same time.

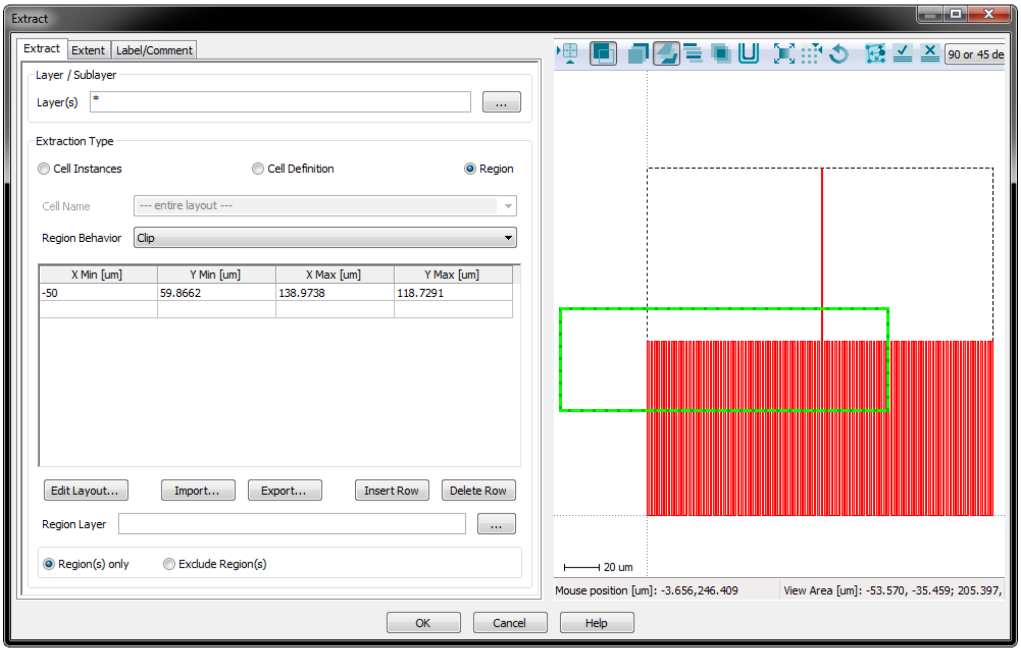
10.1.7 What is the extent? How can I control the extent?

The extent of the layout is a box from the lower left limit to the upper right limit of the originally designed data. In other words, it is the furthest edge of a pattern and is considered to be the writable area of the pattern. This means that no shapes can exist outside the extent boundary. The extent is not to be used to extract an area or region of interest for exposure. It is intended to only define the area to expose. By default, the original extent of a pattern is always maintained when using the Extract module.

BEAMER has multiple options for manipulating the extent (that are defined in the section on the Extent Tab) using the Extract module's Extent Tab. By default, the extent is preserved with the Extract module. The extent is seen as the dotted box around the pattern.



To demonstrate the Extent options, a region of interest (ROI) will be defined in the Extract tab. The green border indicates the ROI for extraction.



Layer / Sublayer

Layer(s): *

Extraction Type

○ Cell Instances ○ Cell Definition ● Region

Cell Name: --- entire layout ---

Region Behavior: Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
-50	59.8662	138.9738	118.7291

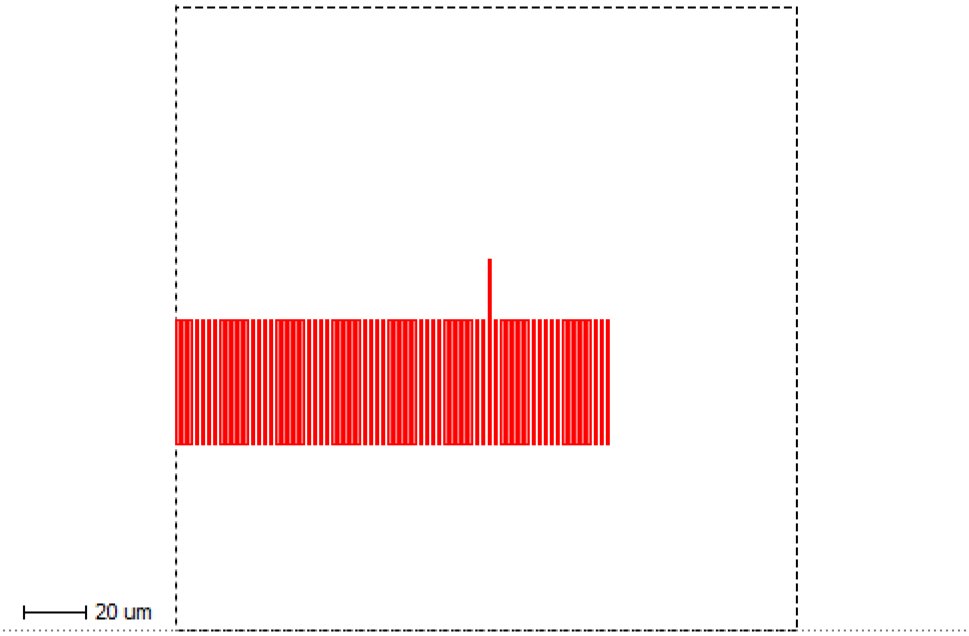
Region Layer:

● Region(s) only ○ Exclude Region(s)

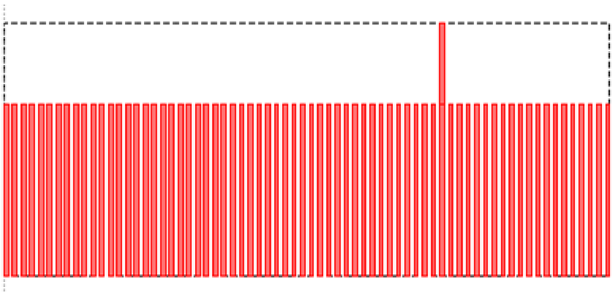
Mouse position [um]: -3.656,246.409

View Area [um]: -53.570, -35.459; 205.397,

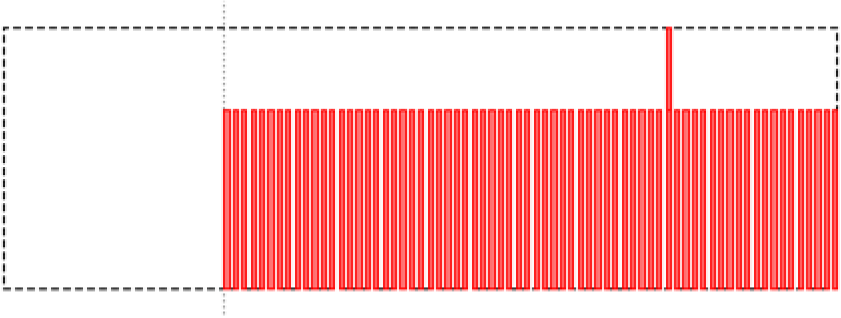
The first option for Extent is Automatic. This option uses the extent of the original layout and applies it to the extracted ROI.



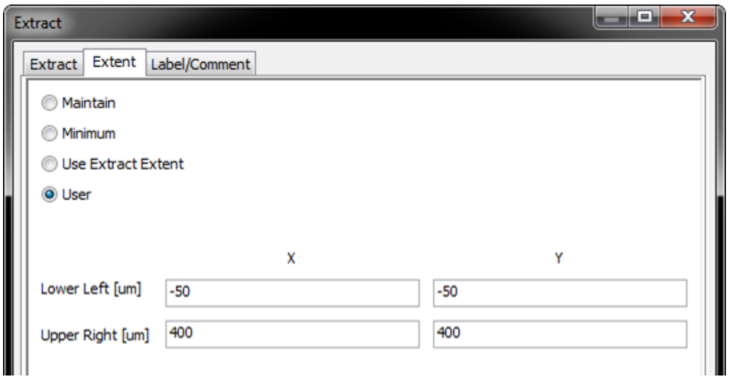
The Minimum option snaps the extent to the furthermost edge of the extracted ROI. If multiple ROIs are extracted, the extent is the furthermost edge of the combined ROIs.



The option Use Extract Extent sets the extent so that it is identical to the ROI. If multiple ROIs are extracted, the extent is the furthermost edge of the combined ROIs



The last option is the User-defined extent. Exact coordinates may be entered to define the extent as shown below. The extent can be no smaller than the furthermost edge of the ROI, otherwise BEAMER will complain. The extent definition is not to be used for extracting a ROI. It is to define the area of exposure; therefore shapes cannot lie outside the extent.



Extract

○ Maintain

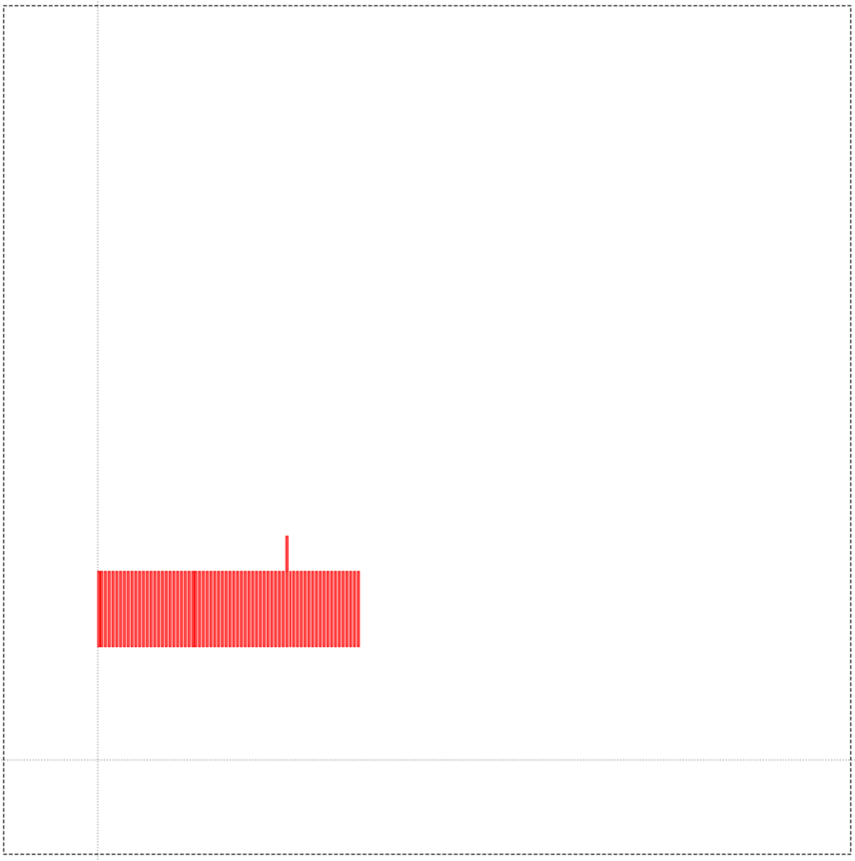
○ Minimum

○ Use Extract Extent

● User

	X	Y
Lower Left [um]	-50	-50
Upper Right [um]	400	400

Below, the user defined extent is defined as a much larger extent than the original pattern.



10.1.8 How can I extract a single layer or multiple layers from a layout?

To extract a layer from a layout, the Import or Extract module can be used. You should be aware of the extent when performing this kind of operation as each module exhibit different extent behavior.

The extent of the layout is a box from the lower left limit to the upper right limit of the originally designed data. In other words, it is the furthest edge of a pattern and is considered to be the writable area of the pattern. This means that no shapes can exist outside the extent boundary. The extent is not to be used to extract an area or region of interest for exposure. It is intended to only define the area to expose. By default, the original extent of a pattern is always maintained when using the Extract module.

The Import module for GDSII, CIF, DXF and OASIS has a field that facilitates a combination of layer/datatype extractions called Layer Set. Below, a screen shot of the GDSII dialog is shown with the Layer Set parameter highlighted in green. See the section on Layer/Datatype Handling to understand the layer/datatype syntax that can be used in this field. By default when importing a pattern, the Layer Set field has an asterisk (*) to indicate an import of all layers within the pattern. Extracting a layer or layers from a pattern using the Import module will not maintain the extent of the pattern, which may be important if you plan to expose different layers in the pattern with high overlay accuracy.

Using the Extract module is the preferred way to extract layers from a pattern while maintaining the extent. Use this module to maintain high overlay accuracy of multiple exposures for different layers within the layout. Use the Layer(s) field at the top of the Extract module to extract layers from a layout:

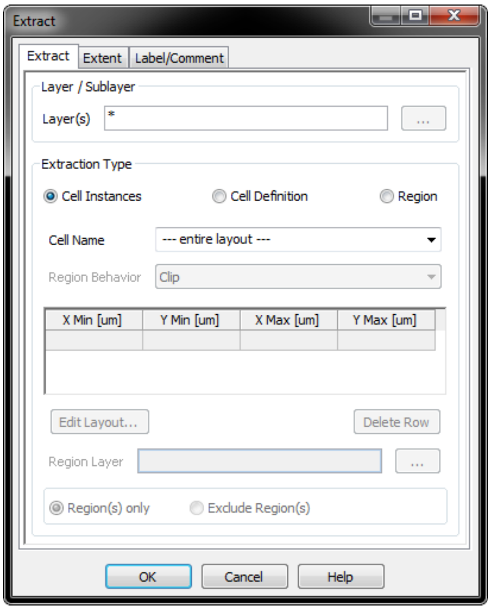
Layer(s)

The default setting, noted by the asterisk sign *, will load all layers and datatypes. To extract a specific layer or datatype, enter the specific value(s) in place of the asterisk sign. Multiple values or a range can be entered into the dialog as described in the following examples:

Layer(s): 5, 12
Will extract layers 5 and 12 from the layout.

Layer(s) 5-12,16
Will extract layers 5 through 12 and layer 16

For more information on special annotations available for this field, please read the section on layer datatype handling. To return to the default setting, enter the asterisk sign into the dialog, which will then load all available layers and datatypes.



Layer(s) *

● Cell Instances ○ Cell Definition ○ Region

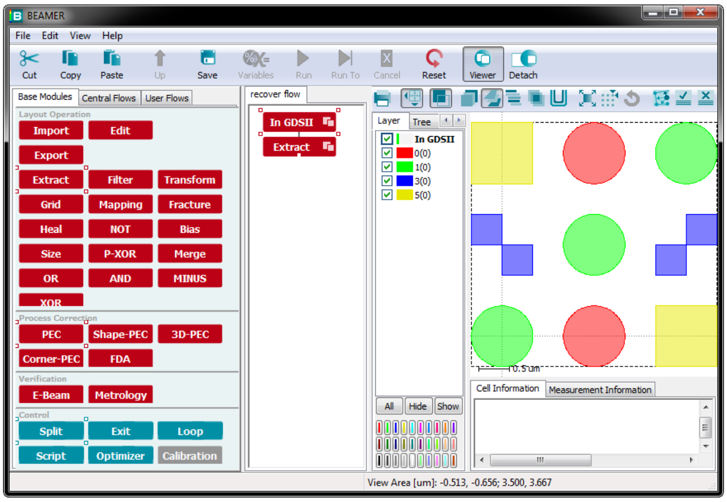
Cell Name --- entire layout ---

Region Behavior Clip

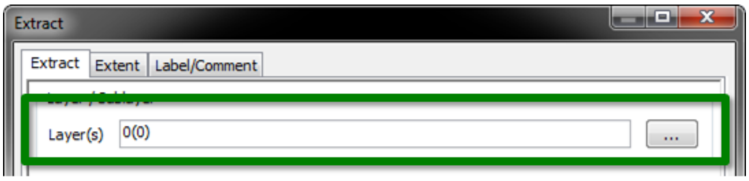
● Region(s) only ○ Exclude Region(s)

When the Extract is connected to an already executed module, the button, ..., will open a dialog displaying a list to select the layers interactively.

For a more concrete example, we have imported a GDSII file with shapes on layers 0, 1, 3 and 5. You can see that by selecting the import module labeled "In GDSII" and using the inline Global VIEWER to immediately view the pattern on the right.

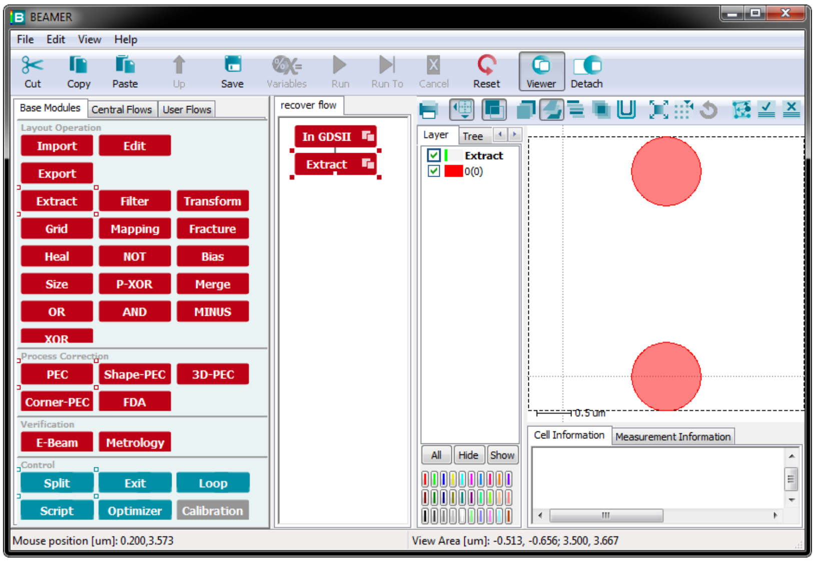

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII
Extract
Layer Tree
In GDSII
0(0)
1(0)
3(0)
5(0)
0.5um
All Hide Show
Cell Information Measurement Information
View Area [um]: -0.513, -0.656; 3.500, 3.667

In the Extract module, only layer 0 is in the Layer(s) field.

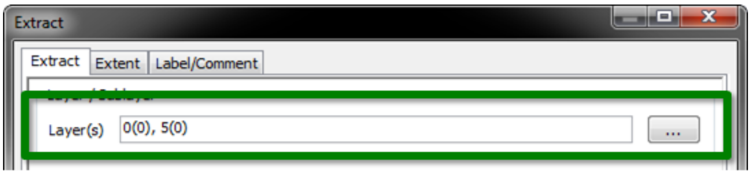


Layer(s) 0(0)

This results in layer 0 as the only extracted layer from the pattern. Notice how the original extent is maintained.

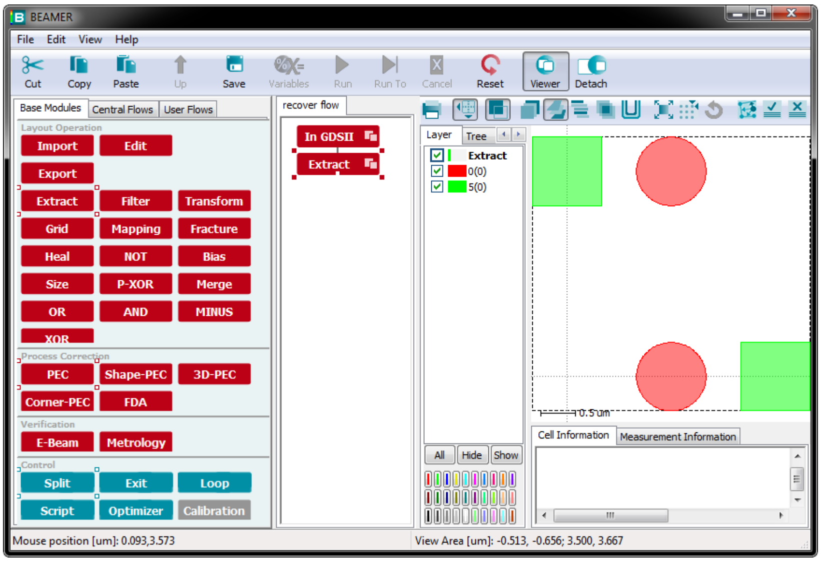

BEAMER
File Edit View Help
Cut Copy Paste Up Save Variables Run Run To Cancel Reset Viewer Detach
Base Modules Central Flows User Flows
Layout Operation
Import Edit Export Extract Filter Transform Grid Mapping Fracture Heal NOT Bias Size P-XOR Merge OR AND MINUS XOR
Process Correction
PEC Shape-PEC 3D-PEC Corner-PEC FDA
Verification
E-Beam Metrology
Control
Split Exit Loop Script Optimizer Calibration
recover flow
In GDSII
Extract
Layer Tree
Extract
0(0)
0.5um
All Hide Show
Cell Information Measurement Information
Mouse position [um]: 0.200,3.573
View Area [um]: -0.513, -0.656; 3.500, 3.667

We can always change the parameters in the Extract module. Here layers 0 and 5 are inputted into the Layer(s) field.



Layer(s) 0(0), 5(0)

Executing the Extract module, yields the following result with both layers 0 and 5 extracted:



Mouse position [um]: 0.093,3.573

View Area [um]: -0.513, -0.656; 3.500, 3.667

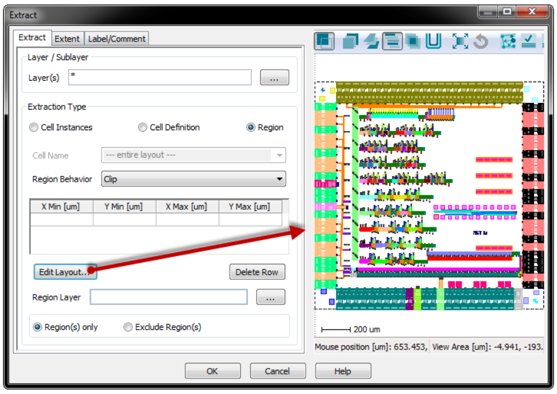
10.1.9 How can I extract a region of interest (ROI) from a layout?

The Extract module can also be used to extract single or multiple regions of interest (ROI's).

An ROI can be defined in one of three ways:

1. Manually entering coordinates into the ROI Table,
2. Defining regions of interest (ROI) using the mouse and keyboard or
3. Using *Region Layer(s)* defined in the pattern.

It is possible to use both the ROI table and the region layer(s) defined in the layout at the same time for a region extraction.



Layer(s) *

Extraction Type: ○ Cell Instances ○ Cell Definition ● Region

Cell Name: --- entire layout ---

Region Behavior: Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]

Region Layer:

● Region(s) only ○ Exclude Region(s)

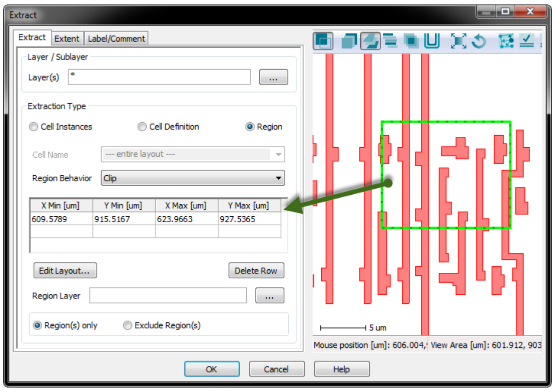
Mouse position [um]: 653.453, View Area [um]: -4.941, -193.

Extracting a region using the mouse and keyboard is performed by clicking the **Edit Layout...** button. Be sure that the module before the Extract module has been ran so that the **Edit Layout...** button is enabled. When this button is clicked, an inline VIEWER will appear as shown in the image above.To select a region inside the VIEWER:

1. Set starting corner of bounding box by holding the SHIFT key on the keyboard and clicking once with the left mouse button in the VIEWER.

2. Let go of the SHIFT key (and left mouse button) and move the mouse in the VIEWER to set the size of the bounding box.

3. Once you have moved the mouse to the desired size, confirm the size by holding the SHIFT key and clicking once with the left mouse button while maintaining the mouse position within the VIEWER. The coordinates of the green bounding box will be placed in the ROI Table as indicated with the arrow in the image.



Layer(s) *

Extraction Type: ○ Cell Instances ○ Cell Definition ● Region

Cell Name: --- entire layout ---

Region Behavior: Clip

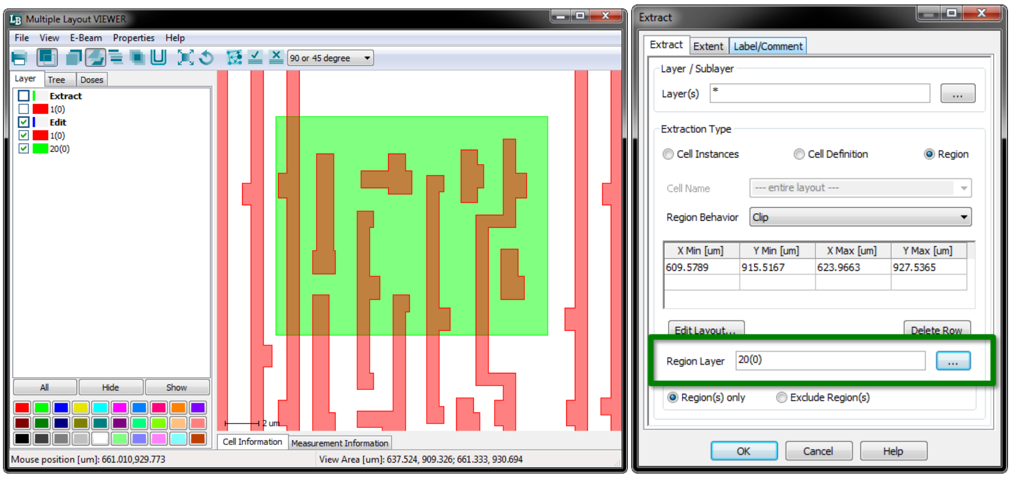
X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
609.5789	915.5167	623.9663	927.5365

Region Layer:

● Region(s) only ○ Exclude Region(s)

Mouse position [um]: 606.004, View Area [um]: 601.912, 903

As mentioned, ROIs can be defined by a layer within the pattern. This layer is referred to as a *Region Layer*. In the next example, layer 20(0), the green layer, will be used as the region layer. To use this layer as a region layer, enter the layer 20(0) in the Region Layer(s) text field.



Multiple Layout VIEWER

Layer list: ☐ Extract, ☐ 1(0), ☑ Edit, ☑ 1(0), ☑ 20(0)

Mouse position [um]: 661.010,929.773

View Area [um]: 637.524, 909.326; 661.333, 930.694

Extract

Layer(s) *

Extraction Type: ○ Cell Instances ○ Cell Definition ● Region

Cell Name: --- entire layout ---

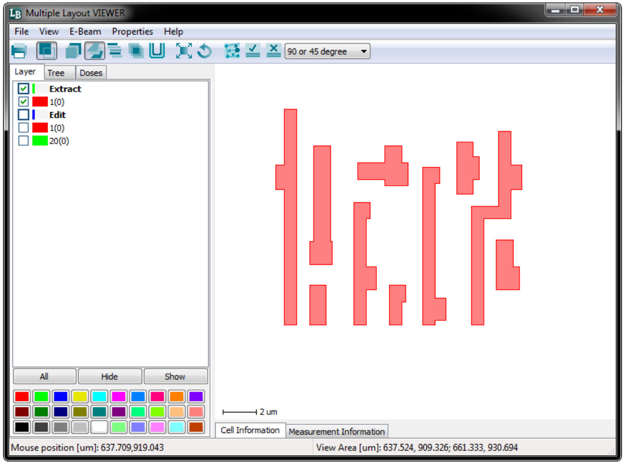
Region Behavior: Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
609.5789	915.5167	623.9663	927.5365

Region Layer: 20(0)

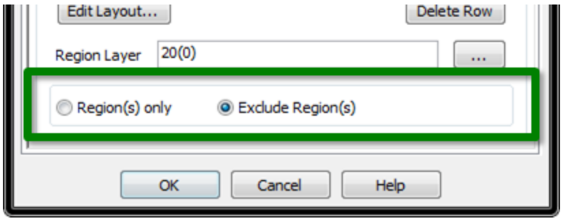
● Region(s) only ○ Exclude Region(s)

As a result, any shapes on layer 20(0) will be taken out of the pattern and their position and size will be used as bounding boxes (a.k.a. ROIs) to extract the data. Below, the region behavior is set to *Clip:*



Mouse position [um]: 637.709,919.043
View Area [um]: 637.524, 909.326; 661.333, 930.694

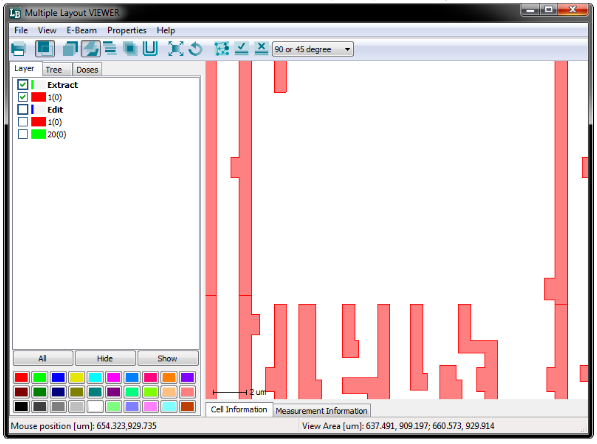
So far, the region extraction examples have been shown using the **Region(s) only** mode. The option **Exclude Region(s)** will do the opposite: it will remove the ROIs from the pattern:



Region Layer 20(0)

○ Region(s) only ● Exclude Region(s)

Using the previous example but with *Exclude Regions* selected, the result will remove the area within the ROI (in this case that is defined by layer 20):



Mouse position [um]: 654.323,929.735
View Area [um]: 637.491, 909.197; 660.573, 929.914

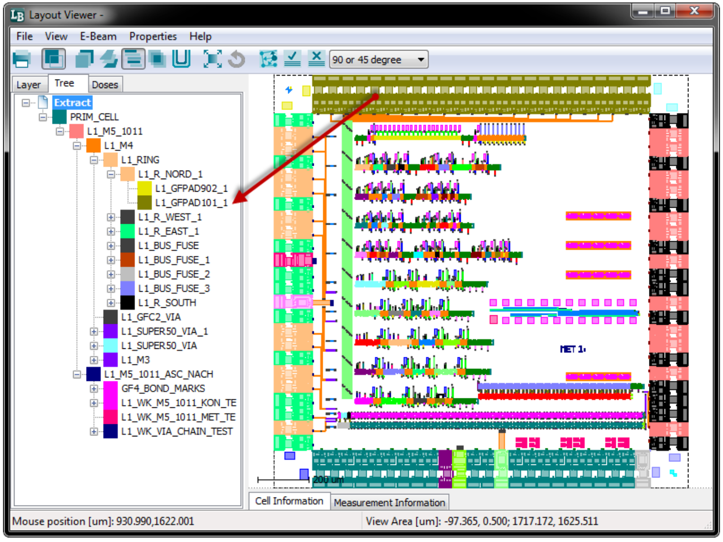
Here the region was removed using the region behavior *Clip.* Alternative region behaviors (Within and Touching) could have been used to remove the data as well.

10.1.10 How can I extract a cell from a layout?

Using the Extract module, you can quickly extract instances of cell or a cell definition. This can be accomplished by selecting either *Cell Instances* or *Cell Definition* in the Extract module. In the examples below, these two modes of Extraction are demonstrated.

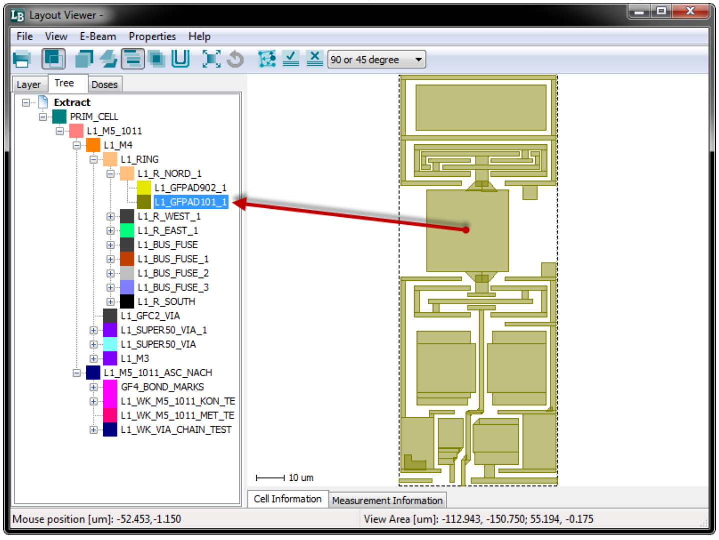
Cell Instances

When selected, the module will extract all instances of a specified cell indicated in the *Cell Name* drop down. For example, in the pattern below, there are several instances of L1_GFPAD101_1 at the top of the layout.



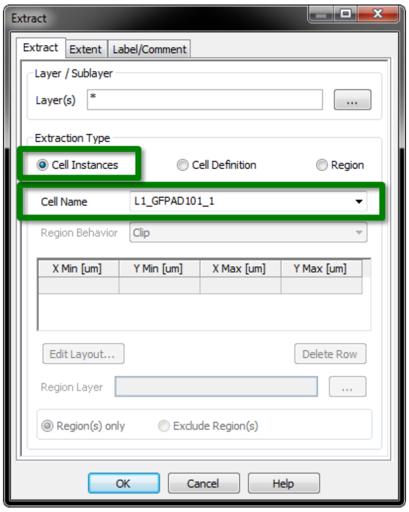
Mouse position [um]: 930.990,1622.001
View Area [um]: -97.365, 0.500; 1717.172, 1625.511

The actual cell of L1_GFPAD101_1 actually looks like the following:



Mouse position [um]: -52.453,-1.150
View Area [um]: -112.943, -150.750; 55.194, -0.175

If a user wanted to simply expose all instances of this cell, the export module configuration would have Cell Instances selected and the Cell Name set to L1_GFPAD101_1 like on the bottom left. The result will have the following layout to the bottom right.



Extract

Extract | Extent | Label/Comment

Layer / Sublayer

Layer(s) *

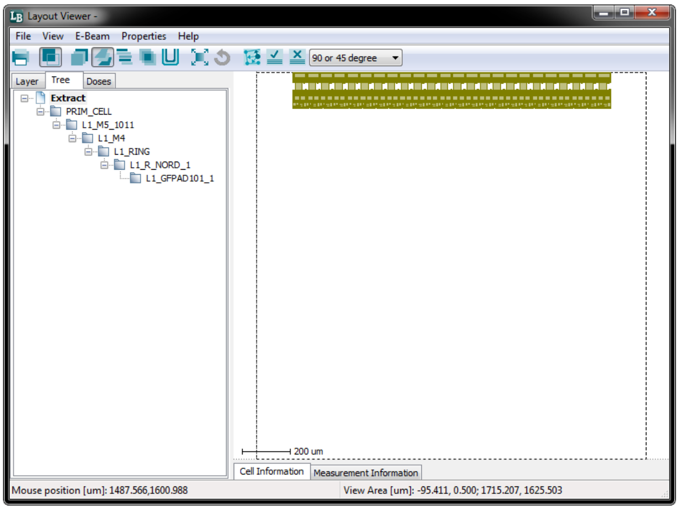
Extraction Type

● Cell Instances ○ Cell Definition ○ Region

Cell Name L1_GFPAD101_1

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]



Layout Viewer -

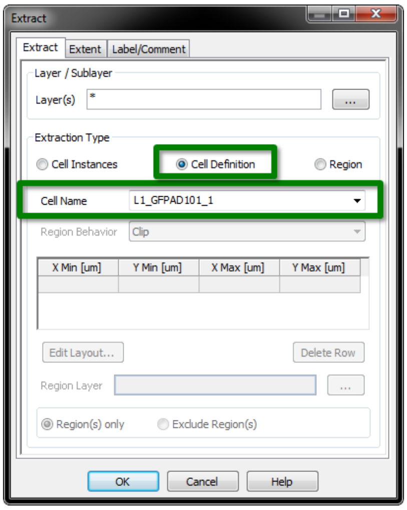
- Extract
  - PRIM_CELL
    - L1_M5_1011
      - L1_M4
        - L1_RING
          - L1_R_NORD_1
            - L1_GFPAD101_1

Mouse position [um]: 1487.566,1600.988

View Area [um]: -95.411, 0.500; 1715.207, 1625.503

Cell Definition

When selected, the module will extract only the design cell selected from the *Cell Name* drop down. For example, if the Extraction Type was set to Cell Definition and the Cell Name was set to L1_GFPAD101_1 (seen below left), only the cell would be extracted (below right).



Extract

Extract | Extent | Label/Comment

Layer / Sublayer

Layer(s) *

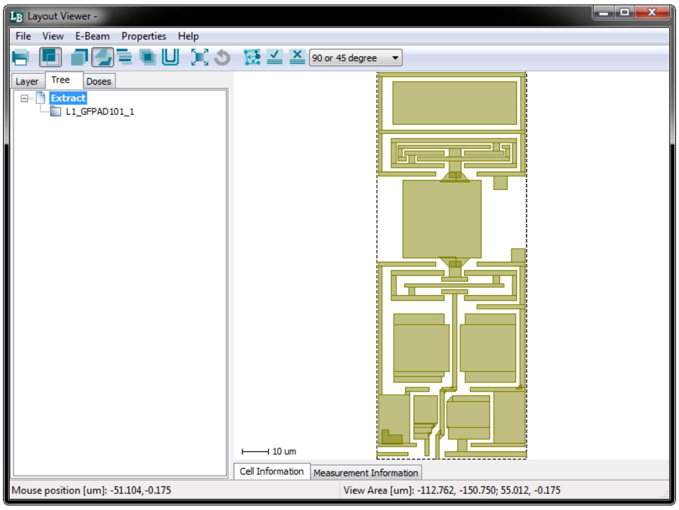
Extraction Type

○ Cell Instances ● Cell Definition ○ Region

Cell Name L1_GFPAD101_1

Region Behavior Clip

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]



Layout Viewer -

- Extract
  - L1_GFPAD101_1

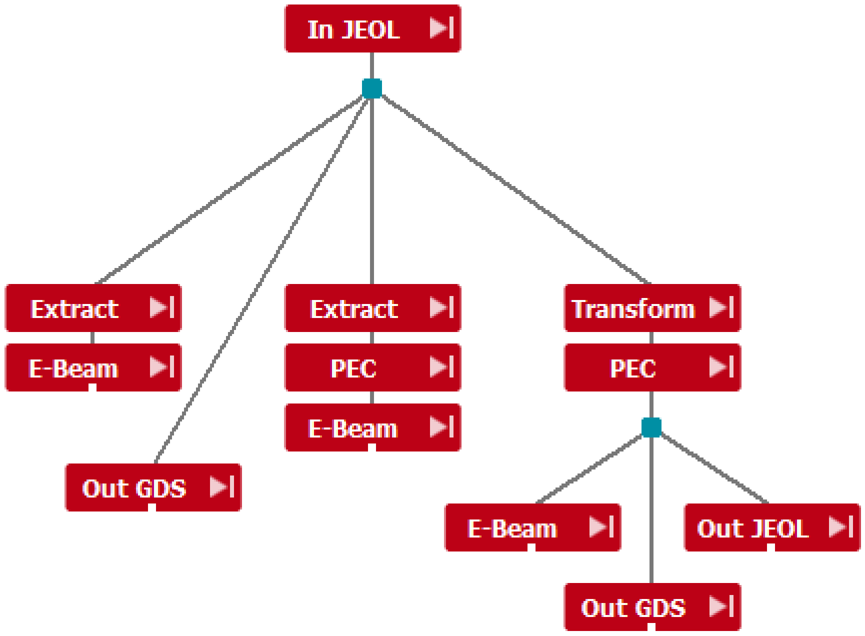
Mouse position [um]: -51.104,-0.175

View Area [um]: -112.762, -150.750; 55.012, -0.175

10.1.11 How does the Splitter module work?

The Splitter module facilitates data preparation in BEAMER and is found in the control library.

The Split module is used to make the previous module's data available to multiple modules. In other words, the data flow can branch into many modules.



Here an imported JEOL file has different regions extracted for e-beam simulation, with and without PEC. Additionally, it is completely exported to a GDSII format file. Another region is rotated (Transform) before PEC treatment, then simulated and exported to GDSII and JEOL output files.

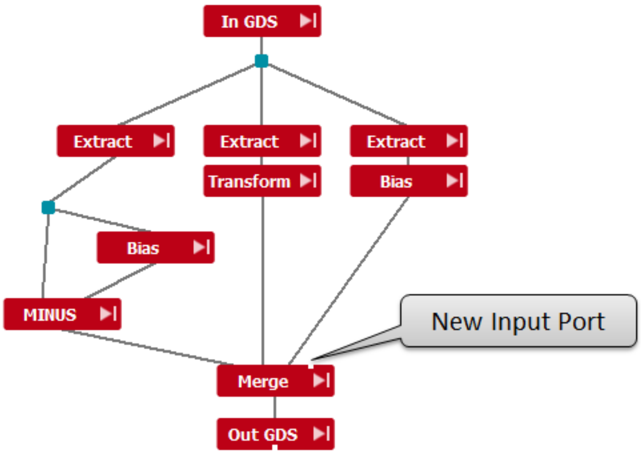
10.1.12 How does the Merge module work and how is it different from a Boolean OR?

The Merge module facilitates data preparation in BEAMER and is found as layout operation module.

The Merge module combines the outputs of multiple modules, preserving datatype and layer information when available.

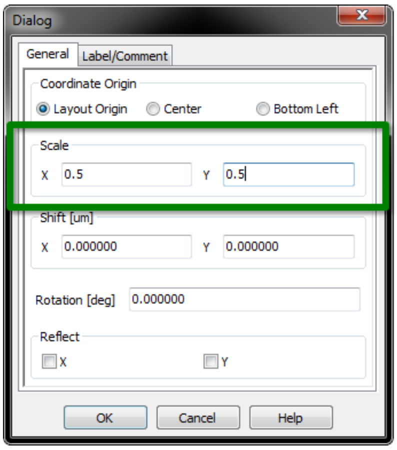
A use case would be the example below, where one input GDSII file has three regions extracted and processed, and then merged back together and exported as a new GDSII file. The blue squares are the Split modules.

Note: Each time a Merge module is connected to another module a new, empty connector is added on the right.



10.1.13 How can I scale a layout and perform general transformations?

The scaling and general transformation of data is done with the Transform module. Using the Scale X and Y parameters, a pattern can be scale up or down. After importing a layout, connect a Transform module and the dialog box will appear.


Dialog

General | Label/Comment

Coordinate Origin: ● Layout Origin ○ Center ○ Bottom Left

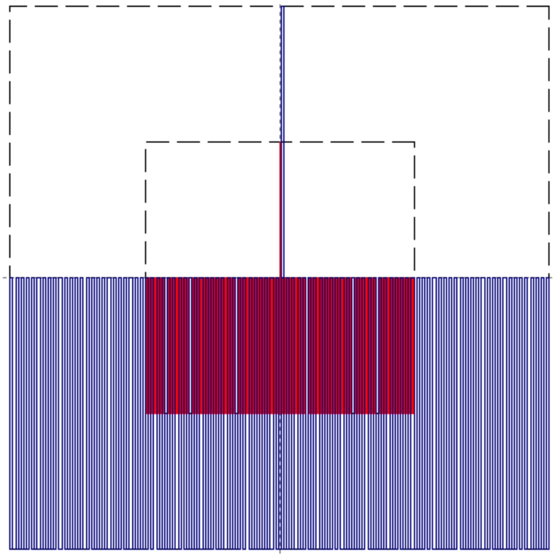
Scale: X 0.5 Y 0.5

Shift [um]: X 0.000000 Y 0.000000

Rotation [deg] 0.000000

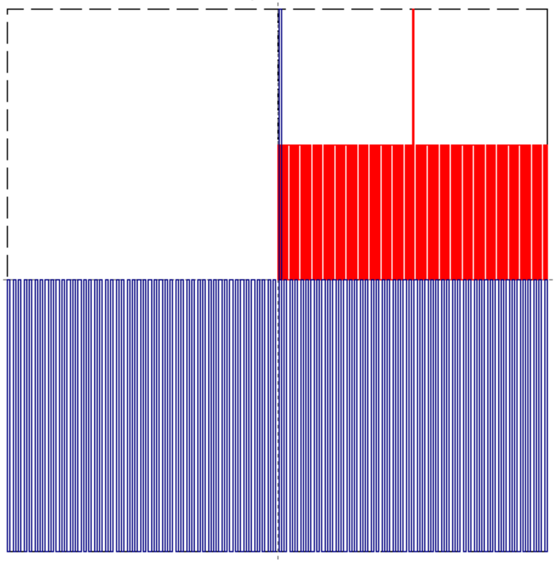
Reflect: □ X □ Y

The radio buttons at the top control the origin of the output from the Transform module. The settings shown maintain the coordinate system of the original layout (blue) and apply a scale factor of 50%. This will lead to the result shown below viewed with the Multi Layout View:



Because the origin of the original layout is its center, applying the same scale but selecting "Center" for Coordinate Origin will produce the same results.

Choosing "Bottom Left" will produce the following:



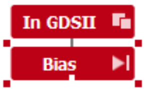
The origin of the scaled layout now corresponds with the origin of the original layout, resulting in an observed shift to the upper right.

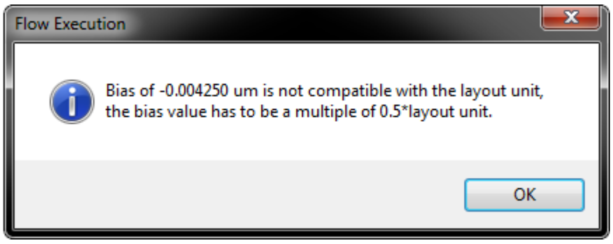
10.1.14 How can I adjust the grid of a layout?

Layout design tools set a base unit, referred to as the grid, to be used during the original drawing of the data. Usually this is one nanometer. All vertices must be on multiples this grid, and one should examine the BEAMER log file "on grid" report if fracturing (Export) to another grid (resolution).

Additionally, no vertices may be on a smaller grid, and this condition can limit a user's processing requirements. The Grid module is able to modify the grid to a finer, or more coarse, unit.

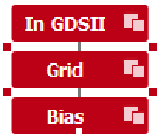
Here is an example of a Bias module asking for a unit of sizing that is not a multiple of the design grid.

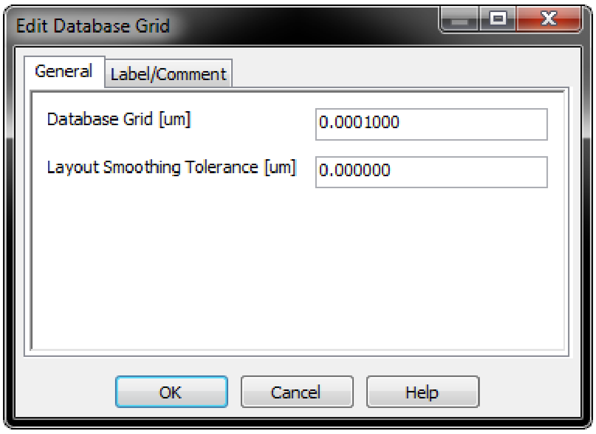



Flow Execution

Bias of -0.004250 um is not compatible with the layout unit, the bias value has to be a multiple of 0.5*layout unit.

In this case, a grid modification is needed to allow further processing.




Edit Database Grid

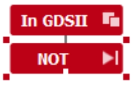
Database Grid [um] 0.0001000

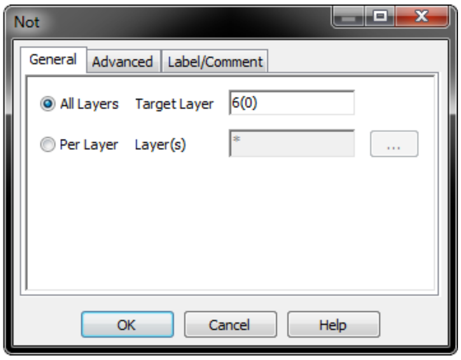
Layout Smoothing Tolerance [um] 0.000000

When the Grid module is dropped in, the dialog box opens and the database unit can be set to 0.1nm, which allows the Bias module to run.

10.1.15 How do I reverse tone a layout?

Tone reversal of a layout inverts the layout making the non-data areas data (polygons), and the data non-data.

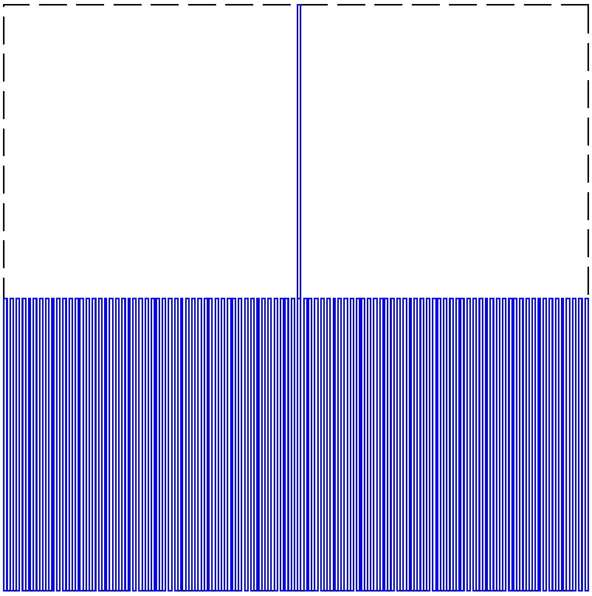




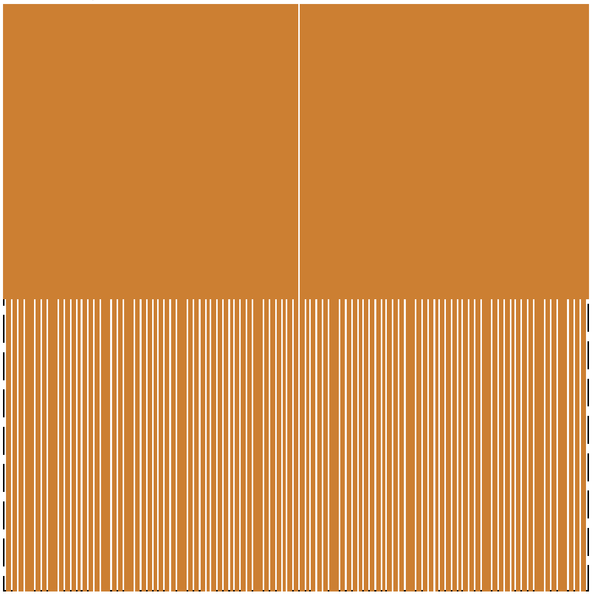
● All Layers Target Layer 6(0)

○ Per Layer Layer(s) * ...

Using the original Finger Structure layout, the pattern is as follows:



The NOT module inverts this layout.

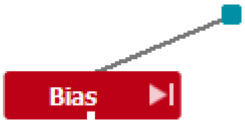


Note: the NOT module reverse tones the pattern up to the extent of the layout.

10.1.16 How can I save a flow as a new module in the BEAMER database?

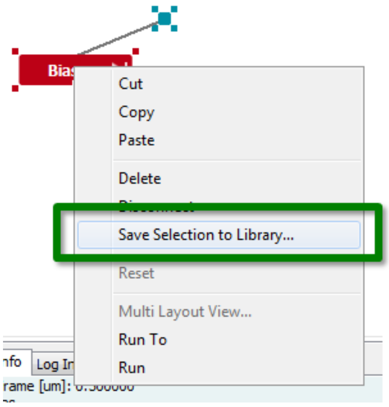
The Save Selection to Library option is intended to make repetitive work easily available by generating a template.

Any part of a flow or an entire flow can be saved using this method. To demonstrate this here is a simple example:

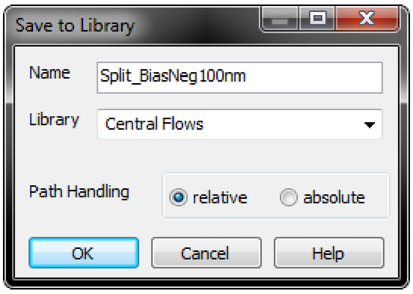


The input is branched and on one side biased by -0.1µm. Any other connection is the original input.

To save this as a database entry first select both elements, and then right click on one. A menu dialog box will appear, showing the option to “Save Selection to Library”.



Here you may specify a unique name for this new module. The Library can be the 'Central Flows' library, with resides physically in the installation directory, or the 'User Flow' directory which resides in the user’s home directory, or a new name may be entered, which will create a new directory in the user’s home directory. The difference between these is that the 'Central Flows' location makes the flow available to any user on the computer, while the 'User Flow' location is only available to the user currently logged in. Should files be loaded within the saved flow, these can be referenced either absolute or relative, which makes the migration of flows easier.



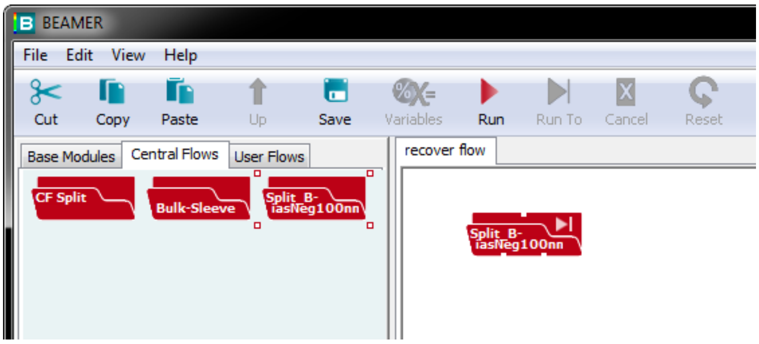
Save to Library

Name: Split_BiasNeg100nm

Library: Central Flows

Path Handling: ● relative ○ absolute

This saved flow can now be found in the Central Flow and be used like any other module. It has a single input port and two output ports The left one is the biased input, the right one is the original input.

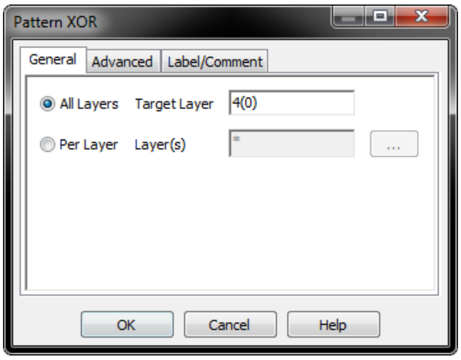


To learn more about Database modules go to the section on Database Modules.

10.1.17 How can I use the P-XOR?

The P-XOR (Pattern-XOR) is a special Boolean operation that removes even overlaps and keeps odd overlaps.

The target layer and datatype in the result can be specified by opening the parameter window by double clicking the module. By default, the target layer and datatype is 4(0). To avoid unnecessary fracturing of the original layout a soft frame can be defined in the parameter window.

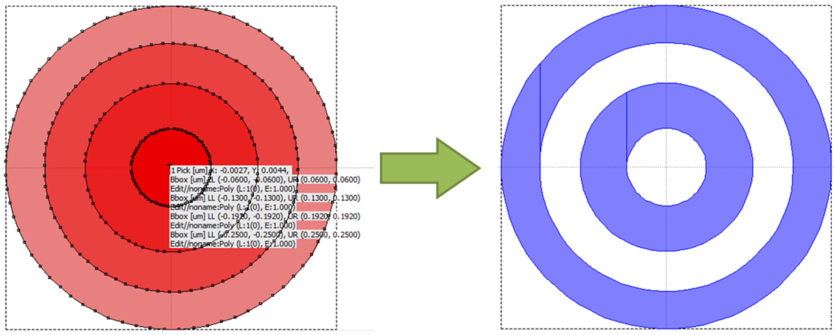

Pattern XOR

General | Advanced | Label/Comment

● All Layers Target Layer 4(0)

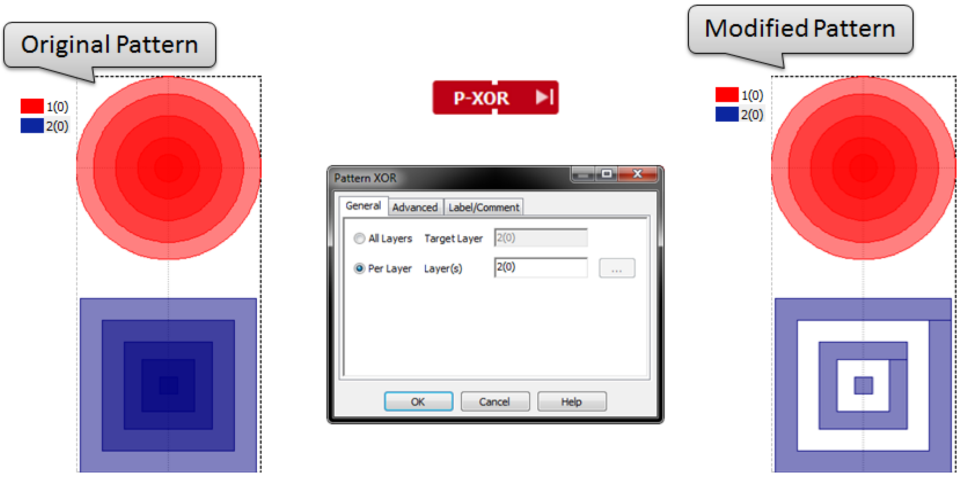
○ Per Layer Layer(s) *

In the example below, a pattern of concentric rings (blue) can be quickly generated from a pattern with concentrically overlapped circles (red).



Areas with an odd (1,3,5,7…) number of polygons overlaying will remain. Areas with an even number of polygons (2,4,6,8…) will be removed. This special function is very useful to easily generate circular gratings like zone plates.

Layerwise operation is also supported in this module. Below in the example, the original layout contains two layers, 1(0) and 2(0). With Per Layer selected, layer 2(0) is entered as the layer to apply a P-XOR. In the resulting modified pattern, the original layers are maintained and only the pattern in layer 2(0) has the P-XOR applied.



Pattern XOR

General | Advanced | Label/Comment

○ All Layers Target Layer 2(0)

● Per Layer Layer(s) 2(0)

10.1.18 How do Conventional, LRFT and Curved fracturing modes differ?

Three fracturing modes available in BEAMER are:

Conventional

The conventional algorithm performs mainly horizontal fractures at polygon vertices that maintain the original design intent as good as possible. This leads to many slices of trapezoids to best represent the original layout.

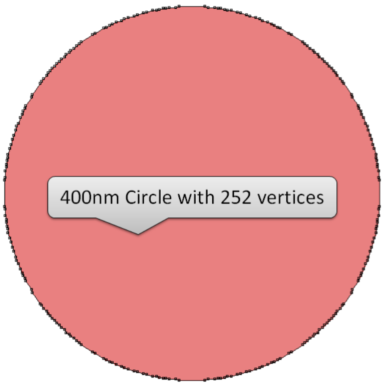
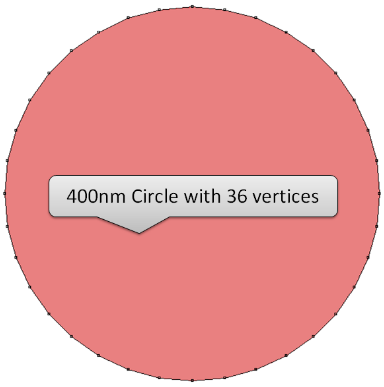
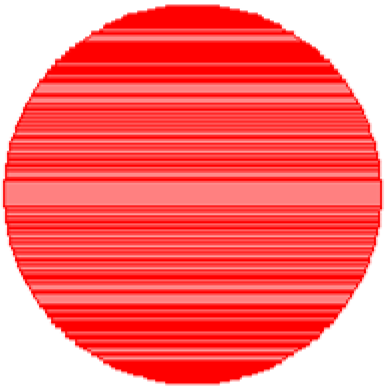
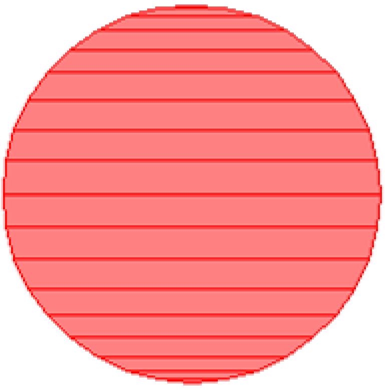
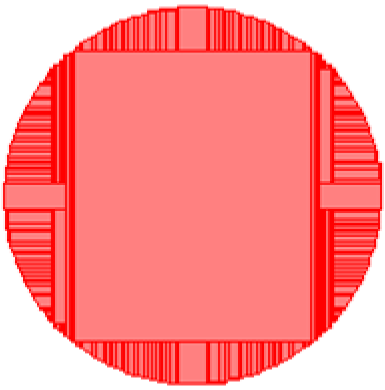
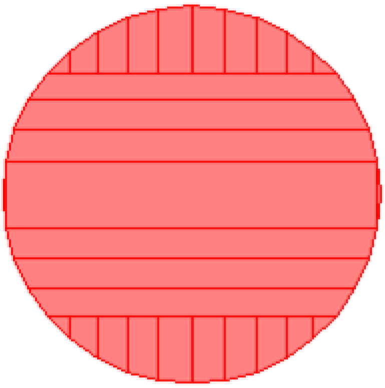
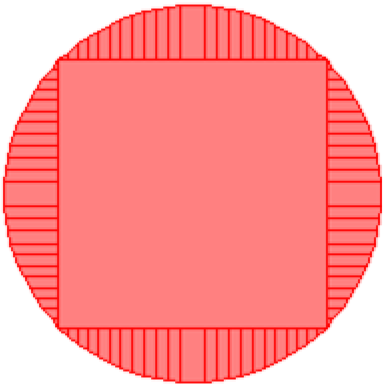
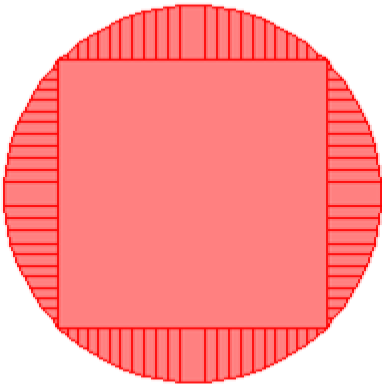
LRFT (Large Rectangle Fine Trapezoid)

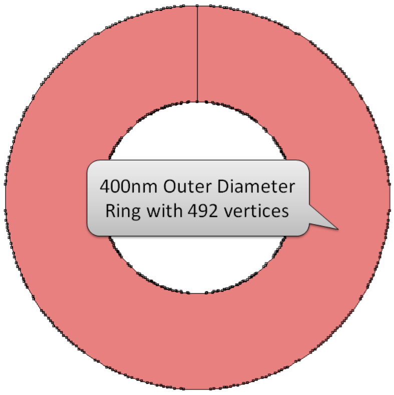
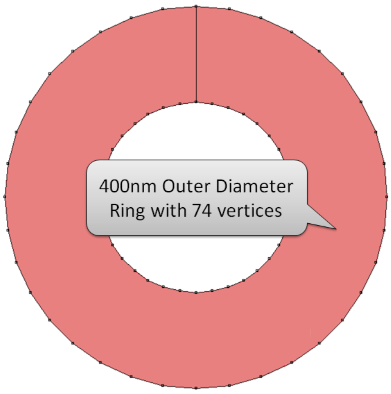
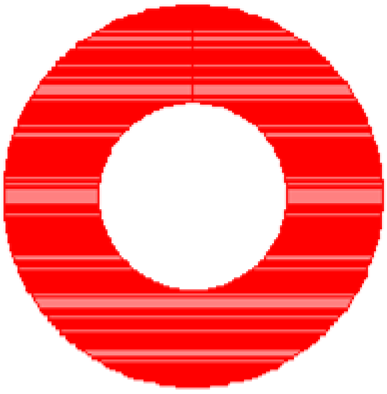
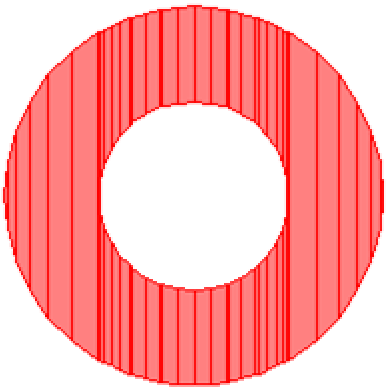
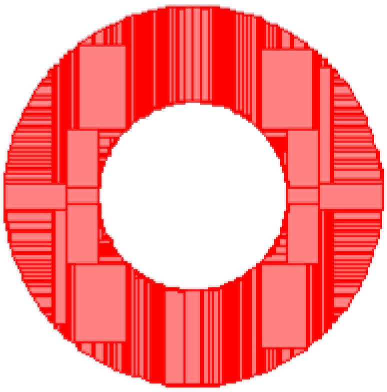
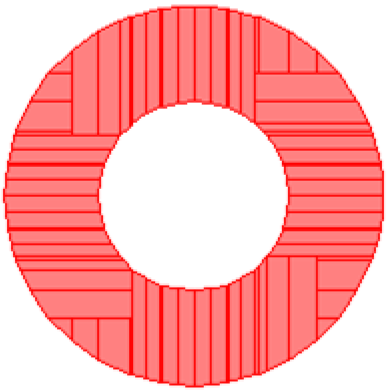
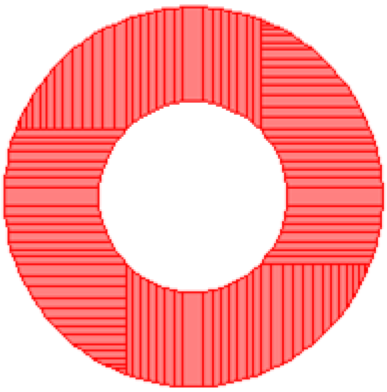
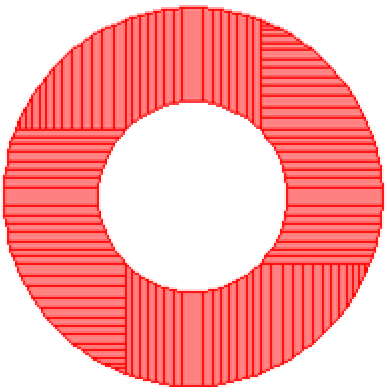
The LRFT algorithm attempts to minimize the number of shapes using very large rectangles for large areas and fine trapezoids for smaller features. In other words, this mode fractures the polygons with as many large rectangles as possible and completes the filling with small trapezoids.

Curved

This algorithm should typically be used to fracture circles and rings. This mode allows BEAMER to shift the vertices along curves to get fractures consistent with the specified resolution and beam step size. In other words, this mode will try to detect curves in the given polygons and fracture these curves within a specified tolerance consistent with the specified resolution and beam step size.

It is always necessary to inspect the fracturing, since every mode may yield a dramatically different result. The fracture mode default is set based on the tool manufacturers recommendation.
The fracturing of a 400nm diameter circle and a 400nm outer diameter ring is performed with each of the fracturing modes. This example demonstrates the advantage of **Curved** fracturing for uniform fracturing of circles and rings independent of the number and position of vertices.

	The 400nm Diameter Circle	
		
Conventional		
LRFT		
Curved		

	400nm Outer Diameter Ring	
		
Conventional		
LRFT		
Curved		

10.1.19 What fracturing mode should I use?

The fracturing mode you use depends on the pattern you are fracturing. The three fracturing modes are quickly defined below.

Three fracturing modes available in BEAMER are:

Conventional

The conventional algorithm performs mainly horizontal fractures at polygon vertices that maintain the original design intent as good as possible. This leads to many slices of trapezoids to best represent the original layout.

LRFT (Large Rectangle Fine Trapezoid)

The LRFT algorithm attempts to minimize the number of shapes using very large rectangles for large areas and fine trapezoids for smaller features. In other words, this mode fractures the polygons with as many large rectangles as possible and completes the filling with small trapezoids.

Curved

This algorithm should typically be used to fracture circles and rings. This mode allows BEAMER to shift the vertices along curves to get fractures consistent with the specified resolution and beam step size. In other words, this mode will try to detect curves in the given polygons and fracture these curves within a specified tolerance consistent with the specified resolution and beam step size.

It is always necessary to inspect the fracturing, since every mode may yield a dramatically different result. The fracture mode default is set based on the tool manufacturers recommendation.

If you have a pattern with whole circles and rings, use Curved fracturing. You can use Curved fracturing on patterns that have other shapes in addition to the circles and rings. For those shapes, LRFT will be used to fracture the pattern.

If you have a pattern with no circles and rings, but with curved features, LRFT can be used to fracture your pattern. Curved features often have too many vertices along the approximated curve, so it is advised to reduce the number of vertices using the Grid module so that the pattern is not over fractured. By reducing the number of vertices, the shape overhead is reduced when fracturing the pattern (see How can I reduce the number vertices along a curve that produces an over fracture?).

Conventional fracturing can be used if the pattern is mainly rectilinear and both Curved and LRFT produce undesired fractures. Caution should be exercised with this algorithm as it has a tendency to over fracture. Always inspect the pattern to ensure the fracture meets your requirements.

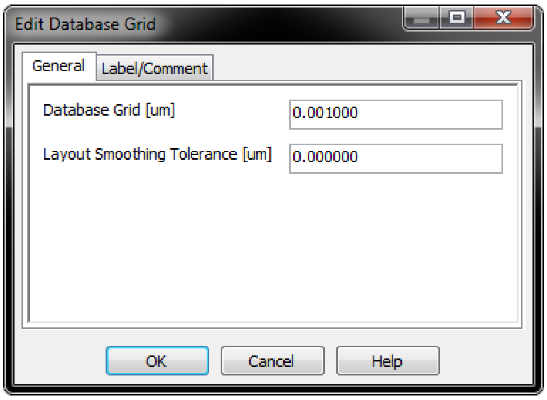
10.1.20 How can I reduce the number vertices along a curve that produces an over fracture?

Use the Grid module to reduce the number of vertices along a curve. The Grid module is known to set a database grid of a pattern to a new grid size, but it is also known for its layout smoothing capabilities. Below is a brief description of the Grid module and its two functionalities with an example of its Layout Smoothing feature.

The Grid module modifies the database unit of a pattern and provides the capability to reduce the number of vertices for curved structures.

Database Grid [um]

The database grid is the design unit to which vertices snap. The default value is 1nm. If a coarser value is used, snapping may occur in the vertices of the pattern do not sit on a grid that is an integer multiple of the coarser grid.



Edit Database Grid

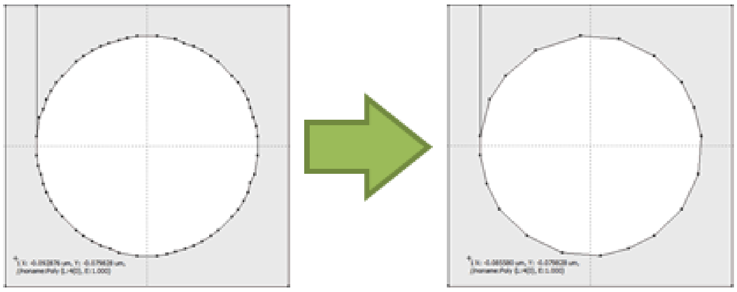
General | Label/Comment

Database Grid [um]: 0.001000

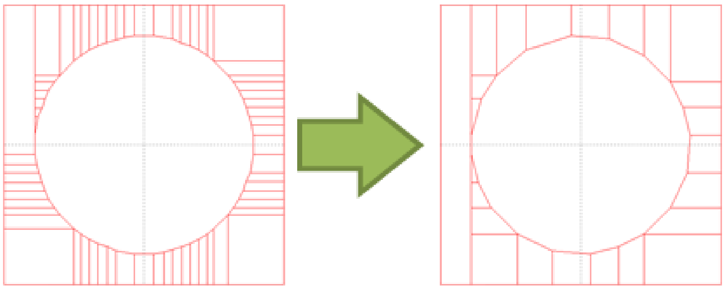
Layout Smoothing Tolerance [um]: 0.000000

Layout Smoothing Tolerance

The smoothing parameter will eliminate a vertex with a distance smaller than the smoothing value to the next vertex. In other words, if a value of 0.005um is entered as the Layout Smoothing Tolerance, then any existing vertices that yield an approximated curve less than 0.005um will be removed, thus eliminating unnecessary vertices. By reducing the number of vertices, this will significantly reduce fracturing time. An example of layout smoothing tolerance is below. To the left, the image can be seen with several vertices that make up the hole in the square pattern. When applying layout smoothing, the number of vertices is reduced significantly. It is advised to always test and inspect different Layout Smoothing Tolerance values to be sure it meets your fracturing needs.



A direct result of the reduced number of vertices is the reduced number of shapes. In other words, the number of fractured shapes is reduced since the number of vertices is reduced. Reducing the number of shapes can reduce data volume and therefore any possible shape overhead delay for the tool.



10.1.21 What is FDA?

For module definition see FDA. For examples, see Assigning a Dose by Layer and Dose Matrix_Creation.

10.1.22 How do I convert a CINC file to FTXT file?

BEAMER supports the import of CINC files by converting a CINC into a FTXT flow.

The definition of variables in the CINC must be covered by the appropriate definition of these variables within the environment.

This is a list of the recognized commands:

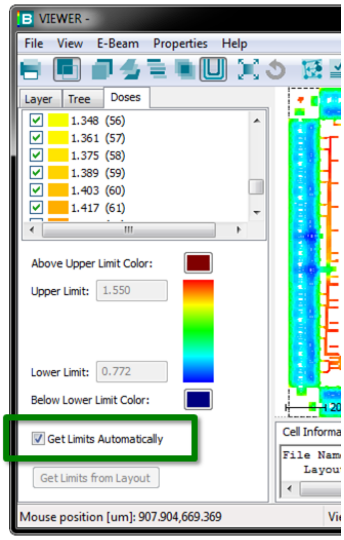
Batch	ignored
Do	ignored
Extent	Always handled as Extent w/o parameters. Parameters are ignored
Function	Only with Parameters VOID
Grow	
Hold	
Include	Only with file name. Environment variables allowed. Other parameters not yet supported
Input	Only with file name. Environment variables allowed. Other parameters not yet supported
Layers	Parameter ALL not yet supported
Limits	Parameter VOID not yet supported
Orientation	
Output	Parameter VOID not yet supported
Resolution	Only with a real number. Other parameters not yet supported
Reverse	
Scale	
Select	Only Parameters NO supported
Sizing	
Structure	Only with structure name. Other parameters not yet supported

In the log file, a list of skipped commands and command options will be noted.

10.1.23 When viewing a PEC'd in Color by Dose, how can the dose limits be obtained automatically?

To automatically obtain the upper and lower limits when clicking **Color-by-Dose** U, enable the checkbox **Get Limits Automatically**. By default this is disabled. When enabled the Upper and Lower Limit text fields are disabled. To keep this feature enabled:

1. Click the checkbox labeled **Get Limits Automatically** to enable the feature via the Doses tab in the VIEWER (bottom left) or accessing the Dose Color dialog from the VIEWER Properties menu:



☑	1.348 (56)
☑	1.361 (57)
☑	1.375 (58)
☑	1.389 (59)
☑	1.403 (60)
☑	1.417 (61)

Above Upper Limit Color:

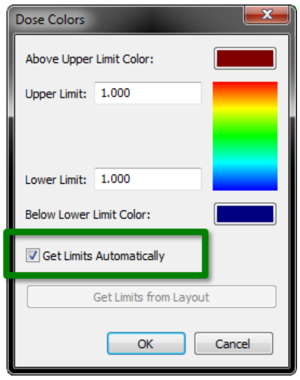
Upper Limit: 1.550

Lower Limit: 0.772

Below Lower Limit Color:

☑ Get Limits Automatically

Mouse position [um]: 907.904,669.369



Above Upper Limit Color:

Upper Limit: 1.000

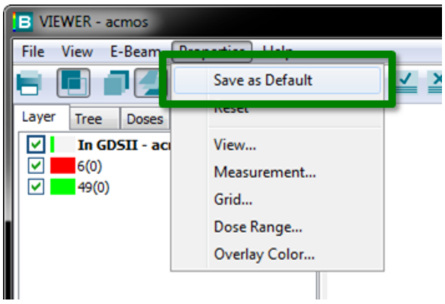
Lower Limit: 1.000

Below Lower Limit Color:

☑ Get Limits Automatically

2. Click **OK** to close the dialog

3. To save the setting, click **Save as Default** in the Properties menu



4. Close and re-open the VIEWER. Now whenever a corrected pattern is viewed and you click **Color-by-Dose**, the limits will automatically be set.

10.2 E-Beam Simulation FAQ

E-Beam simulation allows the modeling of 2D resist contours by modeling the e-beam exposure using a point spread function which describes the absorbed energy in the resist. In this FAQ, simulating resist contours is addressed in addition to more advanced functions that include automatic measurement of critical dimensions (CDs) in the pattern, looping and optimization.

10.2.1 Modeling E-beam Image Intensities

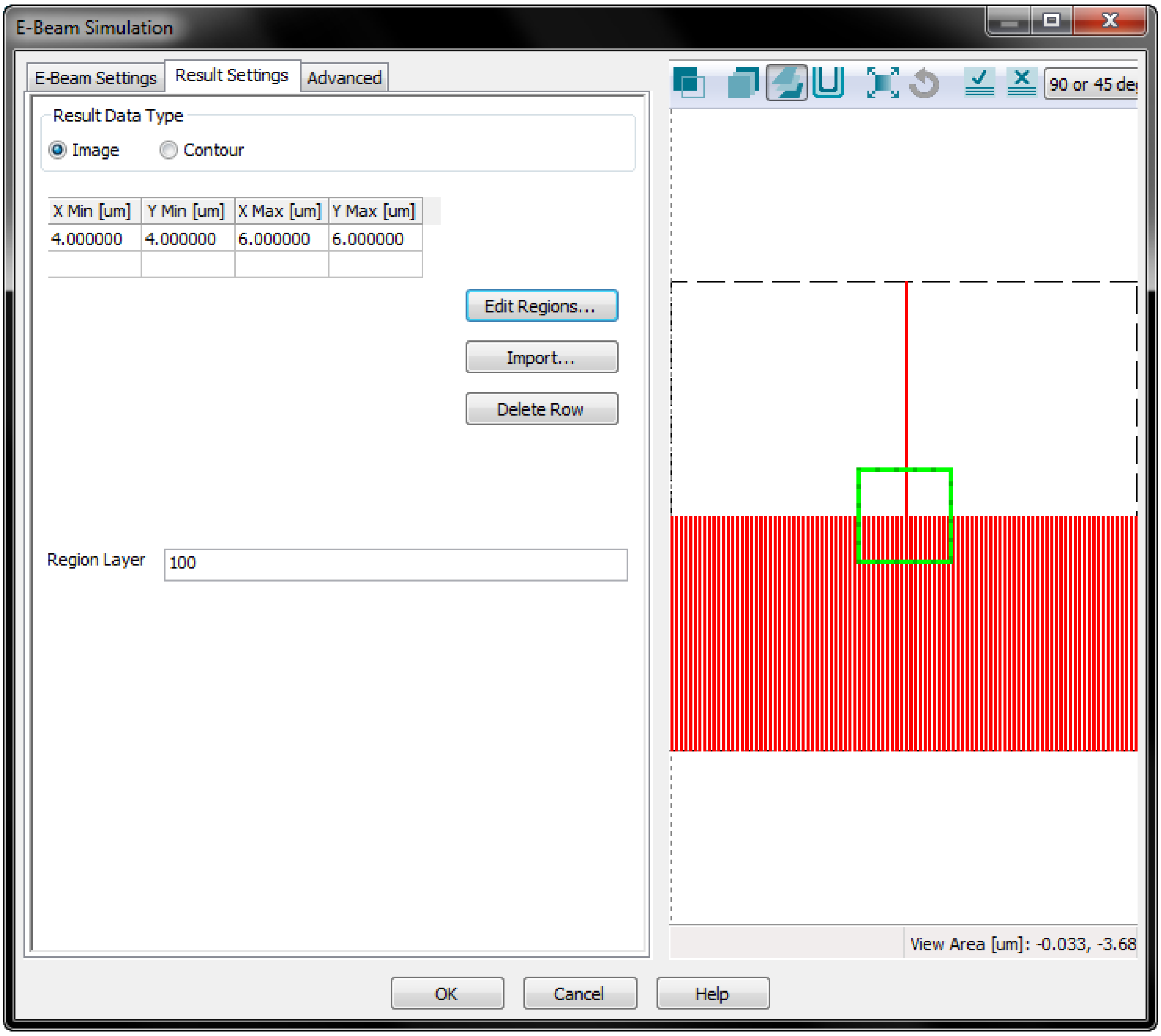
Overview:

The E-Beam module emulates the behavior of an e-beam machine by showing the energy distribution based on machine parameters and settings. This behavior is described by the parameters of the E-Beam module. The most critical are the PSF parameters. They describe how the electrons interact with the stack. The beam blur, beam step size (for shot simulation) and the writing method describe how the shots are placed in the resist.

Application:

The E-Beam image can be obtained by selection on the *Result Settings* tab the *Image* option. This will allow the definition of simulation regions for the image. The regions can be defined either using the given table or interactively by opening the layout using the Edit Regions button and pressing Shift key and drawing region boxes.

In this example, *Import* the Finger_Structure.gds and scale *Transform* it by a factor of 1/20th (0.05) down. Add an E-Beam module open its *Result Settings* tab. Activate the *Edit Regions,* and insert a region box. Adjust the settings to 4,4 and 6,6 as the start and end coordinates, respectively. On the Advanced tab, toggle the High Resolution mode, setting it to 0.005um.



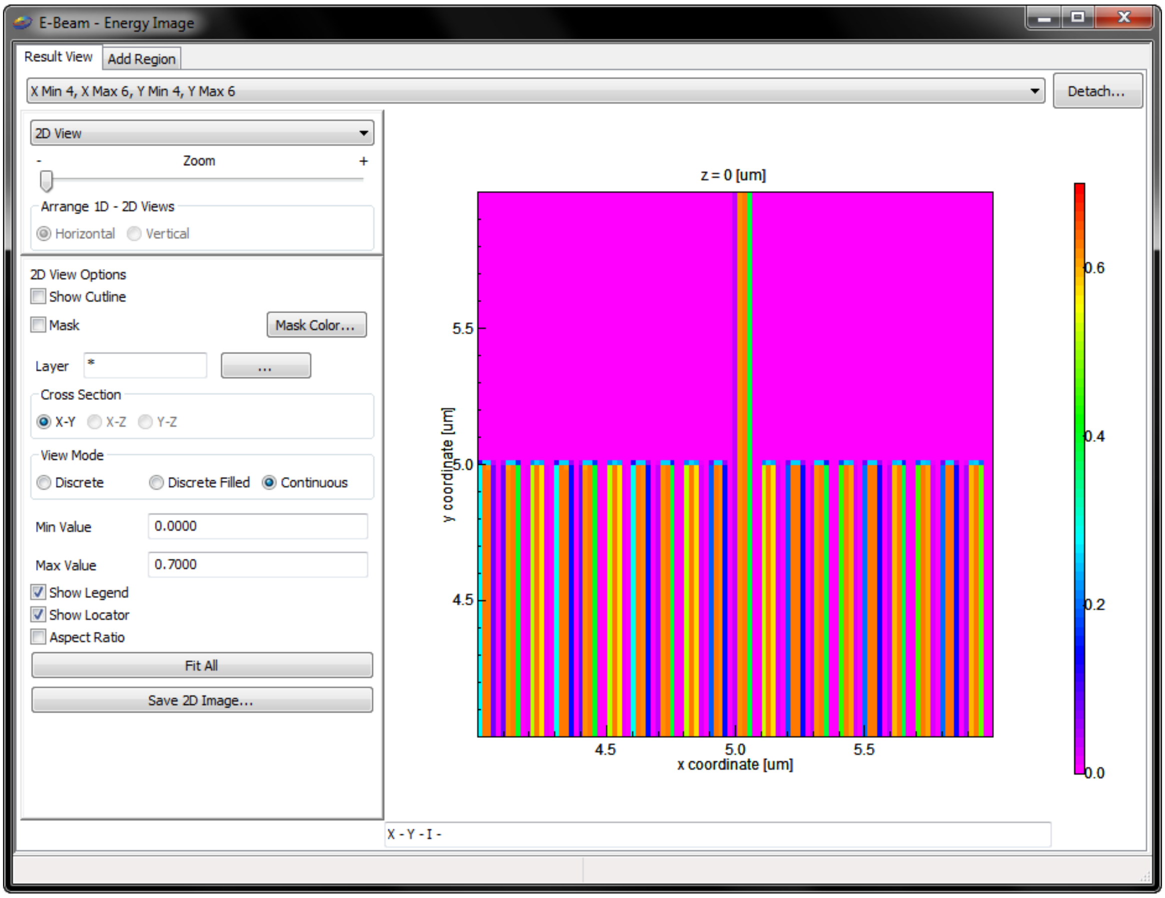
Result Data Type: ● Image ○ Contour

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
4.000000	4.000000	6.000000	6.000000

Region Layer: 100

View Area [um]: -0.033, -3.68

After execution and opening the View of the layout, you will see this image of the intensities.



E-Beam - Energy Image

Result View | Add Region

X Min 4, X Max 6, Y Min 4, Y Max 6 | Detach...

2D View

Zoom - +

Arrange 1D - 2D Views: ● Horizontal ○ Vertical

2D View Options

□ Show Cutline

□ Mask | Mask Color...

Layer: *

Cross Section: ● X-Y ○ X-Z ○ Y-Z

View Mode: ○ Discrete ○ Discrete Filled ● Continuous

Min Value: 0.0000

Max Value: 0.7000

☑ Show Legend

☑ Show Locator

□ Aspect Ratio

Fit All

Save 2D Image...

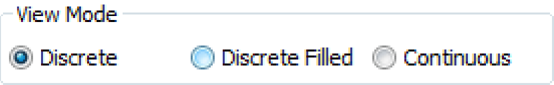
X - Y - I -

The pixelized view is intentional as it accelerates the display of the data.

First you will see the 2D view of the data in a continuous color scale. As you move the mouse into the image you will see below it the coordinates of the mouse pointer and the intensity that relate to that point.

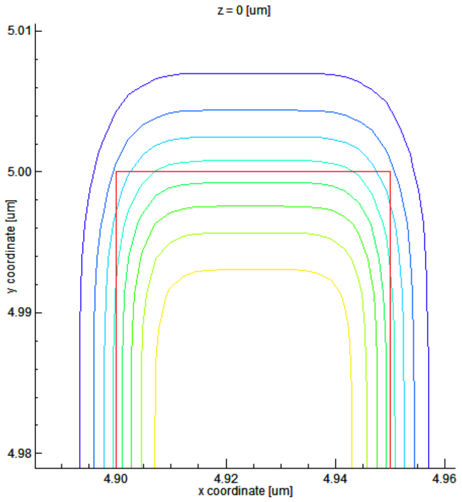
As you zoom to a point, you will get a more accurate display of the simulation data.

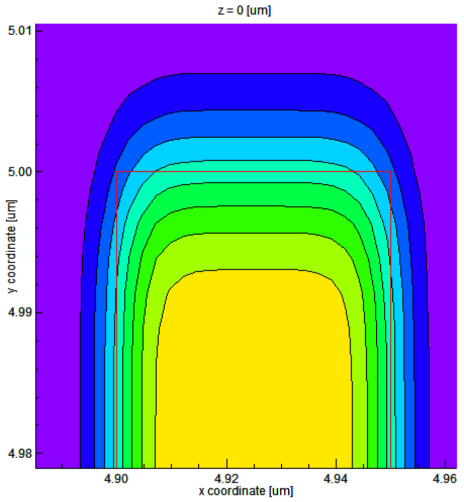
The View Mode gives you more insight to the distribution of the energy.

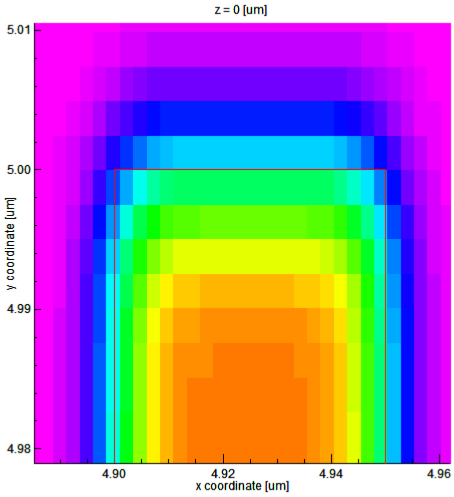


View Mode

● Discrete ○ Discrete Filled ○ Continuous



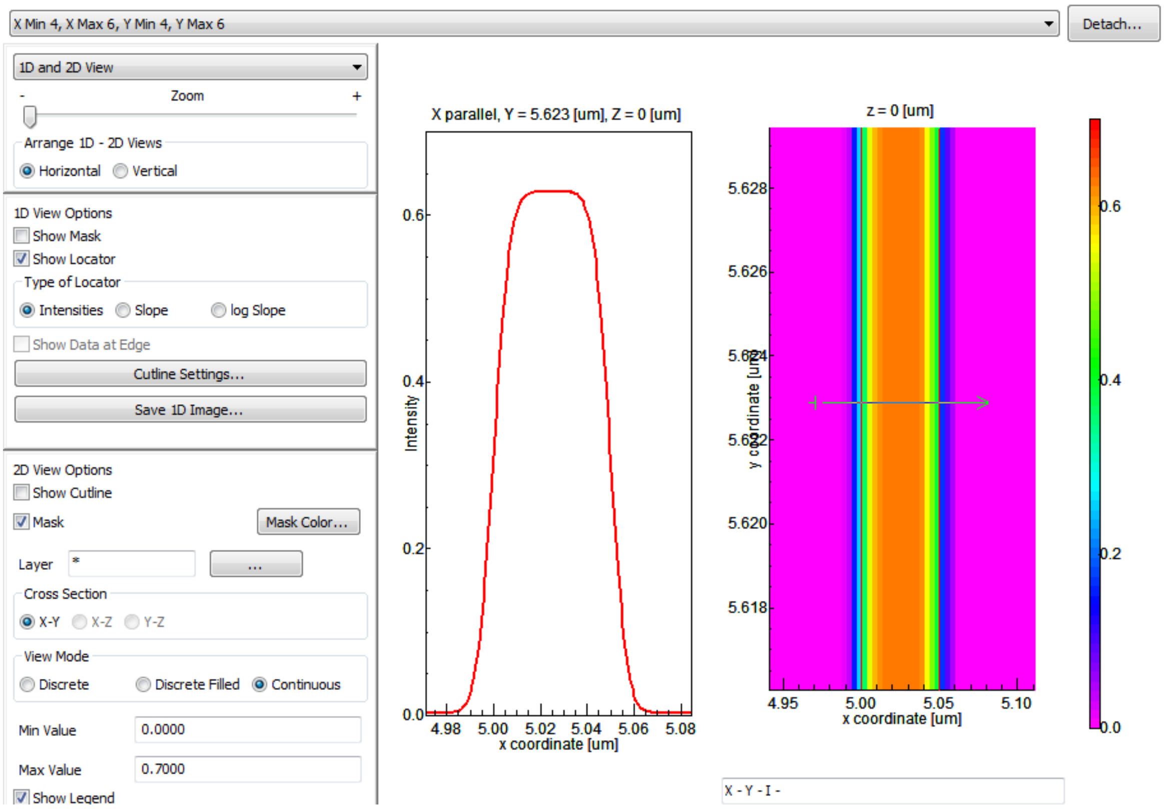




A mask display can be added by toggling the appropriate checkbox.

☑ Mask

Another display available is the 1D & 2D View. Here you can see the 2D intensities on the right hand and add a cutline using shift and left mouse button for definition. This will give you an 1D profile of under the cutline.



X Min 4, X Max 6, Y Min 4, Y Max 6

1D and 2D View

Arrange 1D - 2D Views: ● Horizontal ○ Vertical

1D View Options

□ Show Mask

☑ Show Locator

Type of Locator: ● Intensities ○ Slope ○ log Slope

□ Show Data at Edge

2D View Options

□ Show Cutline

☑ Mask

Layer: *

Cross Section: ● X-Y ○ X-Z ○ Y-Z

View Mode: ○ Discrete ○ Discrete Filled ● Continuous

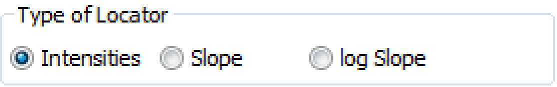
Min Value: 0.0000

Max Value: 0.7000

☑ Show Legend

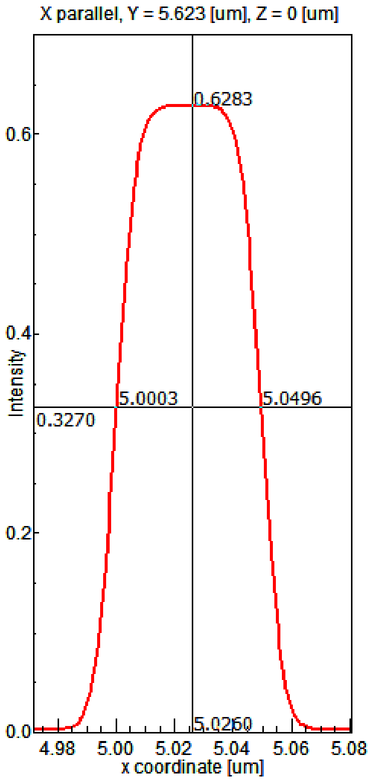
X - Y - I -

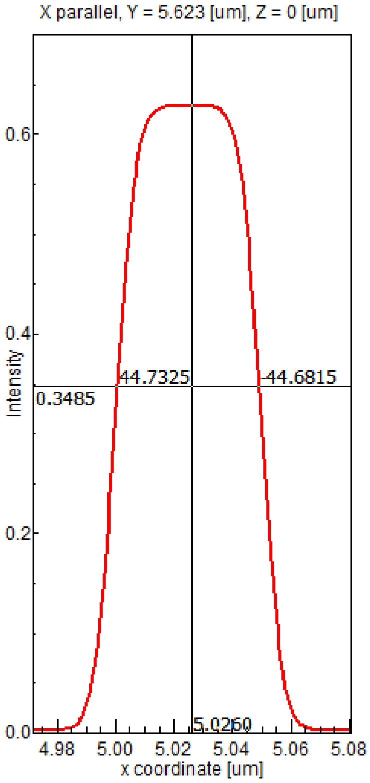
When moving the mouse into the 1D view it will provide information based on the locator type chosen.

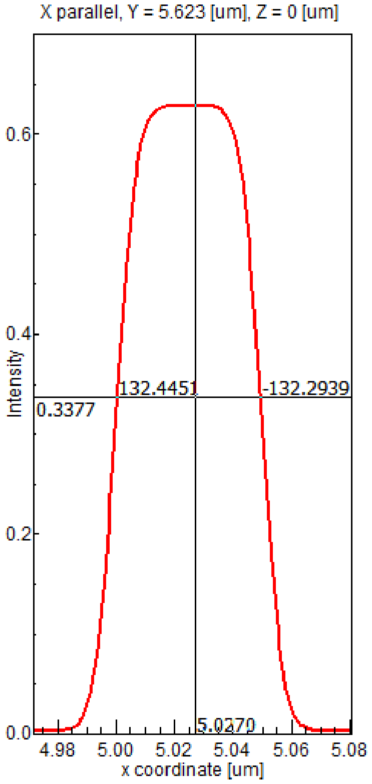


Type of Locator

● Intensities ○ Slope ○ log Slope


X parallel, Y = 5.623 [um], Z = 0 [um]
0.6283
5.0003
5.0496
0.3270
5.0260
Intensity
0.0
0.2
0.4
0.6
4.98
5.00
5.02
5.04
5.06
5.08
x coordinate [um]


X parallel, Y = 5.623 [um], Z = 0 [um]
44.7325
44.6815
0.3485
5.0260
Intensity
0.0
0.2
0.4
0.6
4.98
5.00
5.02
5.04
5.06
5.08
x coordinate [um]


X parallel, Y = 5.623 [um], Z = 0 [um]
132.4451
-132.2939
0.3377
5.0270
Intensity
0.0
0.2
0.4
0.6
4.98
5.00
5.02
5.04
5.06
5.08
x coordinate [um]

For **Type of Locator** parameter definitions, see 1D View.

These are the most basic operations and functions of the new view and show of the usefulness of this view. The E-Beam module can help you to see effects of machine parameters before the actual writing. This can help you to avoid wrong settings in advance and also to optimize the settings.

10.2.2 Bulk and Sleeve Process Correction

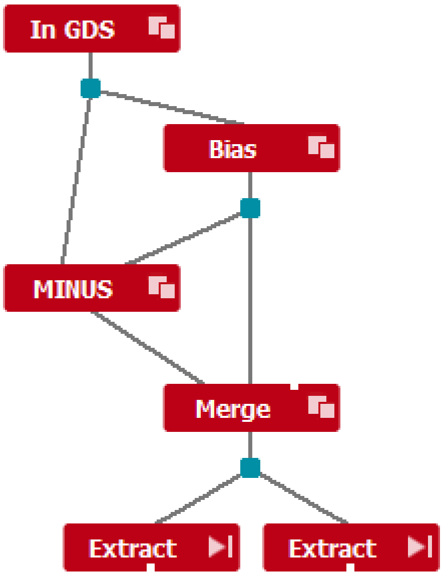
Overview:

The Bulk and Sleeve process correction finds is usage when it is very important to precisely write the outline of a structure but where the writing of the entire structure takes too much time with a high resolution. The structures that will be written are split into a sleeve or border where high precision writing is done and the bulk in the center that is written with lower precision and higher speed.

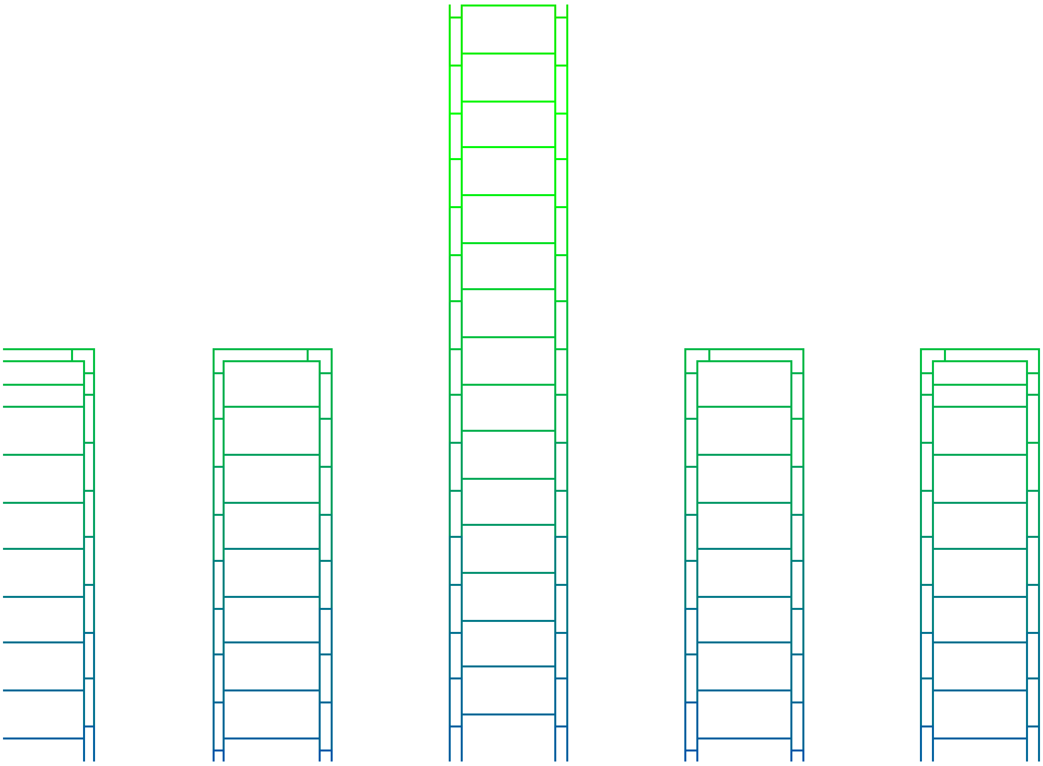
Application:

To start this application we Import the sample layout "Finger Structure.gds." To create the sleeve we need to negatively bias the original layout and subtract this from the original layout. To do this a Splitter module is added after the Import. Then add a MINUS module is added to the Splitter and a Bias module. Set the Bias Parameters to -0.1 µm and assign the target layer to 10. Add another Splitter module after the Bias and then connect the MINUS to this new Splitter. We now have two remaining output ports. The MINUS output is the border of the layout, the Splitter right hand output is the bulk of the layout.
Now we need to do a process effect correction but of course we want to do this for the entire layout. Use a Merge module and bring the two parts of the layout together again. During the Bias and the MINUS the results were assigned new layers. The Merge module will maintain the different layer information, which is also preserved throughout the PEC. After the Merge add a PEC module. PEC will now correct the entire layout by adding dose assignments.
Once the PEC is done, we need to split the corrected layout again since the sleeve is to be written with a fine beam, while the center part will be written with a coarse beam. After the PEC add another Splitter module. After that add an Extract module and enter in the layer the number 1, confirm, and then add another Extract to the Splitter module. For the layer enter 10. Now an Export is added to the sleeve and bulk parts. For guidelines on how to set the parameters here, check the Machine Format Fracturing section for the machine you have.

The complete flow would look like this:



A view of the corrected layout shows how the border and the bulk are present:



This Multi View of the sleeve and the bulk shows the dose correction for each one, and now the bulk and the sleeve can be exported into separate files for writing.

Conclusion:

In this example, you saw how a bulk and sleeve operation can be done, with process correction, and the resultant data is setup for different exports. It was done one for a simple layout to show the principle but is of course applicable to any layout.

10.2.3 Simulating Resist Contours

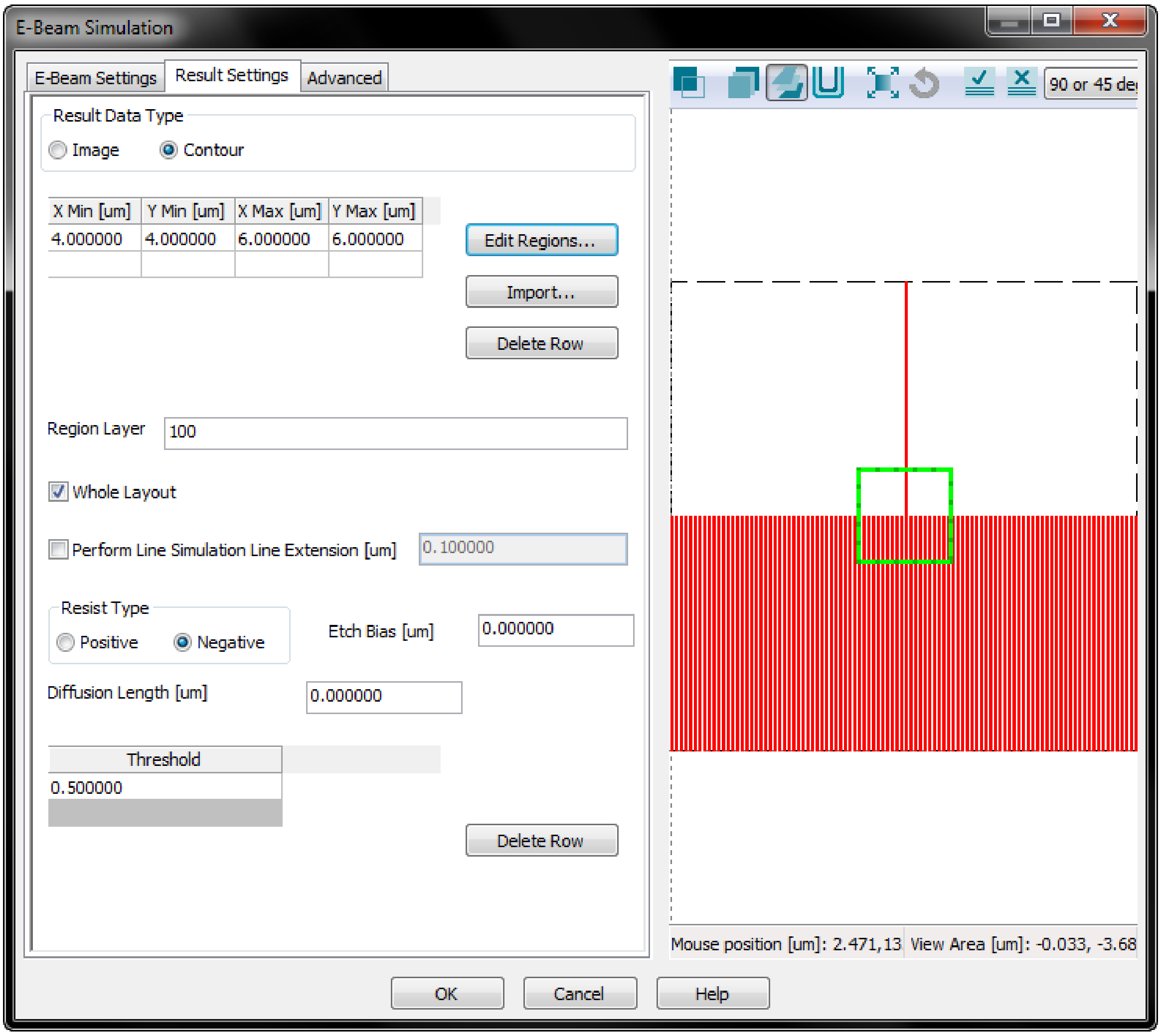
Overview:

Besides simulating the image intensity, the information of energy deposition can be taken a step further to predict the anticipated resist image. To accomplish this task, the PSF and e-beam machine parameters need to be taken into account as well as the resist model and process parameters.

Application:

To predict these effects, an e-beam energy distribution is computed and a threshold method is applied on the result. The threshold model is additionally taking diffusion effects into account as well as having the ability to add an etch bias.

The example we will work on this case is to re-use the flow from the Modeling E-beam Image Intensities chapter and switch the Result Settings to be in Contour mode. Toggle the checkbox to use selected simulation regions and confirm the settings by clicking OK.



E-Beam Simulation

Result Data Type

○ Image ● Contour

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]
4.000000	4.000000	6.000000	6.000000

Region Layer 100

☑ Whole Layout

□ Perform Line Simulation Line Extension [um] 0.100000

Resist Type

○ Positive ● Negative

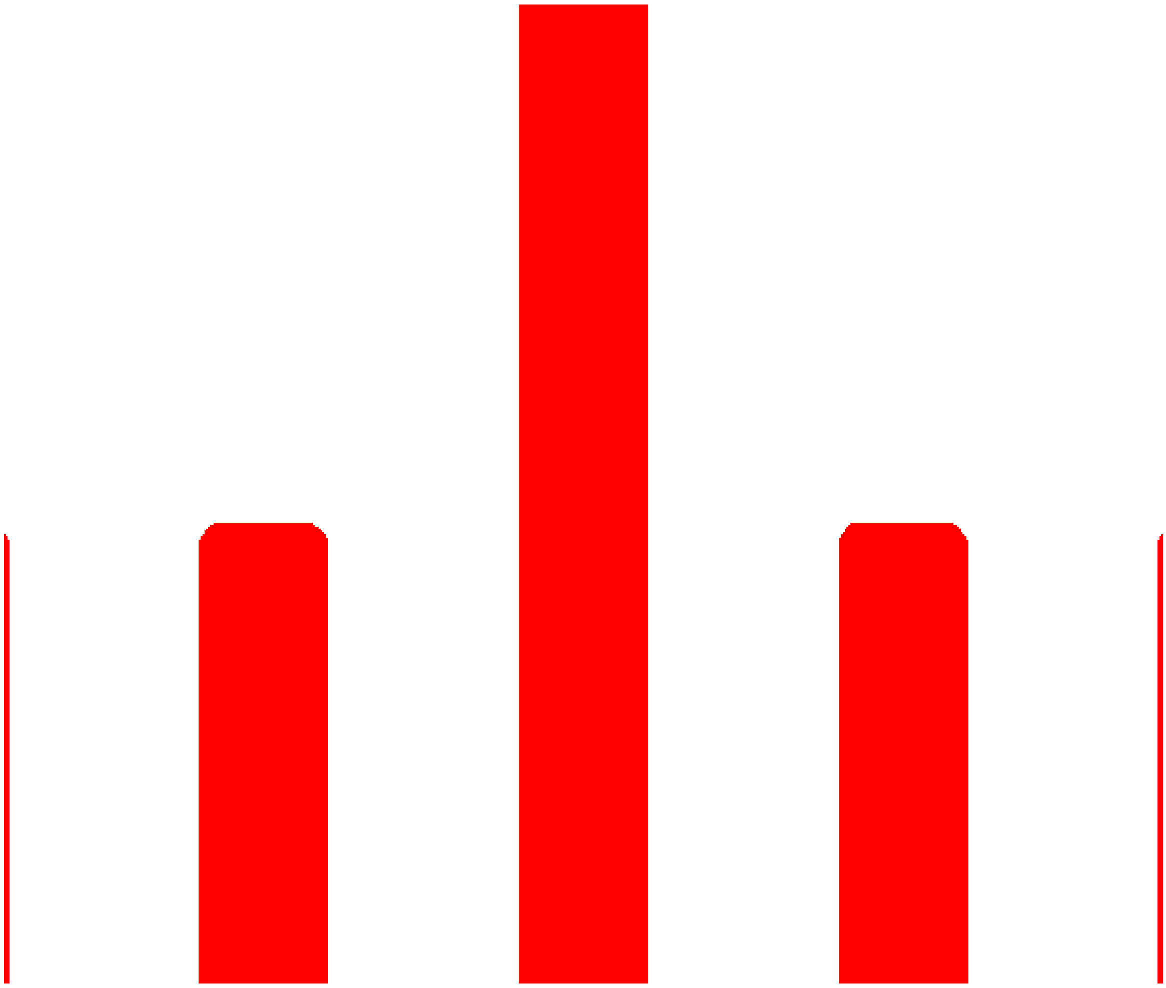
Etch Bias [um] 0.000000

Diffusion Length [um] 0.000000

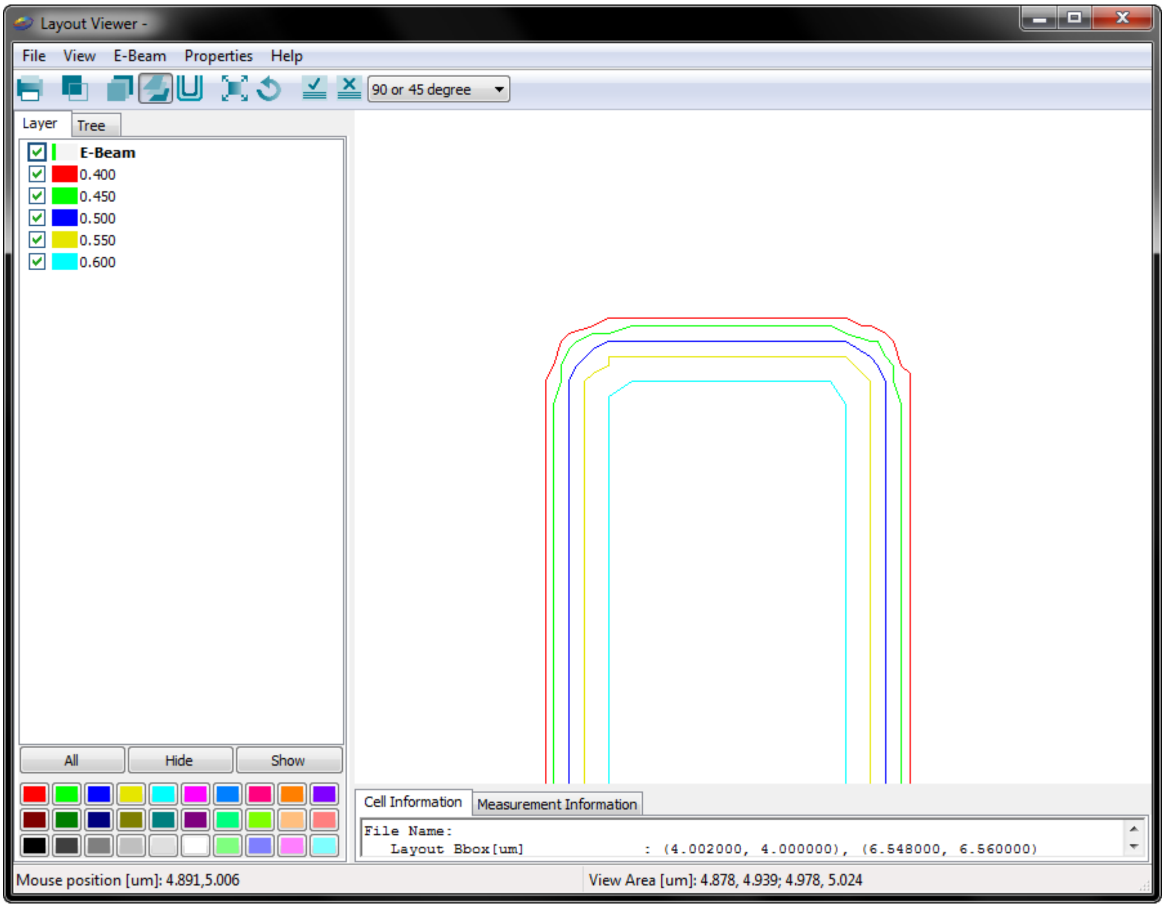
Threshold
0.500000

Mouse position [um]: 2.471,13 View Area [um]: -0.033, -3.68

After execution of the run you can open the Viewer and will get a layout of the simulated resist contour.



This results represents a single threshold. It is possible to add multiple thresholds which would represent different dose levels.



☑ E-Beam
☑ 🟥 0.400
☑ 🟩 0.450
☑ 🟦 0.500
☑ 🟨 0.550
☑ 0.600

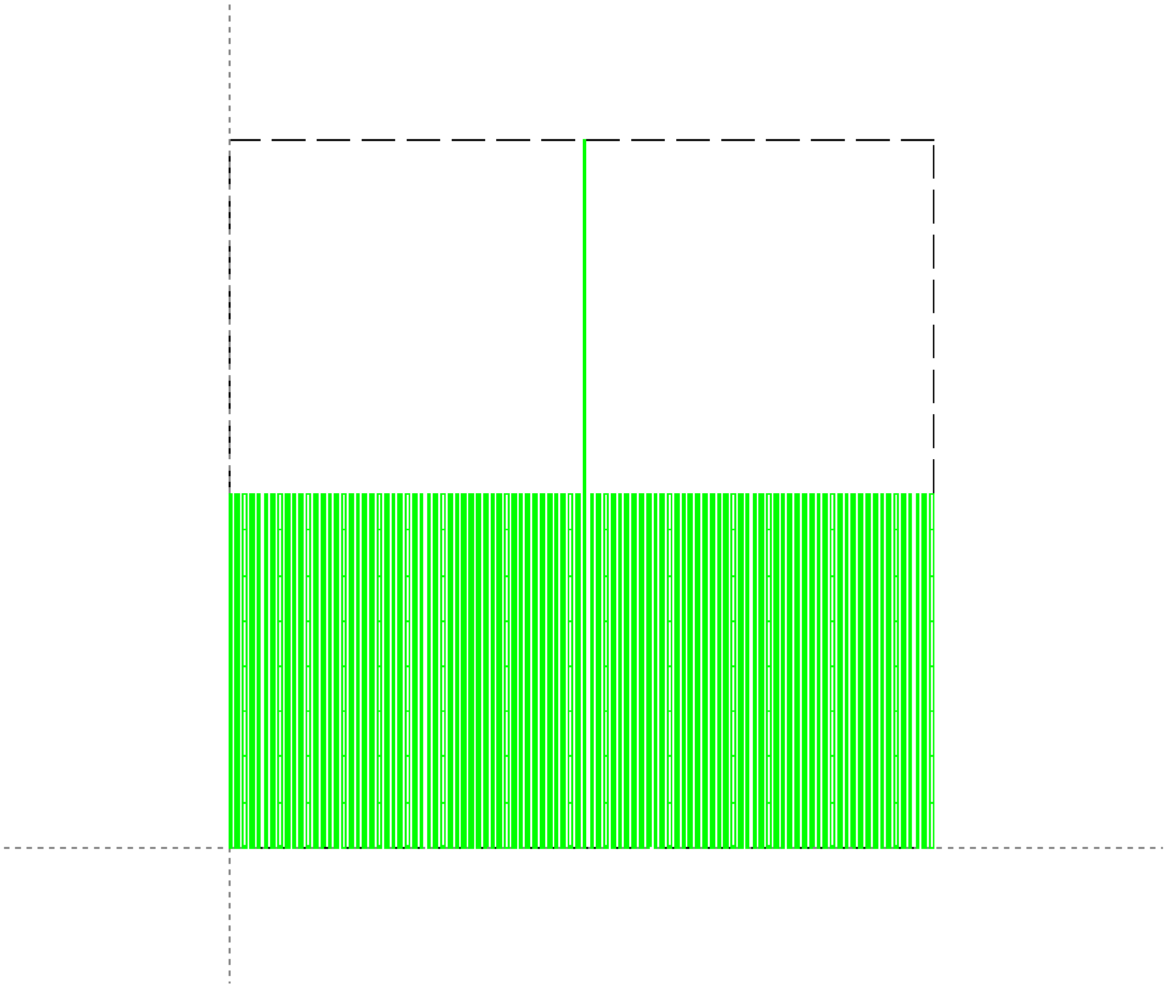
```
File Name:
   Layout Bbox[um]          : (4.002000, 4.000000), (6.548000, 6.560000)
```

Mouse position [um]: 4.891,5.006

View Area [um]: 4.878, 4.939; 4.978, 5.024

The left side of the Viewer displays the different color coding for the threshold levels simulated.

Besides individual regions, the whole layout can be simulated. To do this, uncheck the checkbox Simulate selected regions.



10.2.4 Measuring the CD with Metrology

Overview:

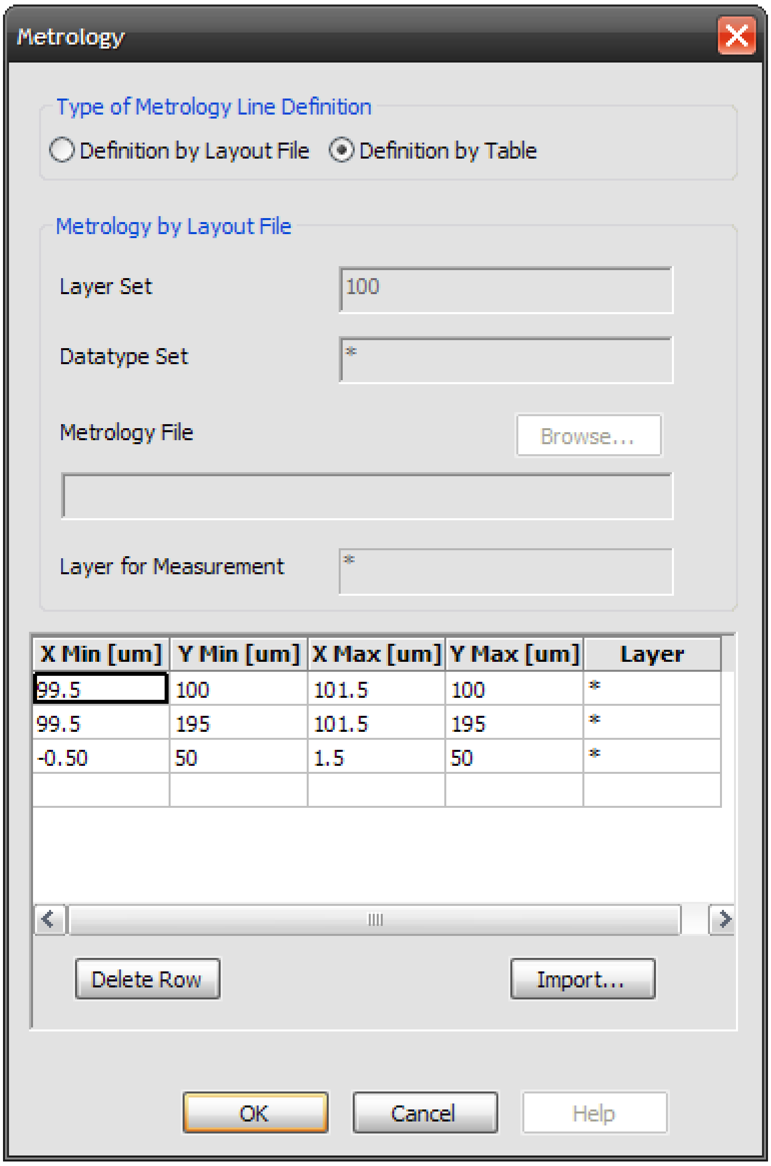
The Metrology module allows the automatic measurement of simulated resist contours. This is helpful when you do want to investigate the impact of process variations.

Application:

First load a sample file (Finger Structure.gds) and attach an E-Beam module to it. Change the settings to full resist contour and run the flow. To measure this we introduce measurement lines, by first inspecting the layout for positions that interest us.
We will choose lines at these coordinates:

x start [µm]	y start [µm]	x end [µm]	y end [µm]
99.50	100	101.50	100
99.50	195	101.50	195
-0.50	50	1.50	50

Add a Metrology module to the flow and select the "Definition by Table" option. This will allow the entry of the measurement lines.



Type of Metrology Line Definition

○ Definition by Layout File ● Definition by Table

Metrology by Layout File

Layer Set: 100

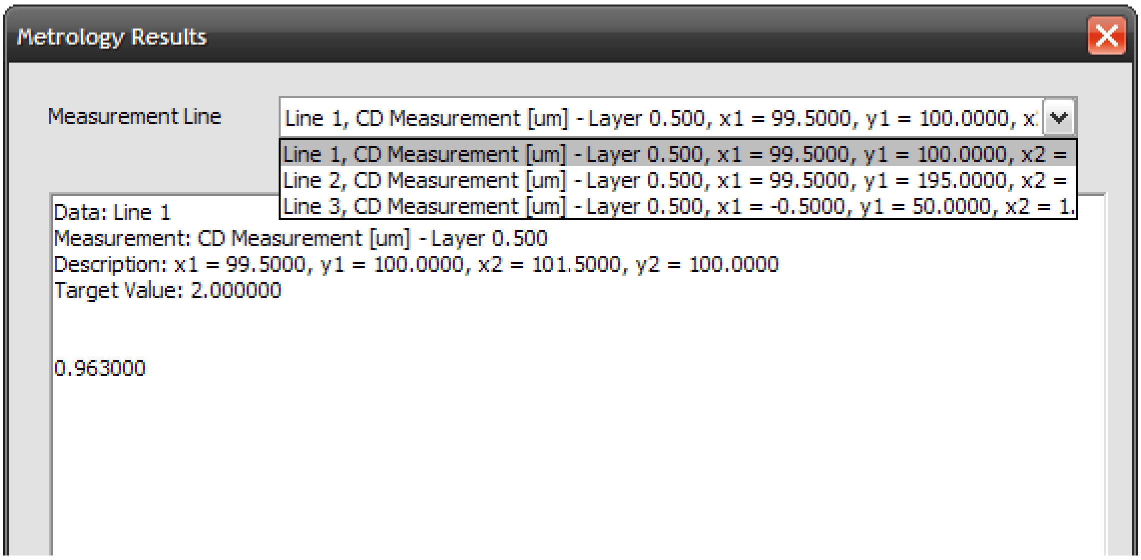
Datatype Set: *

Metrology File:

Layer for Measurement: *

X Min [um]	Y Min [um]	X Max [um]	Y Max [um]	Layer
99.5	100	101.5	100	*
99.5	195	101.5	195	*
-0.50	50	1.5	50	*

Running the module and then selecting the "View Data Sheet" entry will display the measured results.



Measurement Line: Line 1, CD Measurement [um] - Layer 0.500, x1 = 99.5000, y1 = 100.0000, x

```
Line 1, CD Measurement [um] - Layer 0.500, x1 = 99.5000, y1 = 100.0000, x2 =
Line 2, CD Measurement [um] - Layer 0.500, x1 = 99.5000, y1 = 195.0000, x2 =
Line 3, CD Measurement [um] - Layer 0.500, x1 = -0.5000, y1 = 50.0000, x2 = 1.
```

```
Data: Line 1
Measurement: CD Measurement [um] - Layer 0.500
Description: x1 = 99.5000, y1 = 100.0000, x2 = 101.5000, y2 = 100.0000
Target Value: 2.000000


0.963000
```

The measured value is displayed as single value.

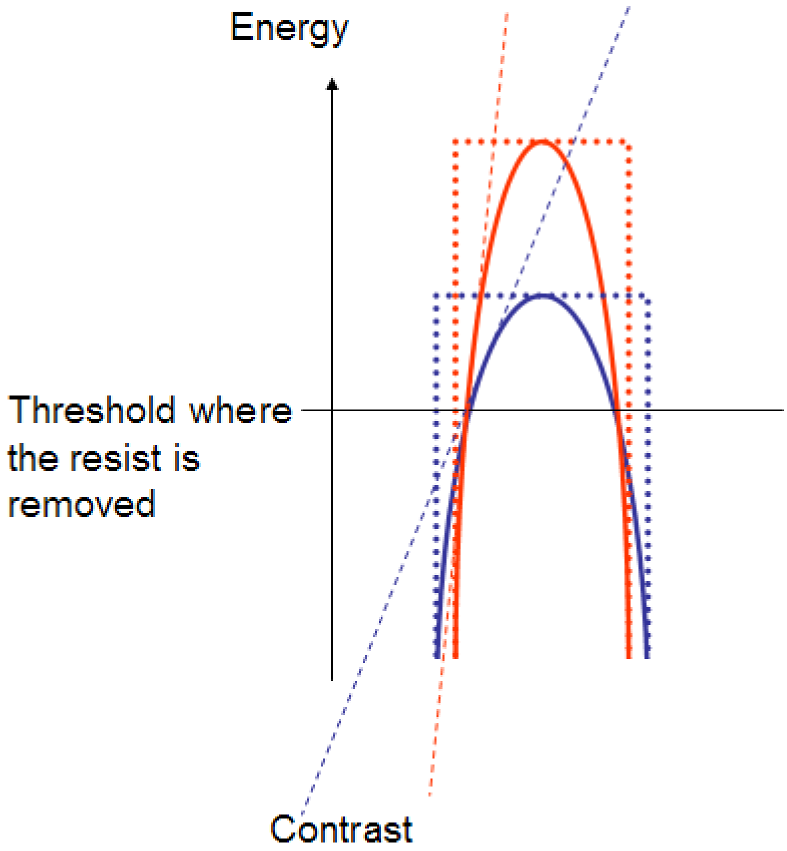
Conclusion:

It has been shown how the automatic measurement can be done with BEAMER. This is especially useful when variables are used to display process Parameters variations.

10.2.5 FDA

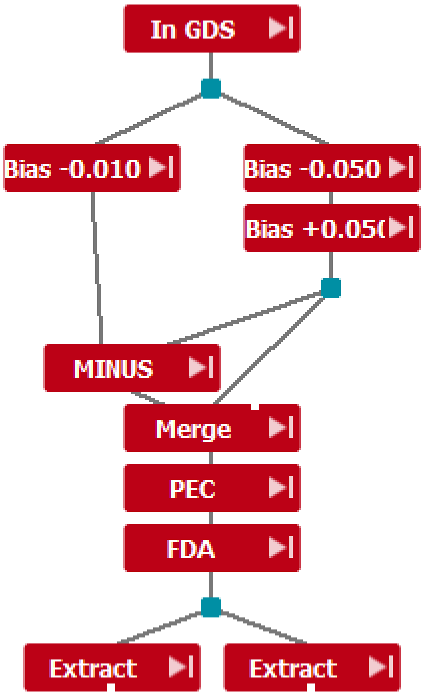
Overview:

The FDA with the coarse / fine split has it's usefulness when a very critical structure needs to be exposed with high contrast. The background is that the reduced size of the critical structure but the higher dose applied to it will trigger the resist to develop at the same energy level but with a higher contrast, improving the writing result.

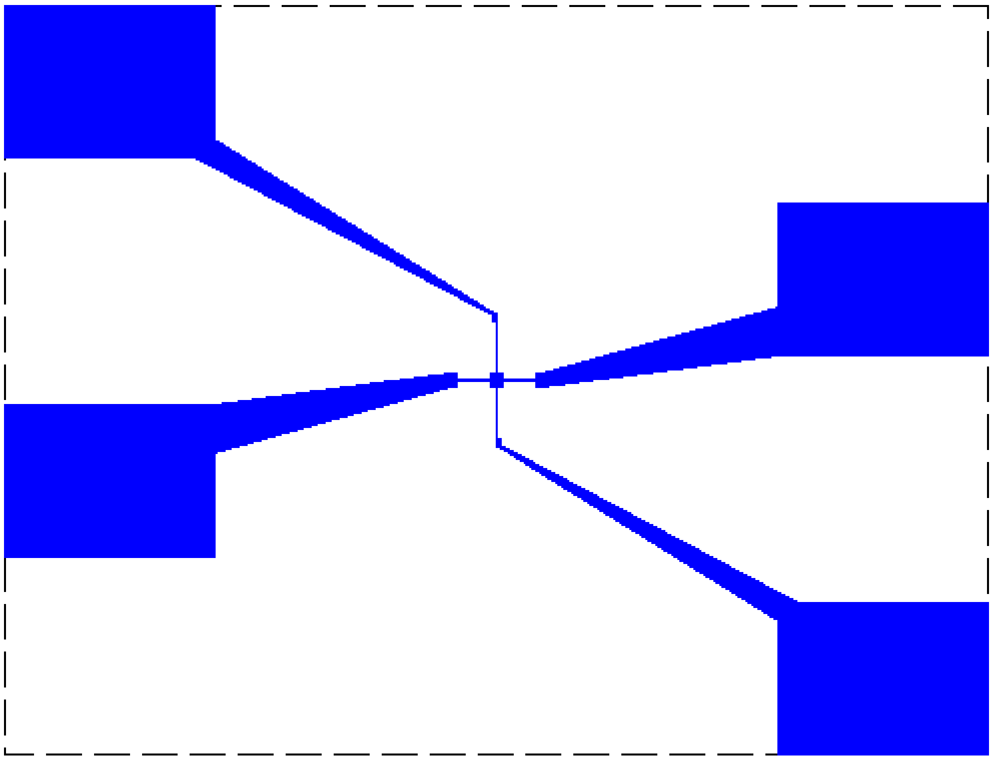


Application:

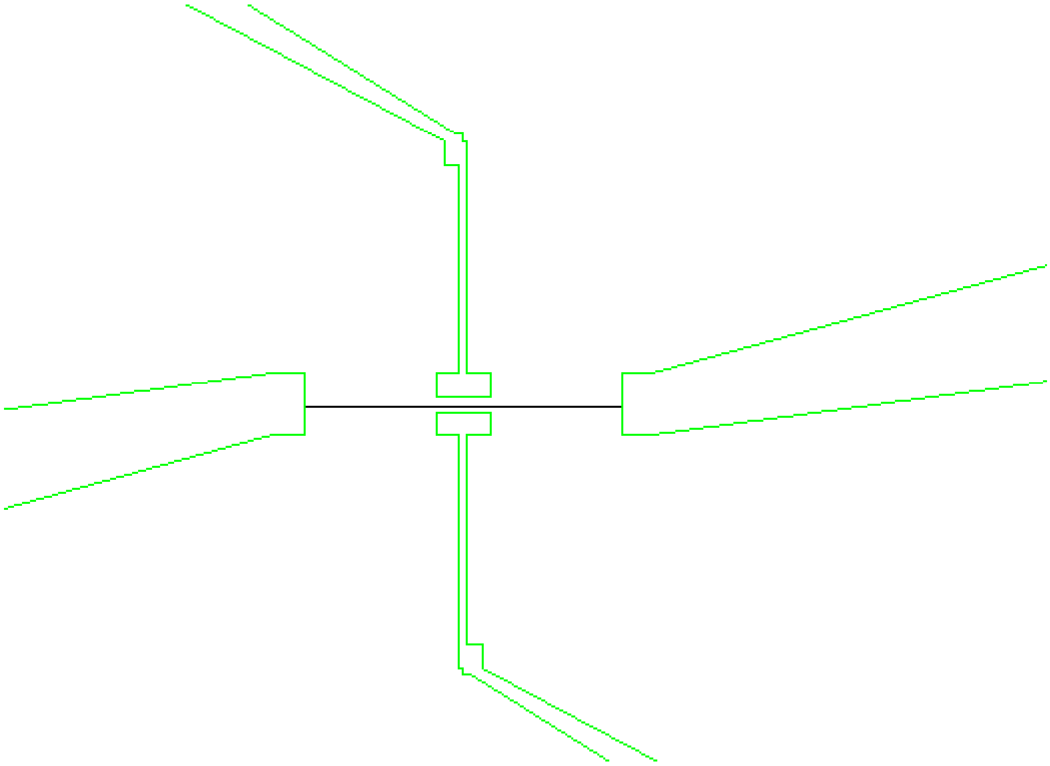
The flow for this application is a slightly modified coarse / fine split. After the Import of the layout we split the layout for the coarse / fine operation. The left hand branch will have a slight negative bias since we want to decrease the size of the critical structures. The right hand branch is first negatively and then positively (with a target layer of 9) biased by the same values, eliminating all structures of 100nm or less. The MINUS then separates fine and the coarse remains at the right hand side of the second Splitter module. A Merge module then puts these two layouts together into the PEC. In the FDA then the layer 0 is assigned a +10% dose is given to its current value. To prepare for export a Splitter and an Extract module are used to separate the fine (layer 0) and coarse (layer 9) structures again. For the Export parameters, check the chapter Machine Format Fracturing.



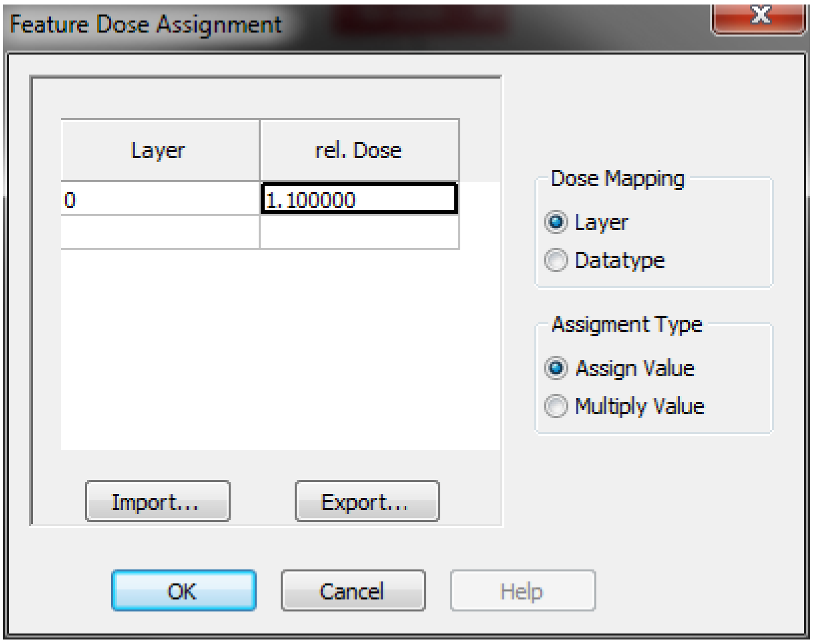
The Layout as whole:



The separated layout for the region of interest:



FDA parameters:



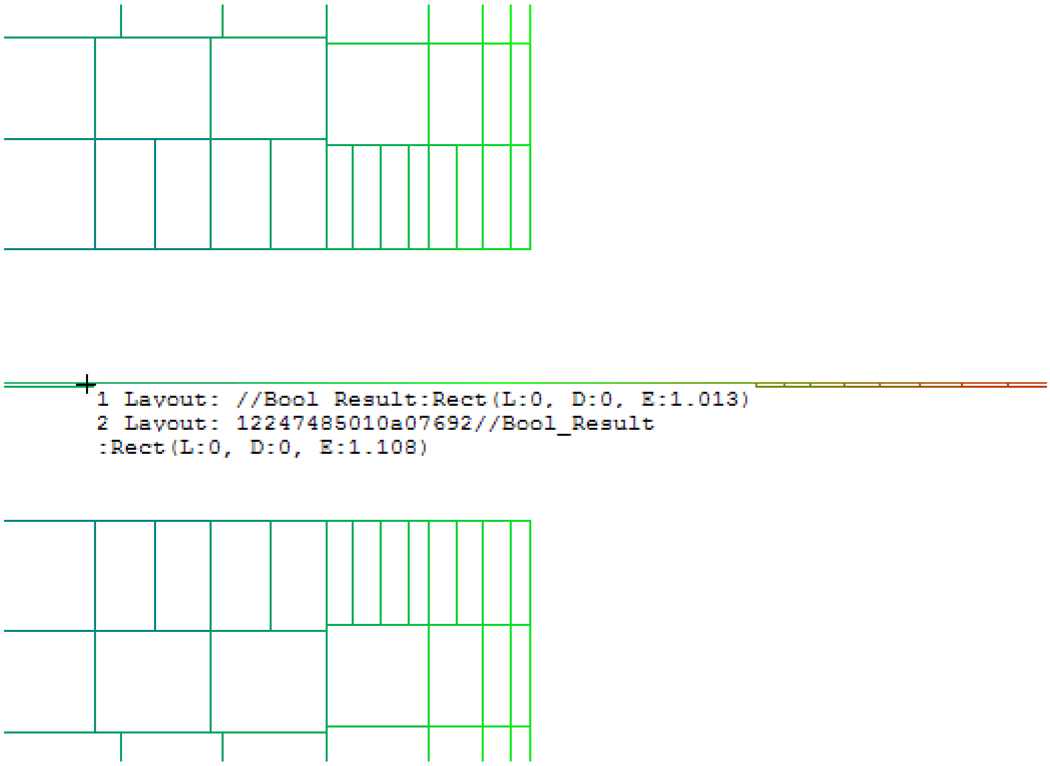
Feature Dose Assignment

Layer	rel. Dose
0	1.100000

Dose Mapping
● Layer
○ Datatype

Assigment Type
● Assign Value
○ Multiply Value

A comparison view of the PEC result and the subsequent FDA:



```
1 Layout: //Bool Result:Rect(L:0, D:0, E:1.013)
2 Layout: 12247485010a07692//Bool_Result
:Rect(L:0, D:0, E:1.108)
```

Conclusion:

Improving the results for critical structures can be done using the described method of coarse / fine split with an FDA dose correction and a negative bias for the critical structures.

10.3 Proximity Effect Correction FAQ

Frequently asked questions on PEC are addressed in this section.

10.3.1 What is PEC?

PEC (pronounced *'pek*) stands for proximity effect correction. Proximity effect correction in BEAMER is an edge-correction technology. PEC connects the threshold of the resist to the resist development to where the resist edge will land. This means for PEC (or 3D-Edge PEC) the absorbed energy of the resist is analyzed in the pattern and dose assignments are made such that the absorbed constant energy at threshold lands at the edge of the intended design.

The Proximity Effect Correction (PEC) module provides a strong set of features for correction of effects caused by electron scattering, beam blur and the pattern transfer process during e-beam exposure. These “proximity effects” are pattern density dependent critical dimension (CD) variations and are observed especially in large arrays (e.g., a large array is greater than 100µm by 100µm at 100kV on Si) of small features. The features in the center will have a larger CD than those at the edge and corners due to the additional absorbed energy in the resist from backscattered electrons. Other examples are layouts with varying pattern density, where the CD in dense areas is larger than the CD of less dense areas. For accurate printing results of very small features (e.g. below 50nm), the correction of short range effects (resist blur) are needed in addition.

The proximity effect is characterized phenomenologically by a Point Spread Function (PSF), which specifies the relative energy deposited at the distance *r* from a point exposure. PSFs are calculated using Monte Carlo (MC) simulation software or measured using experiments (exposure and measurement of a test pattern). BEAMER will interface with the commercial MC simulation packages (e.g. *Sceleton)* and an ASCII interface for others. The BEAMER installation includes MC simulated PSF’s for some common stack / energy combinations.

It is also possible to represent the PSF by Gaussian functions (double, triple or quadruple). The simplest representation is a double Gaussian with the three parameters:

- alpha (α) - specifies the forward (short range) scattering range of the electrons
- beta (β) - specifies the backward (long range) scattering range of the electrons
- Eta (η) - the ratio of the energy in the beta Gaussian to the energy in the alpha Gaussian

The PSF strongly depends on:

- Acceleration voltage (eg. 50kV or 100kV)
- Stack materials and their thicknesses

It is desirable to use a higher order Gaussian for representing the PSF of more complicated stacks (like III-V compounds), or to include mid-range resist and etching effects, as with GaAs. BEAMER supports up to four Gaussians.

The PEC module provides great flexibility for handling both forward scattering (short range effects) and backward scattering (long range effects). The default setting is for correction of long range effects but you can choose to include short range effects, which is useful in cases when the layout has feature sizes, or gaps, smaller than three times the effective blur.

BEAMER performs a physical fracturing, resulting in a very symmetric correction of layout features. The proximity effect corrected file can be exported to a machine format or in a modified GDSII format, by selecting dose mapping by layer. By using Export with GDSII dose mapping by layer or datatype, it saves a file (in text format) with the extension *.ldt containing the dose information.

Dose Correction

Long Range Correction (back-scattering)
When traveling through the material stack, electrons can be scattered back from the substrate into the resist, depositing energy in long distances (long range) from the beam position. The deposited back-scattered energy as a function of the distance from the beam can be approximated by either single or multi Gaussian function(s) with the width beta, gamma$_1$, gamma$_2$. The function strongly depends on electron energy and the stack, especially the substrate. On a Si substrate, beta is on the order of 30µm at 100keV and in the range of 10µm at 50keV. In addition, at 100kV the shape of the backscatter (long-range) PSF is simple and can be approximated with just one Gaussian (of the width beta). In contrast, when writing on GaAs, beta is in the range of 10µm at 100keV. The PSF is more complex with greater energy contributions in the mid-range (100nm-1µm), requiring the approximation by multiple Gaussian's or better yet by using a numerical PSF table.

Short Range Correction (forward scattering)
As a rule of thumb, we enable short range correction when we have CDs that are within 3x the effective blur for a given process. Short range correction is primarily used to deal with process latitude limitations. Process latitude is the change in a critical dimension (CD) with respect to a change in dose. A good process latitude is described by a small change in CD with respect to a large change in dose. A poor process latitude is described by a large change in CD with respect to a small change in dose. In simulation, we can tie this to the effective FWHM blur of a process. The smaller the effective blur the better the process latitude. The larger the effective blur, the poorer the process latitude. This also means for highly dense areas, the resolution of what you can print (e.g., dense lines and spaces) is going to be the limited by the process blur especially when there is a lot of back-scattered energy also contributing to the total absorbed energy of the resist.

Going one step further in tying process latitude to pattern density, the more sparse the pattern density, the better the process latitude improves with a fixed effective blur; however the process latitude is still limited by the effective blur. Since there is very little back-scattered energy to contribute to the total absorbed energy, more dose will be needed to resolve sparse or isolated features. For sparse or isolated features that are larger than 3x the effective blur, their proper dose assignment is taken care of with simple long range correction in both PEC and 3D-Edge PEC. However, when trying to print an isolated feature that is sized within 3x the effective blur, a dose higher than what long range correction can provide is required. Experimentally, the process latitude of features smaller than 3x the effective blur will be limited. This is often described by process engineers as, "If I use a high enough dose, my small isolated features print but are too large. When I lower the dose by a little, the features don't print at all." Again, they simply described a limited process latitude brought on by an effective blur that is roughly the same size as their small isolated features.

To repeat, we enable short range correction when we have CDs that are within 3x the effective blur for a given process. This usually goes hand-in-hand when we have a poor process latitude, which is described by a large change in CD with respect to a small change in dose. If you have a small effective blur, which is achieved using cold development. In the field, the effective short range blur is roughly 10-20nm for PMMA and ZEP for cold development. To obtain a small effective blur for PEC or 3D-Edge PEC, lowering the developer temperature can improve this value to obtain the aforementioned blurs that we see in the field.

Effective Blur FWHM

The Effective Blur is dependent on writing parameters like beam energy, current, focus, as set on the machine (jobdeck) and the resist process. The Effective Blur needs to be entered as Beam FWHM (Full-Width-Half-Max). If “Include Short Range” mode is selected, the Effective Blur will be convolved with the PSF (or alpha if Gaussian functions are used). In most cases, the numerical PSF will not include the beam size effect; if it does, the Beam FWHM parameter should be set to zero (0).

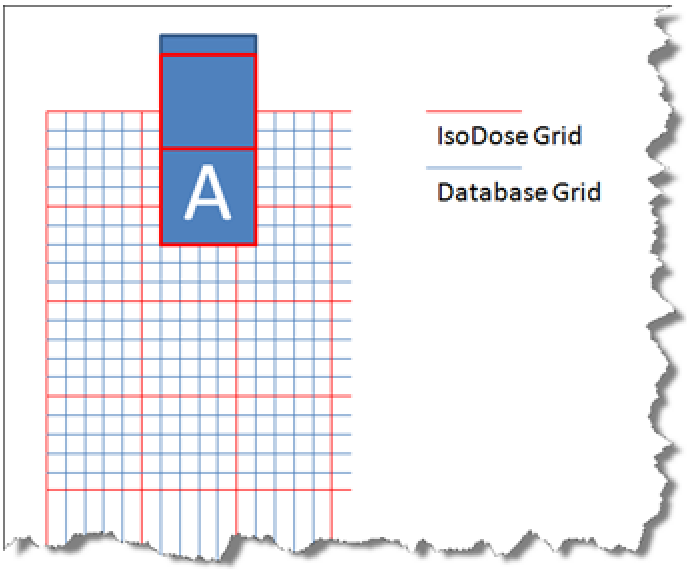
Number of Dose Classes

The number of doses depends on the e-beam system and the desired correction accuracy. Typical settings are 64 or 128. A higher number of doses will result in finer correction, but result in higher data volume, longer calculation times and longer writing times. When choosing Include Short Range Correction, a high number of dose classes should be used.

Isodose Grid

This is the grid for borders between the dose classes (isodose regions). It will define the physical fracturing of the trapezoids to the different dose classes. It is very important to set this grid as a multiple of the beam step size planned for fracturing and writing. Its value depends on the "Include Short Range" correction mode. Smaller grids are needed for short range (on the order of alpha, but not smaller than the beam step size). Without short range correction, the grid may be set to a value around 1% of beta.

In this representation you see a blue database grid and a red isodose grid. If a feature needs to be fractured into multiple shapes in order to be able to assign different doses to it, the physical fracturing will start at lower left corner of the design elements and perform the fracture from that point on as a the virtual origin. This reduces the occurrence of slivers significantly.



10.3.2 What is PEC doing and How do I setup the module correctly?

PEC (pronounced *'pek*) stands for proximity effect correction. Conventional e-beam lithography processes are optimized for ''binary'' patterns (resist or no resist) while controlling the line width as critical dimension (CD). The resist is used in a ''threshold'' mode, and lateral dimensions are controlled by adjusting the absorbed energies at the edge of the features to a constant level (Dose to Clear). The variations in the absorbed energy, due to proximity effects (scattering of electrons), must be corrected using exposure dose modulation.

BEAMER PEC insures (by allocating suitable dose modulation) that the absorbed energy (AE) at features edges is uniform 0.5 (= dose to clear) at all over the layout. Thereby the development reaches all edges at the same time and positive resist is removed (negative resist will remain) in all exposed areas at absorbed energies > 0.5. BEAMER is applying the most powerful Edge equalization algorithms available.

BEAMER takes this 0.5 as a reference to the dose to clear. So whatever the correct results will be, the outline of the layout will receive precisely the dose to clear energy, to ensure that the resist outline on the wafer is equal to the layout design.

The reference dose (Base Dose) of 1 is equivalent to the dose to size. The absolute base dose is dependent on the resist sensitivity and needs to be experimentally calibrated for a given resist process (inquire detailed application note for base dose calibration). The base dose is independent of layout after one time calibration for a given resist process. The layout independent base dose is one of the main benefits of PEC.

When applying the proximity effect correction (PEC) choosing the appropriate correction function is very important. The correction function is called the Point Spread Function or PSF for short. This classical PEC approach corrects the scattering effects of electrons in the material stack. The range of scattering, which depends on the electron energy (acceleration voltage), resist/layer materials and layer thicknesses including the substrate, is divided into two types:

1. **Forward scattering.**
   a. When electrons travel through the resist they interact with the atoms and molecules. This results in a direction change of the electrons (scattering) and exchanges of energy between electrons and atoms (energy deposition). The resulting energy deposition range (energy as function of distance) can be described by a Gaussian function. The range of scattering and energy deposition depends mostly on the electron energy and is in the range of a few nanometers for 100keV machines. The energy deposition of this forward scattering range, also called short range, can be defined by the width of the Gaussian, and is designated by the Greek letter alpha.

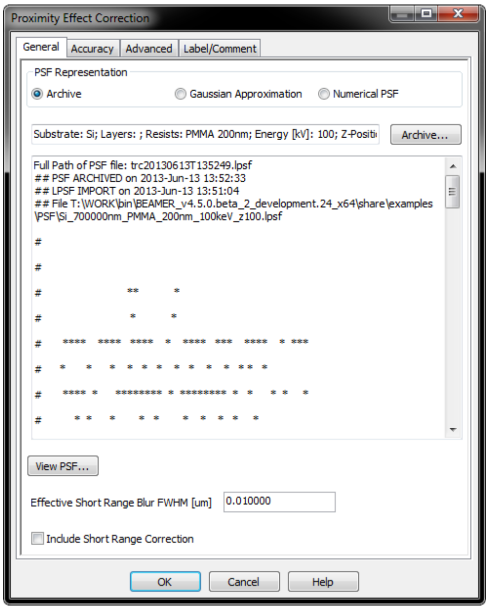
   b. **Beam blur / effective alpha / short range correction.** In addition to the forward scattering the beam size (beam blur) needs to be considered for short range effects. The beam blur is dependent on the beam current of the exposure and is typically in the range of a few nanometers for low beam currents (> 1nA) and may become few 10nm for very high beam current. The forward scattering and beam blur can be combined to an "effective alpha". Depending on this effective alpha short range correction may be needed or not. For most 100keV Gaussian beam exposure short range correction will not be needed for feature >50nm.

2. **Back scattering / long range correction.** When traveling through the material stack, electrons are scattered back from the substrate into the resist, depositing energy in long distances (long range) from the beam position. The deposited energy as a function of the distance from the beam can be approximated by either single or multi Gaussian function(s) with the width beta, gamma1, gamma2. The function strongly depends on electron energy and the stack, especially the substrate. Beta is on the order of 35µm for a 100keV and in the range of 10µm for a 50keV, when writing on a silicon substrate. On the other hand, beta is in the range of 10µm for 100keV when writing on GaAs. The shape of the backscatter (long-range) PSF is simpler and can be approximated with just one Gaussian (beta) when using a 100kV machine writing on silicon. The PSF is more complex with greater energy contributions in the mid-range (100nm-1µm) for 100keV on GaAs, requiring the approximation by multiple Gaussians or better yet by using a numerical

PSF table.

There are several ways of determining the right correction functions (PSF's):

- Monte Carlo simulation - Monte Carlo simulation software can simulate the PSF using computational models to calculate the scattering of electrons in the material for a given stack and electron energy. The result is a table with energy as a function of distance from the beam position. This PSF function table can be used directly, or approximated with Gaussian functions.
- Experimental methods - A correction function can also be determined by using experimental methods (writing and measuring of test patterns) by providing a table equivalent to energy deposition as a function of distance. Such experimental methods can cover process effects, e.g., resist bake, development and etching, in addition to the electron scattering effects.



PSF Representation: ● Archive ○ Gaussian Approximation ○ Numerical PSF

Substrate: Si; Layers: ; Resists: PMMA 200nm; Energy [kV]: 100; Z-Positi

```
Full Path of PSF file: trc20130613T135249.lpsf
## PSF ARCHIVED on 2013-Jun-13 13:52:33
## LPSF IMPORT on 2013-Jun-13 13:51:04
## File T:\WORK\bin\BEAMER_v4.5.0.beta_2_development.24_x64\share\examples
\PSF\Si_700000nm_PMMA_200nm_100keV_z100.lpsf

#

#

#             **      *

#             *      *

#   ****  ****  ****  *  ****  ***  ****  * ***

#  *   *   *  *  *  *  *  *  *  *  ** *

#   **** *   ********* * ********* *  *   * *  *

#     * *   *    * *    *  *  *  *   *
```

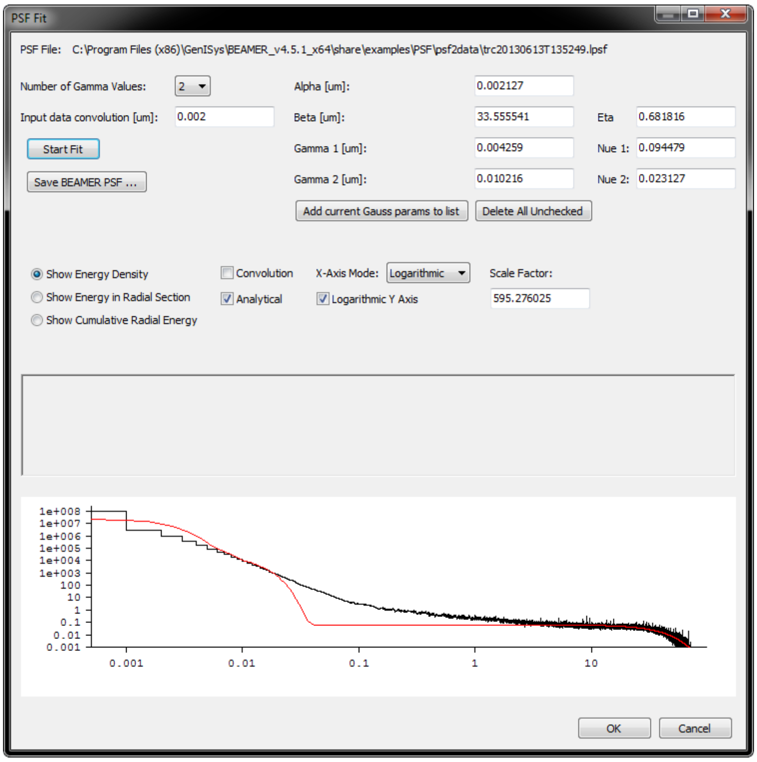
Effective Short Range Blur FWHM [um]: 0.010000

☐ Include Short Range Correction

The PEC module provides great flexibility for determining a PSF representation

- Gaussian approximation with 2-4 Gaussian (alpha, beta, Eta)
- Numerical PSF loaded from a file
  - Sceleton Monte Carlo simulation file
  - Chariot Monte Carlo simulation file
  - text file interface for other Monte Carlo simulator
  - text file for experimental correction function

In case of loading PSF files, those can be viewed and fitted to multi-Gaussian:



PSF File: C:\Program Files (x86)\GenISys\BEAMER_v4.5.1_x64\share\examples\PSF\psf2data\trc20130613T135249.lpsf

Number of Gamma Values: 2

Input data convolution [um]: 0.002

Alpha [um]: 0.002127

Beta [um]: 33.555541 Eta 0.681816

Gamma 1 [um]: 0.004259 Nue 1: 0.094479

Gamma 2 [um]: 0.010216 Nue 2: 0.023127

● Show Energy Density

○ Show Energy in Radial Section

○ Show Cumulative Radial Energy

□ Convolution

☑ Analytical

X-Axis Mode: Logarithmic

☑ Logarithmic Y Axis

Scale Factor: 595.276025

Depending on the PSF, beam blur and smallest features to print, you need to decide on including short range correction or not:

- The default setting of the PEC module is takes care of long range effects
- Optionally, you can choose to include short range effects, which is useful in cases when the layout has feature sizes smaller than three times effective alpha

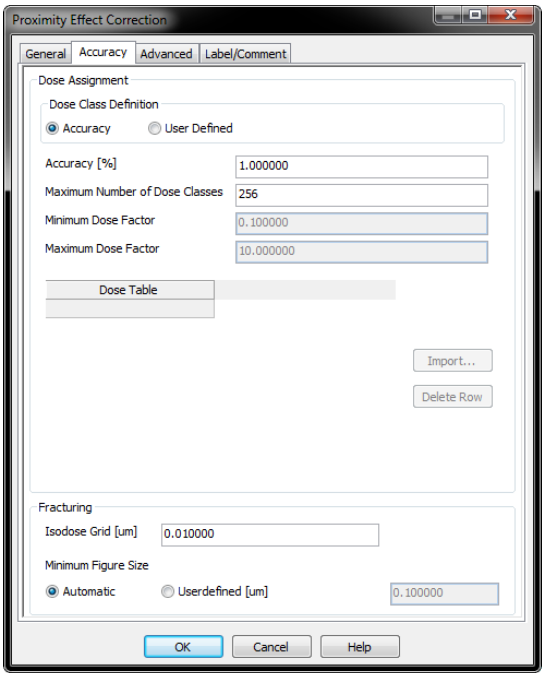
NOTE: These defaults settings are for the exposure at 100kV of thin resist on silicon.

Simple guidelines to follow:

- Use a numerical PSF (from Monte Carlo simulation or experimental results) if available
- Include short range correction only in case you have structures or gaps < 3 times alpha. In most other cases, the short range correction will not have an effect on the dimension control and will just increase the calculation time and data volume.

The second PEC tab gives control on the accuracy. The default (and recommendation) is to define the accuracy of the correction by % (dose change). This assures that only the needed number of dose classes are used to reach the defined accuracy. 1% is a good number, as in practice the control of the dose on the machine mostly is in that order. If the machine a a limitation with regards to the number of dose classes, the maximum number of dose classes can be set accordingly.

The second option is for user defined dose classes. This assures that all corrected layouts will have the same dose classes. The most common application case for this is at machine where the dose information is not included in the machine format, but needs an additional modulation table to be included in the job-deck file. By assigning always the same dose classes, the same modulation table can be uses (no need to update the modulation table in the job-deck file every time).



Dose Assignment

Dose Class Definition

● Accuracy ○ User Defined

Accuracy [%] 1.000000

Maximum Number of Dose Classes 256

Minimum Dose Factor 0.100000

Maximum Dose Factor 10.000000

Fracturing

Isodose Grid [um] 0.010000

Minimum Figure Size

● Automatic ○ Userdefined [um] 0.100000

The most important point on the Isodose Grid is to set it equal or multiple the beam step size (shot pitch) which is going to be used for the machine fracturing. The minimum figure size by default is automatic. However you can manually set it depending on the correction function, settings and layout. To manually set the minimum figure size, follow these guidelines:

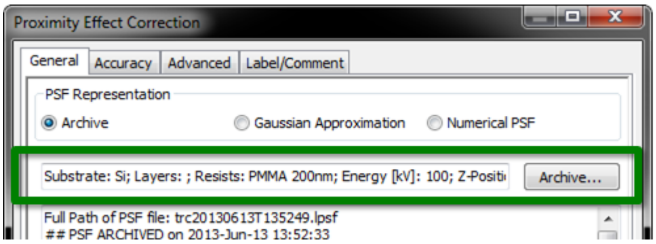
- Always set it multiple the Isodose Grid
- If short range correction is switched off, set it to > Beta * % Accuracy (e.g. for Beta of 30µm with 1% accuracy > 300nm)
- If short range correction is used, set it small (e.g. 10nm), but newer smaller that beam step size.

For avoiding long correction time, large data volume and large exposure time set the maximum value for Minimum Figure Size required for the correction.

10.3.3 How do I perform a long range correction?

The long range correction of layouts takes only the back-scattering effect of the writing process into account. This example will show how to use the long range correction.

We will perform a long range correction using a default PSF from the archive, which is 200nm of PMMA atop of Si at 100kV:



PSF Representation

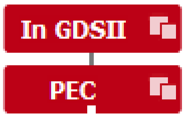
● Archive ○ Gaussian Approximation ○ Numerical PSF

Substrate: Si; Layers: ; Resists: PMMA 200nm; Energy [kV]: 100; Z-Positi

Full Path of PSF file: trc20130613T135249.lpsf

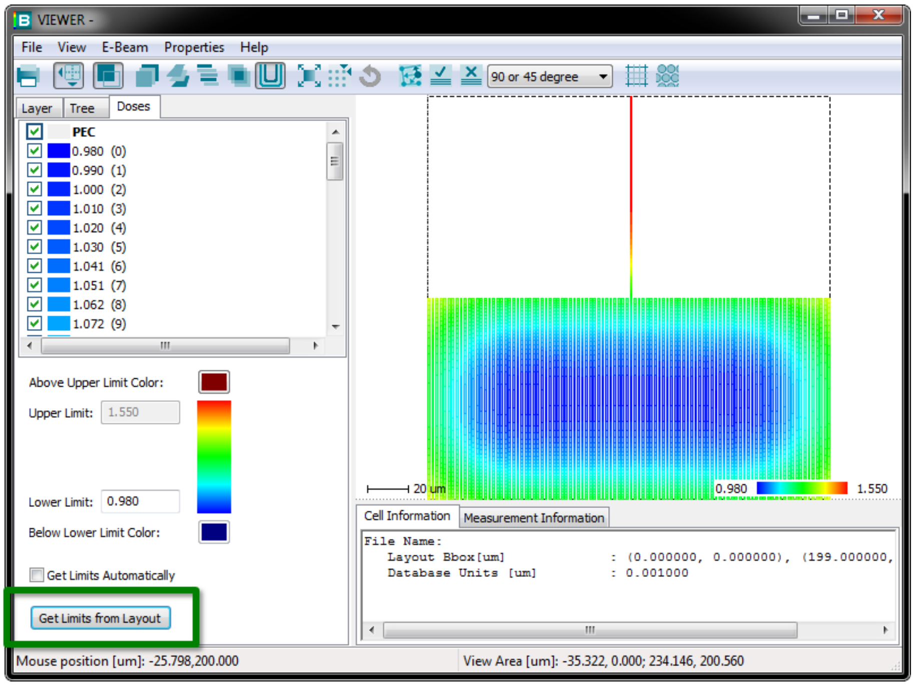
PSF ARCHIVED on 2013-Jun-13 13:52:33

1. For this example, import a GDSII file (Finger_Structure.gds) from the BEAMER example directory (C:\Program Files (x86)\GenISys\BEAMER_v4.*X*.*X*_x64\share\examples\Layouts) and attach the PEC module (using the default settings above) to the import.



2. Execute the **Proximity Effect Correction** by clicking **Run** ▶| on the module or clicking the **Run** button on the tool bar. When executed, the ▶| icon will change to ⧉.

3. View the layout by clicking . Click **Color by Dose** on the VIEWER tool bar and in the Doses Tab click **Get Limits from Layout**. You will see the following in the VIEWER:



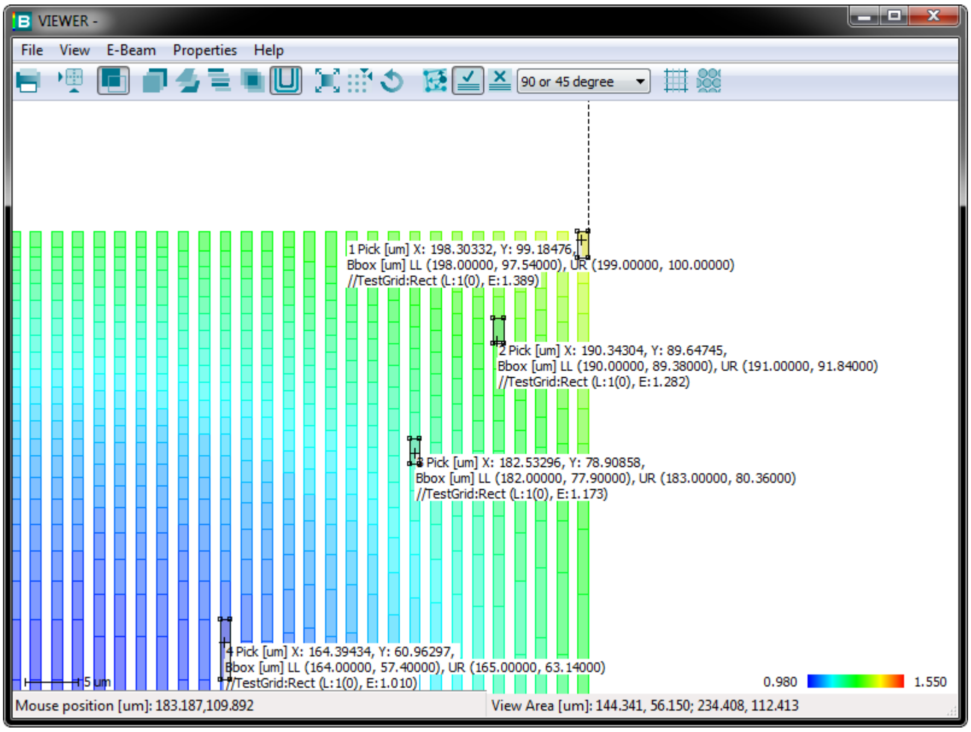
☑ PEC
☑ 0.980 (0)
☑ 0.990 (1)
☑ 1.000 (2)
☑ 1.010 (3)
☑ 1.020 (4)
☑ 1.030 (5)
☑ 1.041 (6)
☑ 1.051 (7)
☑ 1.062 (8)
☑ 1.072 (9)

Above Upper Limit Color:
Upper Limit: 1.550
Lower Limit: 0.980
Below Lower Limit Color:
□ Get Limits Automatically
Get Limits from Layout

```
File Name:
   Layout Bbox[um]               : (0.000000, 0.000000), (199.000000,
   Database Units [um]           : 0.001000
```

Mouse position [um]: -25.798,200.000
View Area [um]: -35.322, 0.000; 234.146, 200.560

The colors indicate a temperature scale: blue is the lowest dose and red is the highest dose. BEAMER performs a physical fracturing, resulting in a very symmetric correction for layout elements. Click to show hide the side and bottom tabbed panes. If you zoom into the layout, you will see the following. The further away from the center, the higher the dose as we approach the corner. If you examine the the upper right corner of the lines and spaces array, you'll notice the corner lines get higher doses. This is because the amount of backscattered energy is lower in the corners which requires a higher applied dose to those areas. You can read the dose value of a specific polygon by double-right clicking on the polygon and looking for the **E:** value.



1 Pick [um] X: 198.30332, Y: 99.18476,
Bbox [um] LL (198.00000, 97.54000), UR (199.00000, 100.00000)
//TestGrid:Rect (L:1(0), E:1.389)

2 Pick [um] X: 190.34304, Y: 89.64745,
Bbox [um] LL (190.00000, 89.38000), UR (191.00000, 91.84000)
//TestGrid:Rect (L:1(0), E:1.282)

3 Pick [um] X: 182.53296, Y: 78.90858,
Bbox [um] LL (182.00000, 77.90000), UR (183.00000, 80.36000)
//TestGrid:Rect (L:1(0), E:1.173)

4 Pick [um] X: 164.39434, Y: 60.96297,
Bbox [um] LL (164.00000, 57.40000), UR (165.00000, 63.14000)
//TestGrid:Rect (L:1(0), E:1.010)

Mouse position [um]: 183.187,109.892

View Area [um]: 144.341, 56.150; 234.408, 112.413

Note: The proximity effect corrected file can be exported to a machine format (if the formatter license is available) or in a machine independent format like a modified GDSII by selecting dose mapping by layer or the LEDB format.

By using the **Export** module with GDSII dose mapping by layer or datatype, two files are saved:

1. the modified GDSII and
2. a dose table (in text format) with the extension *.ldt. This files contains the dose class information (layers and corresponding dose). This is a convenient way to transfer the corrected data another tool.

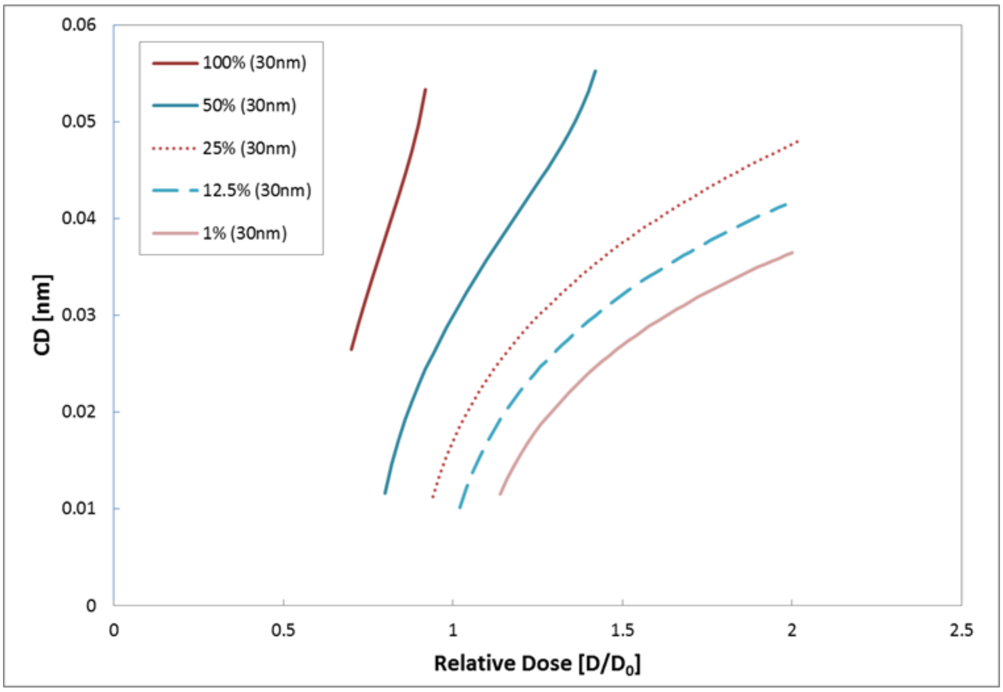
Alternatively, the LEDB (*.ldb) format will produce a single file that can be imported other BEAMER users to be exported to their machine specific format.

10.3.4 What is short range correction and when do I need it?

Proximity effect correction in BEAMER is an edge-correction technology. PEC connects the threshold of the resist to the resist development to where the resist edge will land. This means for PEC (or 3D-Edge PEC) the absorbed energy of the resist is analyzed in the pattern and dose assignments are made such that the absorbed constant energy at threshold lands at the edge of the intended design.

Short range PEC dose corrects features that are smaller than 3 times the effective short range FWHM blur. Using the short range correction for features larger then this value is not useful and will only increase the computational effort on a pattern. Internal algorithms limit the short range computation to areas where truly needed, but still careful consideration needs to be made before turning on this capability.

As a rule of thumb, we enable short range correction when we have CDs that are within 3x the effective blur for a given process. Short range correction is primarily used to deal with process latitude limitations. Process latitude is the change in a critical dimension (CD) with respect to a change in dose. A good process latitude is described by a small change in CD with respect to a large change in dose. A poor process latitude is described by a large change in CD with respect to a small change in dose. In simulation, we can tie this to the effective FWHM blur of a process. The smaller the effective blur the better the process latitude. The larger the effective blur, the poorer the process latitude. This also means for highly dense areas, the resolution of what you can print (e.g., dense lines and spaces) is going to be the limited by the process blur especially when there is a lot of back-scattered energy also contributing to the total absorbed energy of the resist. Examining the chart below, it plots the change in CD with respect to a change in dose by pattern density. Here we simulated a 150um by 150um array of 30nm wide lines (using a 30nm effective short range FWHM blur) at different pitches to achieve the pattern densities listed in the legend. From there, we measured the width of the line at the center of the pattern. The 50% pattern density (solid blur curve) exhibits a poor process latitude. It takes a very small change in dose to result in a large change in CD.

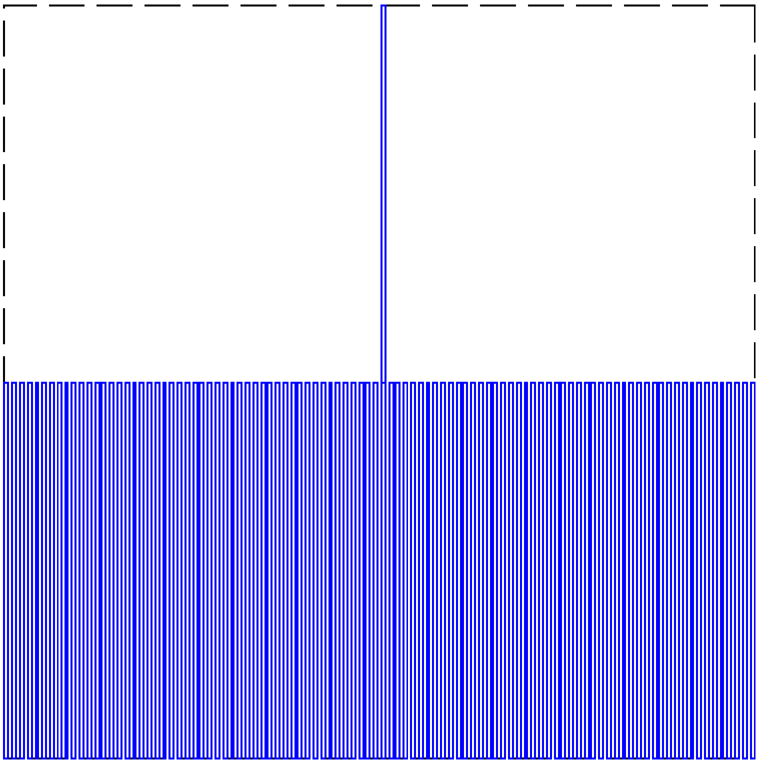


Going one step further in tying process latitude to pattern density, the more sparse the pattern density, the better the process latitude improves with a fixed effective blur; however the process latitude is still limited by the effective blur. Since there is very little back-scattered energy to contribute to the total absorbed energy, more dose will be needed to resolve sparse or isolated features. For sparse or isolated features that are larger than 3x the effective blur, their proper dose assignment is taken care of with simple long range correction in both PEC and 3D-Edge PEC.

However, when trying to print an isolated feature that is sized within 3x the effective blur, a dose higher than what long range correction can provide is required. Experimentally, the process latitude of features smaller than 3x the effective blur will be limited. This is often described by process engineers as, "If I use a high enough dose, the small isolated features print but are too large. When I lower the dose by a little, the features don't print at all." Again, they simply described a limited process latitude brought on by an effective blur that is roughly the same size as their small isolated features.

To repeat, we enable short range correction when we have CDs that are within 3x the effective blur for a given process. This usually goes hand-in-hand when we have a poor process latitude, which is described by a large change in CD with respect to a small change in dose. If you have a small effective blur, which is achieved using cold development. In the field, the effective short range blur is roughly 10-20nm for PMMA and ZEP for cold development. To obtain a small effective blur for PEC or 3D-Edge PEC, lowering the developer temperature can improve this value to obtain the aforementioned blurs that we see in the field.

To demonstrate short range correction, import the Finger Structure.gds. Add a Transform after the Import and set the scaling factor to 0.04. This will generate a pattern with a 40nm grating, and a single line protruding from the grating that you see below.



We shall compare this layout with a short and a long range correction, specifically with regards to line end shortening. For both methods, we will use a double Gaussian approximation of the PSF.

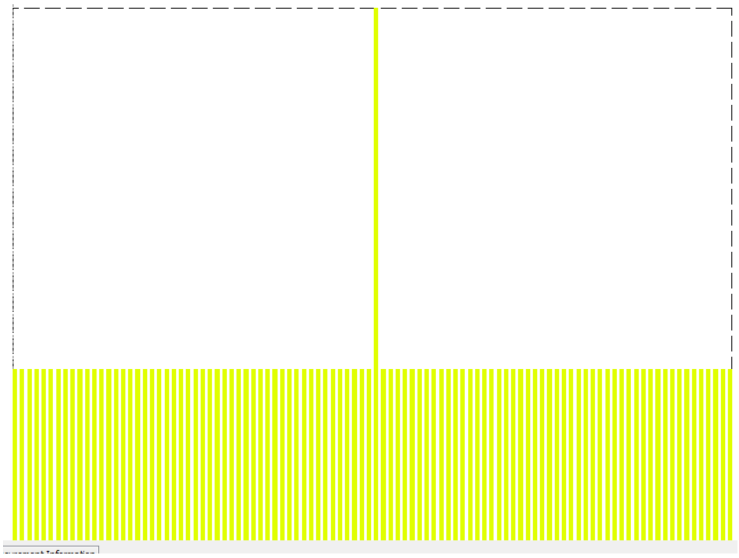
The Parameters for the PEC are the default setting except these modifications:

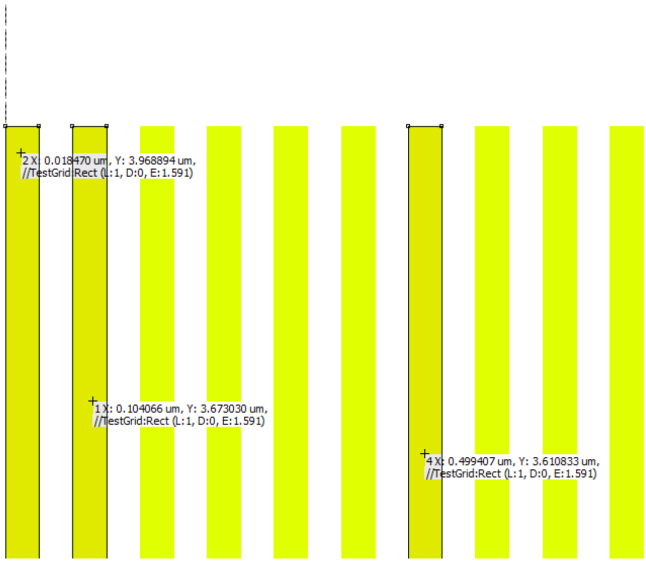
	Long Range	Short Range	Comment
Include Short Range	**OFF**	**ON**	Toggles the short range correction mode.
Effective Blur FWHM	**0.010**	**0.010**	
PSF	**Archive: 200nm PMMA atop Bulk Si**	**Archive: 200nm PMMA atop Bulk Si**	

Iso grid	**0.010**	**0.010**	Set this value to be an integer multiple of the beam step size (shot pitch).
Minimum Figure Size	**automatic**	**automatic**	Keep automatic.

Let us start and inspect this layout with long range PEC and compare this against short range PEC

Long range PEC corrects for the backscatter loss and increases the dose by a factor of 1.59. As Beta is much larger that the pattern size there is no major dose modulation over the pattern. There is no dose increase on the line ends to compensate for the energy loss from the short range loss:



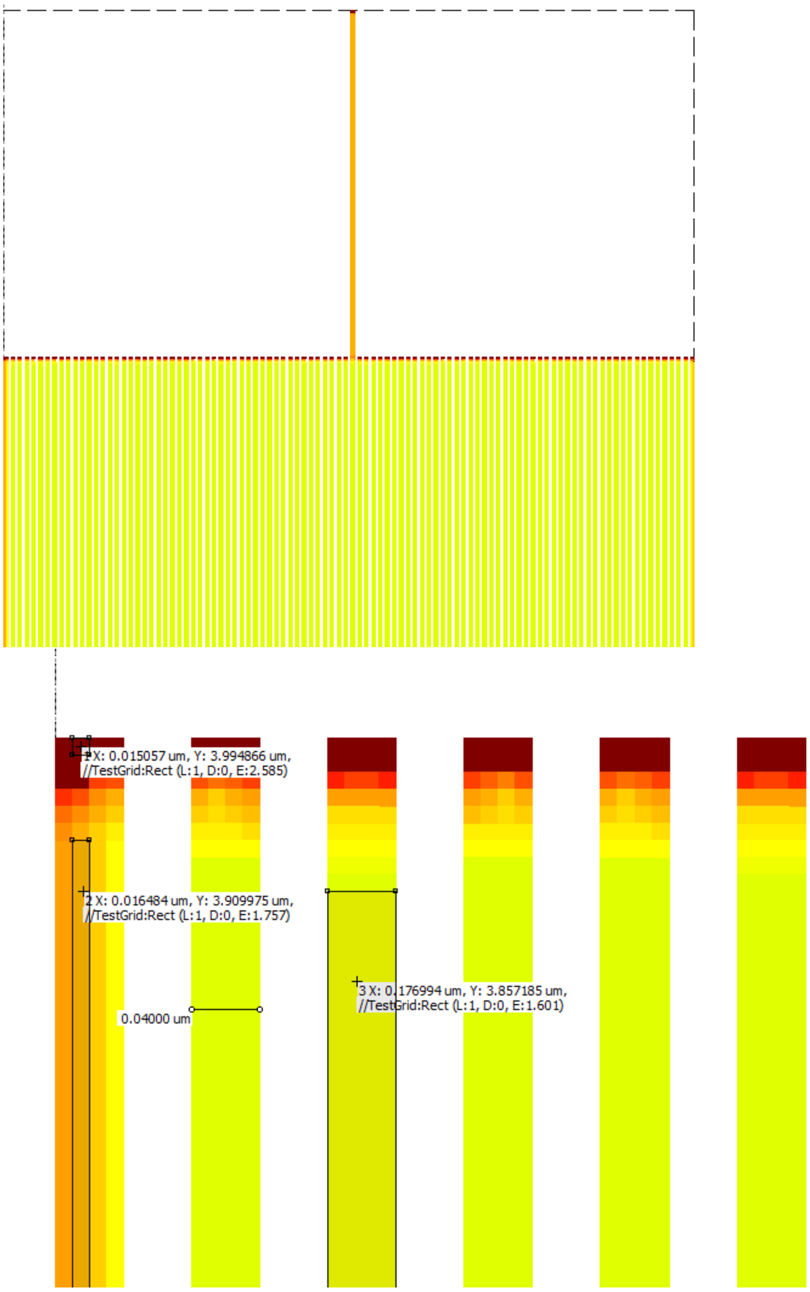


2 X: 0.018470 um, Y: 3.968894 um,
//TestGrid:Rect (L:1, D:0, E:1.591)

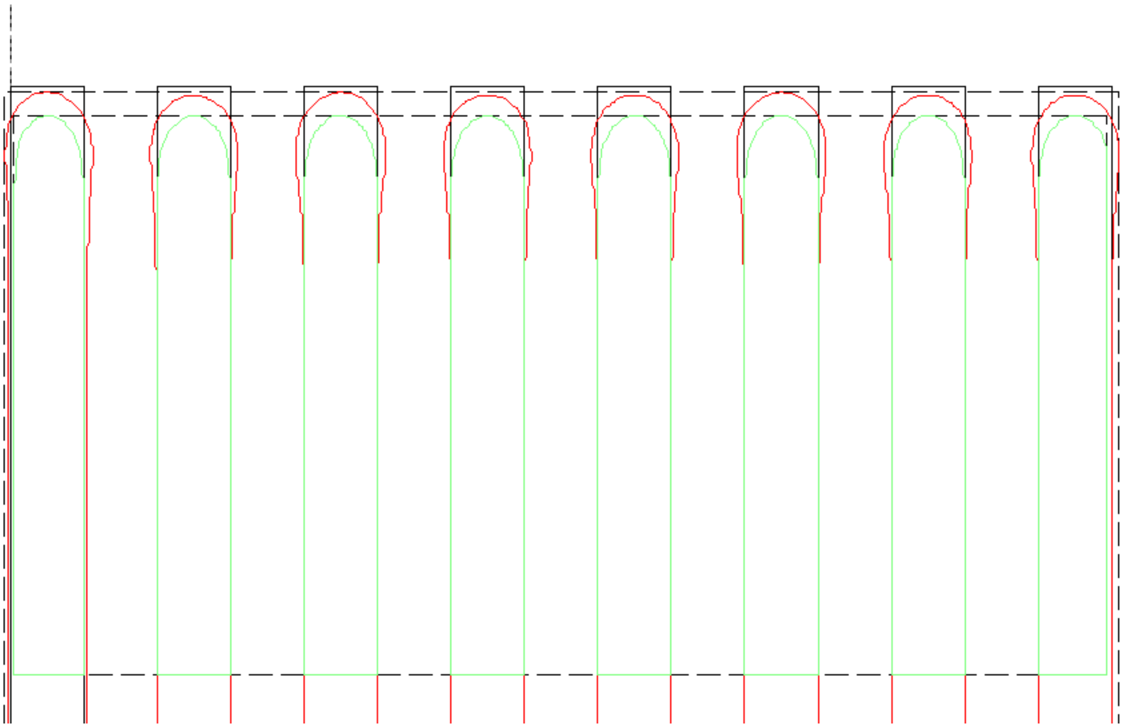
1 X: 0.104066 um, Y: 3.673030 um,
//TestGrid:Rect (L:1, D:0, E:1.591)

4 X: 0.499407 um, Y: 3.610833 um,
//TestGrid:Rect (L:1, D:0, E:1.591)

With short range correction, we see the dose increase at the line ends (up to a factor of 2.6) and also the line at the edge of the patter to compensate for the energy loss by alpha and beam blur:


1 X: 0.015057 um, Y: 3.994866 um,
//TestGrid:Rect (L:1, D:0, E:2.585)
2 X: 0.016484 um, Y: 3.909975 um,
//TestGrid:Rect (L:1, D:0, E:1.757)
0.04000 um
3 X: 0.176994 um, Y: 3.857185 um,
//TestGrid:Rect (L:1, D:0, E:1.601)

Comparing the simulated resist profiles one can see how the line end shortening was corrected.

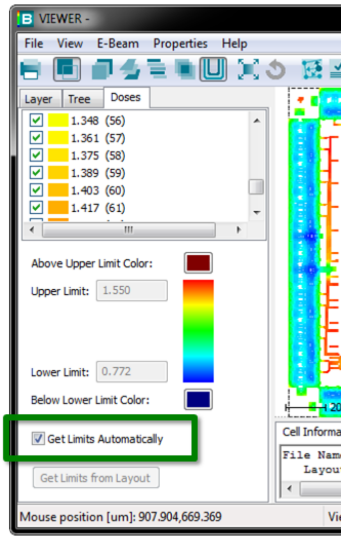


The black outline is for the original layout, the green outline for the long range PEC only showing line-end shortening, the red outline is including short range correction.

10.3.5 When viewing a PEC'd in Color by Dose, how can the dose limits be obtained automatically? 2

To automatically obtain the upper and lower limits when clicking **Color-by-Dose** , enable the checkbox **Get Limits Automatically**. By default this is disabled. When enabled the Upper and Lower Limit text fields are disabled. To keep this feature enabled:

1. Click the checkbox labeled **Get Limits Automatically** to enable the feature via the Doses tab in the VIEWER (bottom left) or accessing the Dose Color dialog from the VIEWER Properties menu:



☑	1.348 (56)
☑	1.361 (57)
☑	1.375 (58)
☑	1.389 (59)
☑	1.403 (60)
☑	1.417 (61)

Above Upper Limit Color:

Upper Limit: 1.550

Lower Limit: 0.772

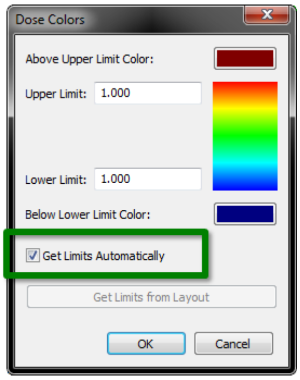
Below Lower Limit Color:

☑ Get Limits Automatically

Get Limits from Layout

File Nam
Layou

Mouse position [um]: 907.904,669.369



Dose Colors

Above Upper Limit Color:

Upper Limit: 1.000

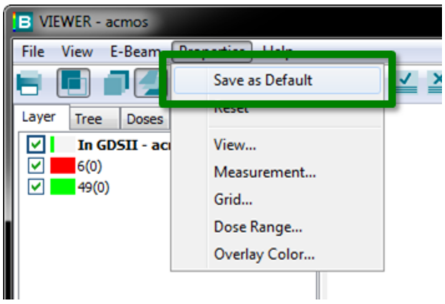
Lower Limit: 1.000

Below Lower Limit Color:

☑ Get Limits Automatically

2. Click **OK** to close the dialog

3. To save the setting, click **Save as Default** in the Properties menu



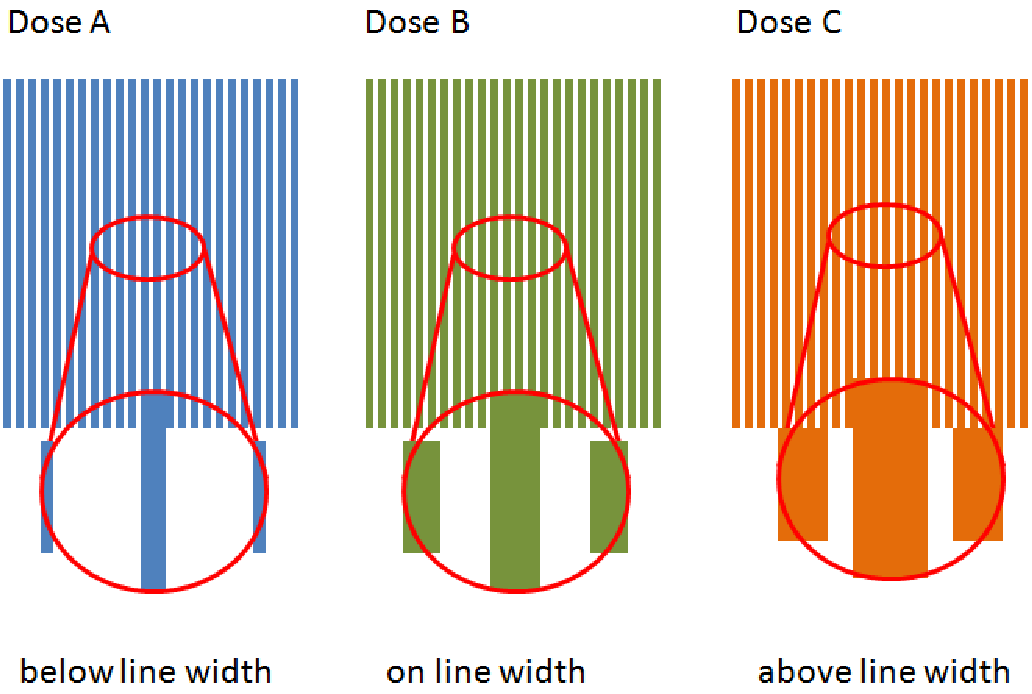
4. Close and re-open the VIEWER. Now whenever a corrected pattern is viewed and you click **Color-by-Dose**, the limits will automatically be set.

10.3.6 How do I determine the base dose for PEC?

BEAMER dose values are relative doses and thereby independent of the resist sensitivity. The dose reference point is called the *base dose* and is defined as the dose that results in correctly sized large features. This base dose needs to be determined for the different stacks and machine settings that are used in your process. The base dose is independent of layout design and is applicable for arbitrary layouts once determined.

To determine the base dose one can use a 1:1 line/space pattern with a line width of 100nm on a 200nm pitch. The size of the layout needs to be such, that proximity effects from the border are not seen anymore in the center of the test structure. In other words, the center of the pattern should be homogeneously sitting atop the backscattered energy. Typically a 120µm by 120µm area for bulk Si at 100kV will be sufficient. For 50kV, a 50µm by 50µm square of lines and spaces can be used instead.

This pattern needs to be exposed at different doses to obtain a line-width (at the center) vs dose curve. Inspection of the exposed structures at the center will reveal the base dose, which is the dose that results in equal lines and spaces.

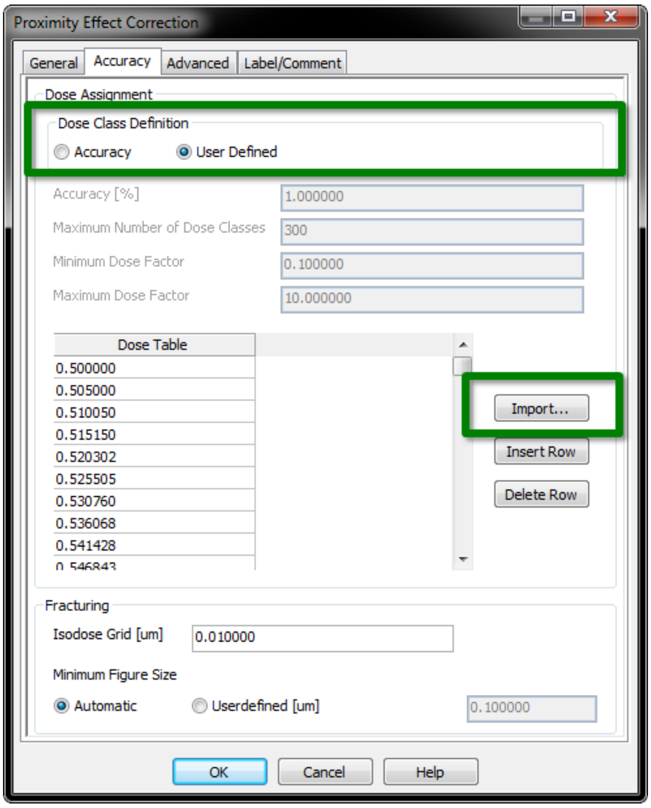


For this case then the dose B is identified to be used as base dose.

10.3.7 How can I use the same number of dose classes (or same shot modulation table) when correcting my pattern?

When using PEC on multiple patterns and exporting them as separate files, you should try using a predefined dose table in PEC to To begin, generate a text file of dose classes (call it doses.txt) using a spreadsheet. Simply enter the starting dose class (for example, 0.50) and set the next dose class 1% higher than the previous. Using a spreadsheet formula, a simple list of dose classes 1% change in dose can be created and saved.

In the PEC module, go to the Accuracy tab. Change the Dose Class Definition to User Defined and click “Import…” to import the doses.txt file, which will be loaded in the table to the left:



Dose Assignment

Dose Class Definition: ○ Accuracy ● User Defined

Accuracy [%]: 1.000000

Maximum Number of Dose Classes: 300

Minimum Dose Factor: 0.100000

Maximum Dose Factor: 10.000000

Dose Table
0.500000
0.505000
0.510050
0.515150
0.520302
0.525505
0.530760
0.536068
0.541428
0.546843

Fracturing

Isodose Grid [um]: 0.010000

Minimum Figure Size: ● Automatic ○ Userdefined [um] 0.100000

From there you can run your flow. You’ll discover that the shot modulation file (if needed for your tool) will always have the same dose classes specified in the PEC user defined dose table no matter what pattern you correct. Now you no longer have to worry about multiple modulation tables since the same modulation is used for all of your corrected patterns.

NOTE: If your tool can only use a fixed number of dose classes (say 256 dose classes), it will only grab the first 256 doses from the table for export.

10.3.8 What are the multi-threading capabilities in PEC?

BEAMER efficiently uses multi-threading when performing PEC. This sections briefly describes how this is done.

Short range PEC preserves the hierarchy in a layout. As such, the short range tiling algorithm will recognize a tile in the hierarchy and perform short range PEC on that tile. In parallel, other unique tiles in the hierarchy are also processed. If the same tile is repeated multiple times, the tiling algorithm will not process the repeated tile. To see hierarchical processing in action, change the Eta to 0; this turns off long range correction. Upon completion, inspect the layout. You will find that the short range PEC preserves the hierarchy.

Long range PEC works by processing the entire layout as one tile. The computation for this algorithm uses only one core.

10.3.9 Can you explain Periodic Repetition in PEC?

The periodic repetition option in the PEC module (to be activated at the Advanced tab) is for correction of very large arrays, where it is sufficient to correct the inner part of the large array.

The correction can be limited to just one cell assuming that this cell will be repeated in all direction unlimited times with the defined x- and y- pitch. The correction results is just the corrected cell.

It is possible to generate an array using the corrected cell by defining the number of cells in x- and y- direction.

Important: The edges of the new array with the corrected cell will not be properly corrected, as at the correction an unlimited array was assumed!

The application for this correction is a very large (some cm) field (e.g. of photonic crystal) where the edge of the field is not important.

10.3.10 3D PEC

The standard application in lithography is to generate a binary resist structure:

- Resist remains in full thickness in defined areas
- Resist is completely removed in other areas

The standard PEC makes sure that the X-Y dimensions of the resist structures fit to the layout dimensions. This is done by adjusting the energy levels of the exposure to a single defined value at all edges, which results to the correct dimension over the full layout at the correct base dose.

BEAMER supports corrections for two special 3D PEC applications in addition to the standard PEC:

1. 3D Surface PEC
2. 3D Edge PEC

10.3.10.1 3D Surface PEC

Description

The objective in a “3D Surface” application is to generate a 3 dimensional resist structure, meaning the thickness of the final resist structure (resist profile) is changing over the layout.

Example for “3D Surface” applications are:

- 3 dimensional optical elements like lenses, lens arrays, etc.
- 3D gratings, zone plates, 3D holograms, etc.
- 3D MEMS devices

These applications are usually based on a thick (e.g. 1µm) resist with a “flat” contrast curve (e.g. PMMA 950k). The single layer resist is exposed with a dose modulation to achieve a defined 3D profile in the resist.

The dependency of the development rate as a function of the absorbed energy (contrast curve) is used for adjusting the thickness of the resist over the layout. Resists with a low slope of contrast curve (low contrast) are preferred for 3D applications.

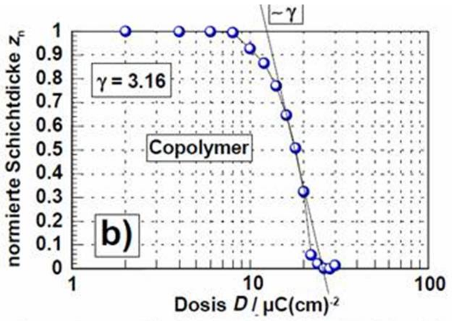


Abb. 4.12a,b: Kontrastkurven a) für Polyimid 408 (U_{acc} = 30 kV, I_B = 100 pA, t_{Spot} = 6.519 ns, d_{step} = 30.51 nm); b) für PMMA/MAA AR-P 610.08 (U_{acc} = 30 kV, I_B = 25 pA, t_{Spot} = 1.863 µs, d_{step} = 15.26 nm, T_{pre} = 210 °C).

The relationship of the resist thickness at a position and the required absorbed energy (final dose in the resist) is defined by the contrast curve.

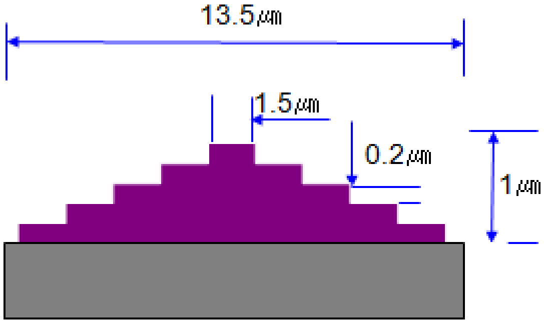
The absorbed energy after exposure is influenced by proximity effects during exposure. The proximity effects are depending on the resist stack and the exposure energy and are described by the Point-Spread-Function (PSF).

The “3D Surface PEC” function of BEAMER is correcting the proximity effects by adjusting the exposure dose to reach the defined absorbed energies (resist thickness) at any position of the layout.

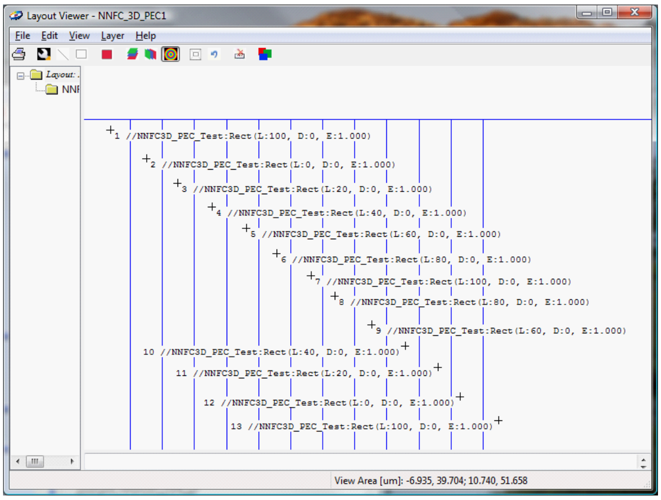
Function

The following example is demonstrated for the case:

Inversed stair type structure



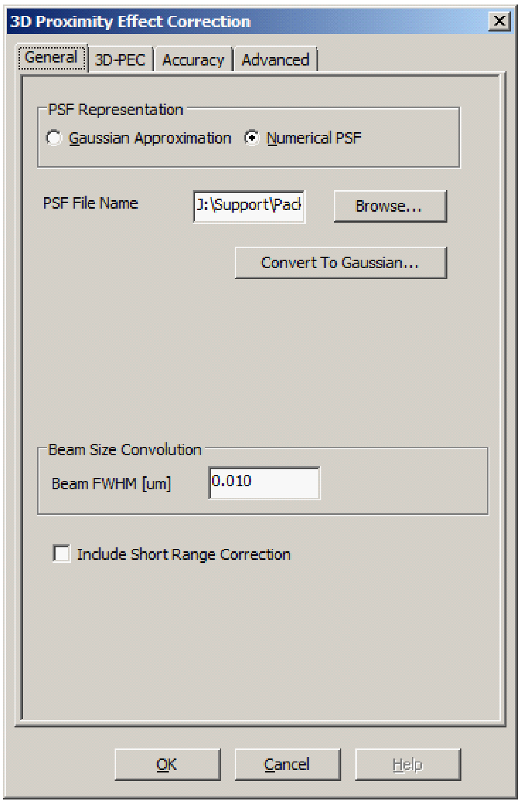
The input for the 3D Surface PEC is a layout with the target resist thickness (or target absorbed energy) in different layers or datatypes. The layout for the stair case example with the resist thickness in percentage in different layer (resist thickness 100% in layer 100, resist thickness 80% in layer 80,…):



```
1 //NNFC3D_PEC_Test:Rect(L:100, D:0, E:1.000)
2 //NNFC3D_PEC_Test:Rect(L:0, D:0, E:1.000)
3 //NNFC3D_PEC_Test:Rect(L:20, D:0, E:1.000)
4 //NNFC3D_PEC_Test:Rect(L:40, D:0, E:1.000)
5 //NNFC3D_PEC_Test:Rect(L:60, D:0, E:1.000)
6 //NNFC3D_PEC_Test:Rect(L:80, D:0, E:1.000)
7 //NNFC3D_PEC_Test:Rect(L:100, D:0, E:1.000)
8 //NNFC3D_PEC_Test:Rect(L:80, D:0, E:1.000)
9 //NNFC3D_PEC_Test:Rect(L:60, D:0, E:1.000)
10 //NNFC3D_PEC_Test:Rect(L:40, D:0, E:1.000)
11 //NNFC3D_PEC_Test:Rect(L:20, D:0, E:1.000)
12 //NNFC3D_PEC_Test:Rect(L:0, D:0, E:1.000)
13 //NNFC3D_PEC_Test:Rect(L:100, D:0, E:1.000)
```

View Area [um]: -6.935, 39.704; 10.740, 51.658

Following are the required settings for the 3D PEC module for BEAMER:



PSF Representation

○ Gaussian Approximation ● Numerical PSF

PSF File Name: J:\Support\Pacl

Beam Size Convolution

Beam FWHM [um]: 0.010

□ Include Short Range Correction

The “General” tab defines the proximity function as a Gaussian approximation or numerical PSF (e.g. import of a Monte Carlo simulation file). Refer to the PEC description section in the BEAMER manual for more detailed information on the proximity function.

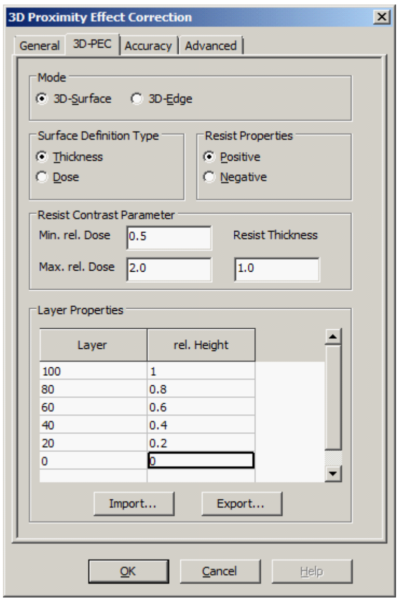
Short range (forward scattering) effects do not influence the 3D Edge mode, therefore “Include Short Range Correction should be switched off.

The “3D PEC” Tab starts with the adjustment of the “Mode”:

- 3D surface, described here: Defines the thickness of the resist (surface at any point)
- 3D Edge, described in other application note: Defines the dimensions at multiple edges (at multiple resist levels) for application like T-Gate or “Bridges”

The menu for “3D Surface” Mode offers 2 options for defining a 3D resist surface:

a) Resist Thickness with relative resist height in different layers:



Mode: ● 3D-Surface ○ 3D-Edge

Surface Definition Type: ● Thickness ○ Dose

Resist Properties: ● Positive ○ Negative

Resist Contrast Parameter

Min. rel. Dose: 0.5

Max. rel. Dose: 2.0

Resist Thickness: 1.0

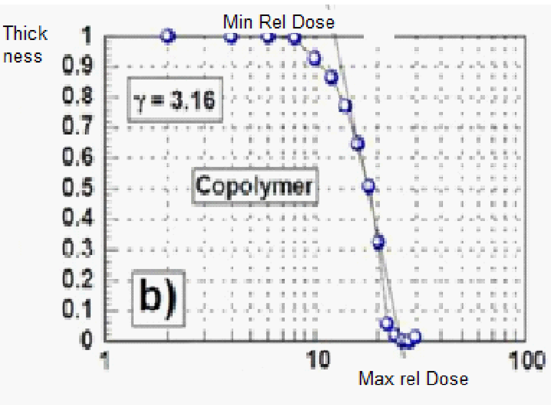
Layer Properties

Layer	rel. Height
100	1
80	0.8
60	0.6
40	0.4
20	0.2
0	0

Next step is to describe the resist by defining its type (negative or positive), the resist contrast curve parameter:

- Min relative dose:
  - For negative resist: Dose where the resist is “just” starting to develop
  - For positive resist: Dose where the resist is “just” starting to remain (dose to clear)
- Max relative dose:
  - For negative resist: Dose where the resist is “just” removed completely (dose to clear)
  - For positive resist: Dose where the resist is “just starting to remain (dose to??clear)
- Resist Thickness (from contrast curve) for 100% remaining

The values for the contrast curve example of the negative resist above are:

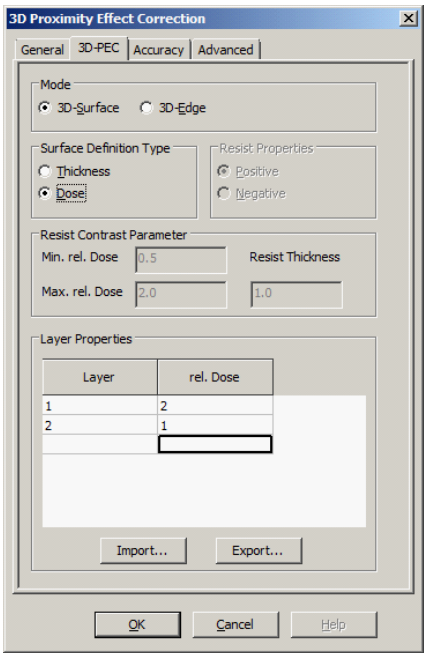


The next step is to define the relative target thickness for each layer.

BEAMER will calculate absorbed doses for the specified resist thickness using the contrast curve parameter and the approximation of a logarithmic behavior between the min and max relative dose.

b) Target absorbed energy (Dose) in different layers

A more accurate approach is to directly define the target absorbed energies for the required thicknesses from experimental data (measured contrast curve). The parameter menu for this case is as follows:



Mode
● 3D-Surface ○ 3D-Edge

Surface Definition Type
○ Thickness
● Dose

Resist Properties
● Positive
○ Negative

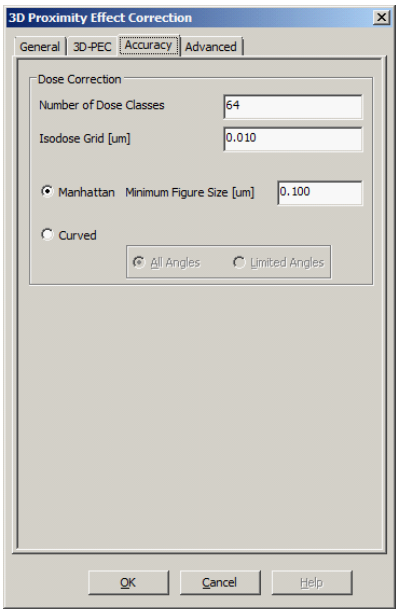
Resist Contrast Parameter
Min. rel. Dose 0.5
Max. rel. Dose 2.0
Resist Thickness 1.0

Layer Properties

Layer	rel. Dose
1	2
2	1

Only the definition of the target absorbed energy (Relative Dose) for each layer needs to be defined by entering the values manually or loading the table from a file by using the “Import” button. The file format is a simple text file, similar to the dose table (ldt) format.

The other Tabs in the 3D PEC menus are similar to those available in the standard PEC module menu. For a detailed explanation, go to the PEC base module library overview section in the manual:



Dose Correction

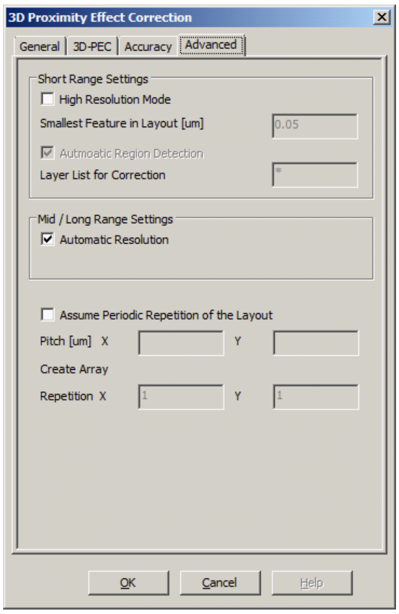
Number of Dose Classes: 64

Isodose Grid [um]: 0.010

● Manhattan   Minimum Figure Size [um]: 0.100

○ Curved

● All Angles   ○ Limited Angles



Short Range Settings

□ High Resolution Mode

Smallest Feature in Layout [um]: 0.05

☑ Autmoatic Region Detection

Layer List for Correction: *

Mid / Long Range Settings

☑ Automatic Resolution

□ Assume Periodic Repetition of the Layout

Pitch [um] X   Y

Create Array

Repetition X 1   Y 1

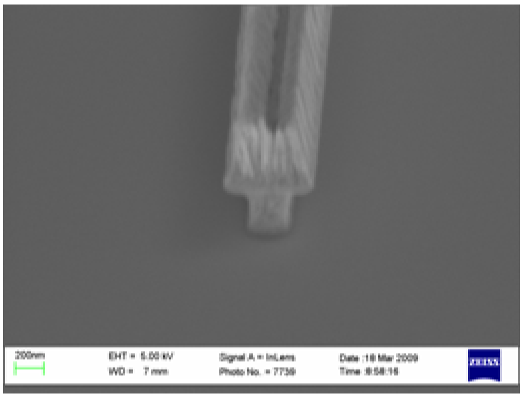
10.3.10.2 3D Edge PEC

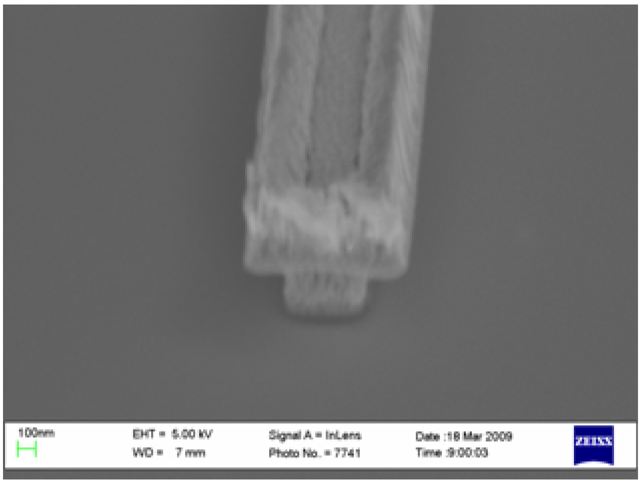
Description

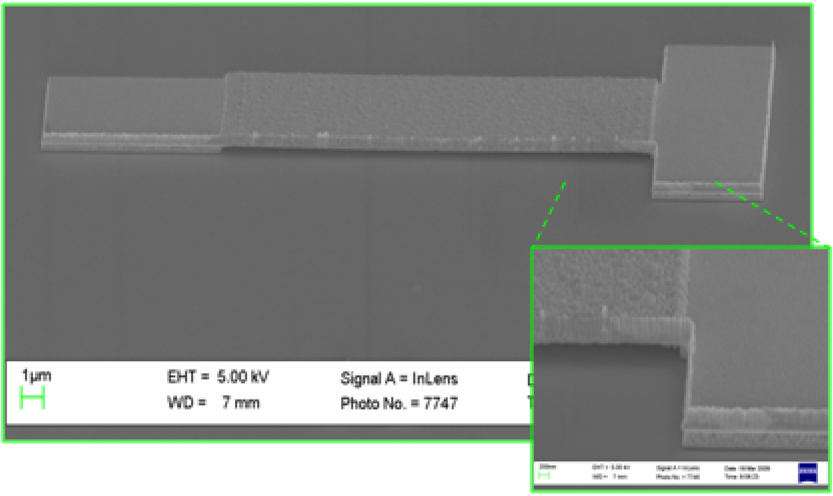
The task in “3D Edge” applications is also generating 3 dimensional resist structures by using multiple layer resists and a single exposure with dose modulation. The main objective in this application is to define the lateral dimension (CD) in different layers. The height is defined by the thickness of the resist layer.

Example for “3D Edge applications” are:

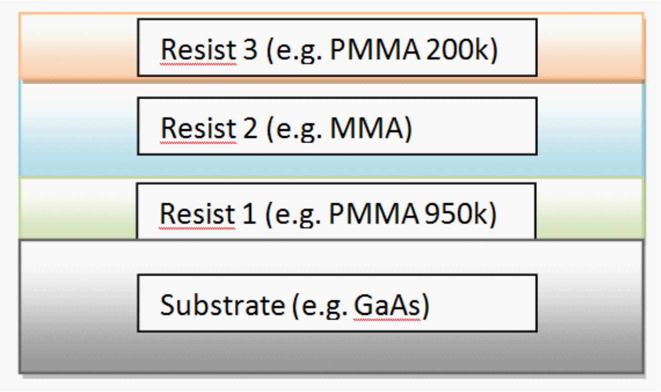
- T-Gate structures
- Bridges for interconnection







Usually double or triple layer resists are used:



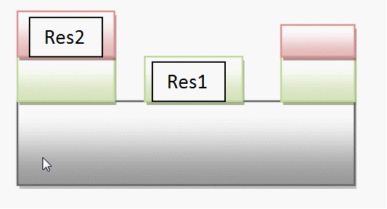
The resists used for the different layers have different sensitivities.

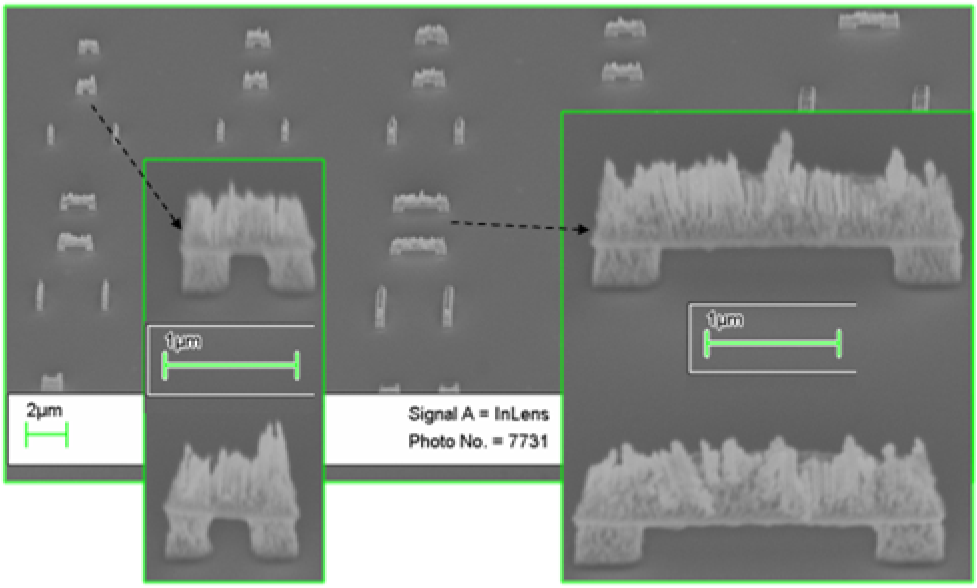
For the case above: Sensitivity (Resist 1) < Sensitivity (Resist 3) < Sensitivity (Resist 2)

The lateral dimension of each layer can now be controlled by adjusting absorbed energies at the edge to different values for each layer. BEAMER corrects both inter- and intra proximity effects.

Function

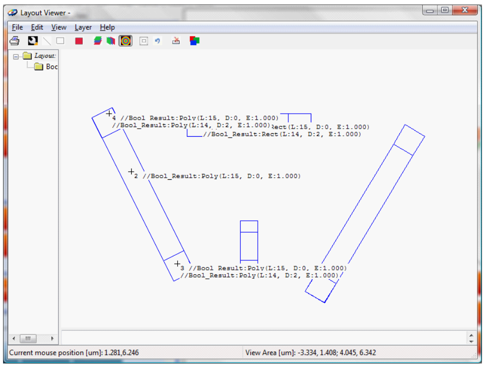
The bridge example is demonstrated for the case:



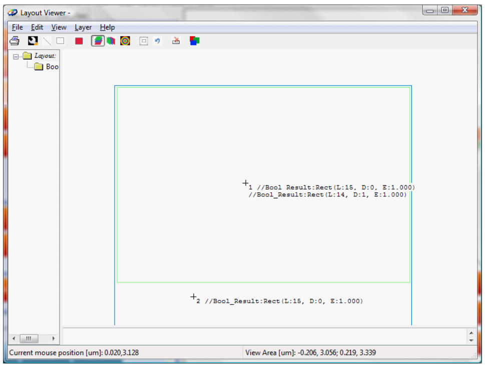


The double layer resist is:
Resist 1 = PMMA 950k with relative (target) dose = 1
Resist 2 = PMMA with relative (target) dose 3 = 3

This means that the sensitivity of PMMA is 1/3rd of Resist1. The relative (target) dose is the ratio of dose to size (or sensitivity) of the resist layer.
The input for the 3D Edge PEC is a layout with the layout of the different resist layers in different layers or datatypes. It is important that the elements (shapes) for the lower sensitivities are always inside (smaller) the ones for higher sensitivities. Example layout for the bridge is:

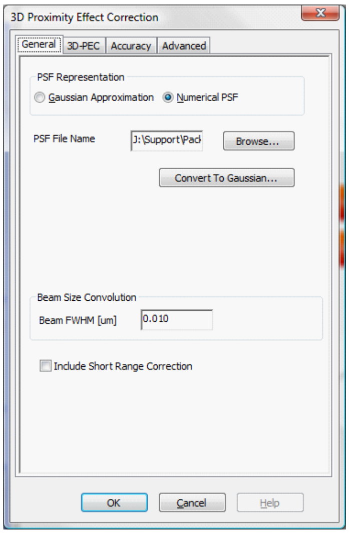

Layout Viewer -
File
Edit
View
Layer
Help
Layout:
Boc
4 //Bool Result:Poly(L:15, D:0, E:1.000)
//Bool_Result:Poly(L:14, D:2, E:1.000)
Rect(L:15, D:0, E:1.000)
//Bool_Result:Rect(L:14, D:2, E:1.000)
2 //Bool_Result:Poly(L:15, D:0, E:1.000)
3 //Bool Result:Poly(L:15, D:0, E:1.000)
//Bool_Result:Poly(L:14, D:2, E:1.000)

Current mouse position [um]: 1.281,6.246
View Area [um]: -3.334, 1.408; 4.045, 6.342


Layout Viewer -
File
Edit
View
Layer
Help
Layout:
Boo
1 //Bool Result:Rect(L:15, D:0, E:1.000)
//Bool_Result:Rect(L:14, D:1, E:1.000)
2 //Bool_Result:Rect(L:15, D:0, E:1.000)

Current mouse position [um]: 0.020,3.128
View Area [um]: -0.206, 3.056; 0.219, 3.339

Layer 14 is for Resist 1 and Layer 15 is for Resist 2. Settings of the 3D PEC module are:


3D Proximity Effect Correction

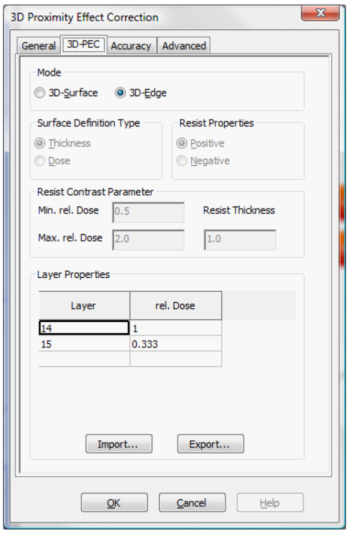
General | 3D-PEC | Accuracy | Advanced

PSF Representation
○ Gaussian Approximation ● Numerical PSF

PSF File Name: J:\Support\Pack

Beam Size Convolution
Beam FWHM [um]: 0.010

□ Include Short Range Correction


3D Proximity Effect Correction

General | 3D-PEC | Accuracy | Advanced

Mode
○ 3D-Surface ● 3D-Edge

Surface Definition Type
● Thickness
○ Dose

Resist Properties
● Positive
○ Negative

Resist Contrast Parameter
Min. rel. Dose: 0.5
Max. rel. Dose: 2.0
Resist Thickness: 1.0

Layer Properties

Layer	rel. Dose
14	1
15	0.333

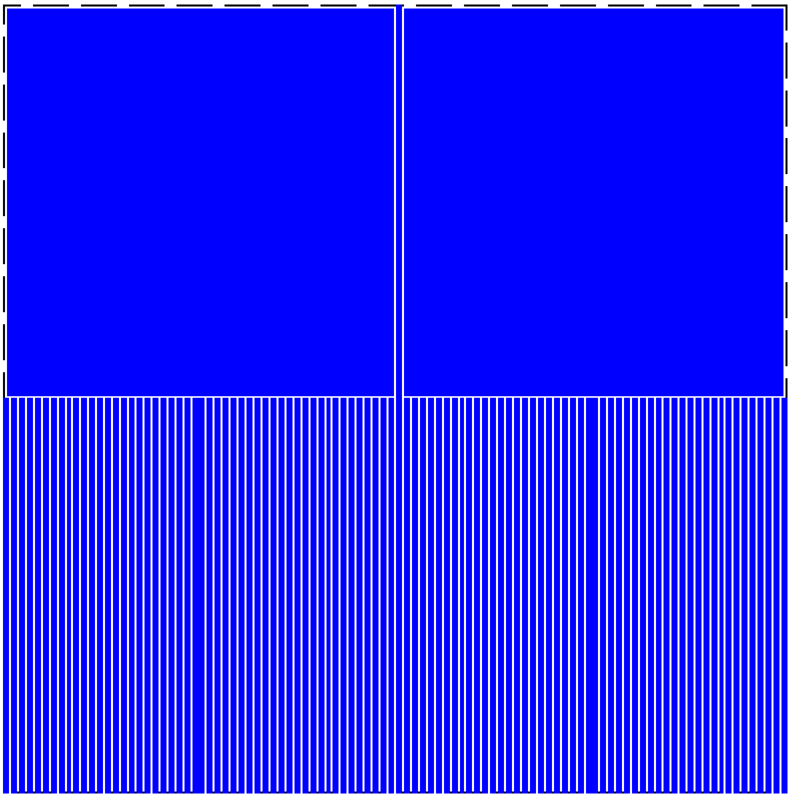
The General tab is for selecting the correction function. It is best to use a Monte Carlo simulation file. As the correction works with one PSF, it needs to be decided, at which z-position the PSF should be taken. It is recommended to take the PSF in the center of the resist layer with the most critical CDs. In this example case it would be the center of Resist 1 (foot of the bridge). The decision, if short range (forward scattering) is needed or not, depends on the dimensions of the layout and the short range blur (alpha * beam size).

The “3D-PEC” tab is for selecting the mode to “3D Edge” and defining the target doses for each layer. The relative dose needs to be entered to each layer accordingly or imported from a file. The settings of Accuracy and Advanced tab are equivalent to the standard PEC module (see the PEC section in the manual).

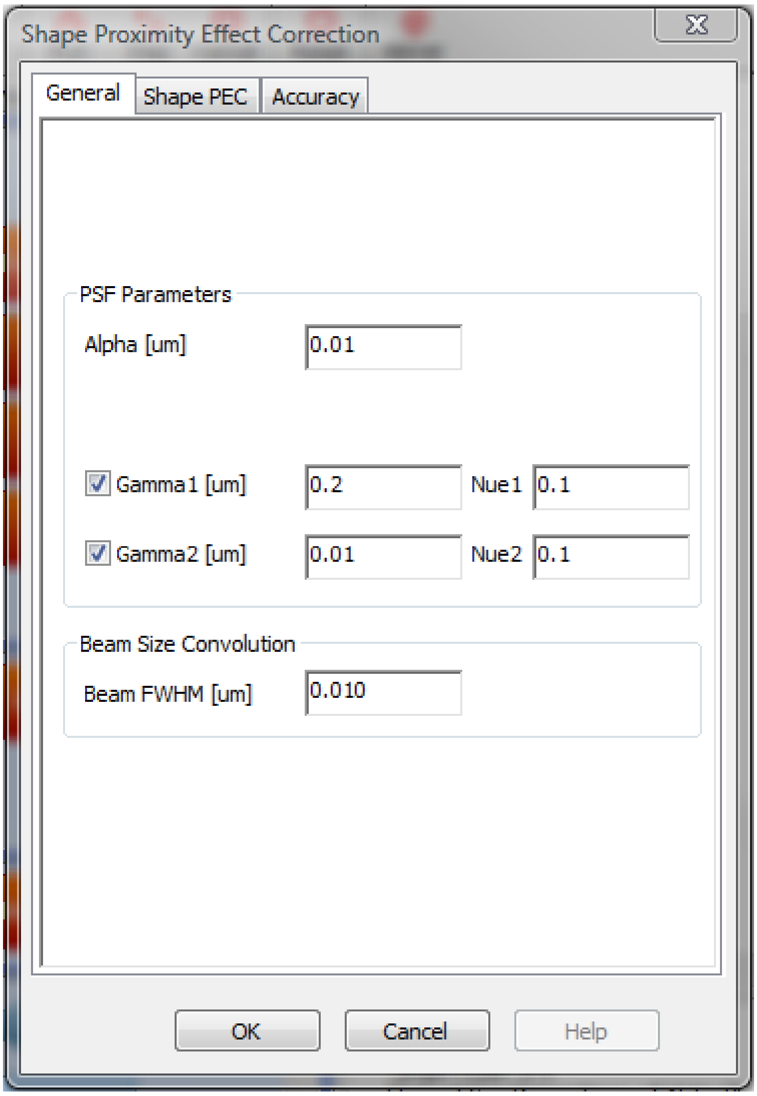
10.3.11 Shape PEC

Shape process effect correction is performing a shape modulation of the base layout to compensate for short and mid range effects in the process. Use cases for shape PEC include
1. E-Beam writers that only have a limited number of dose classes, and
2. asymmetrical process effects such as line end shortening.

The following example consists of 50nm lines and spaces with two large pads and an isolated line in the center.



The shape PEC has been set to these parameters:



PSF Parameters

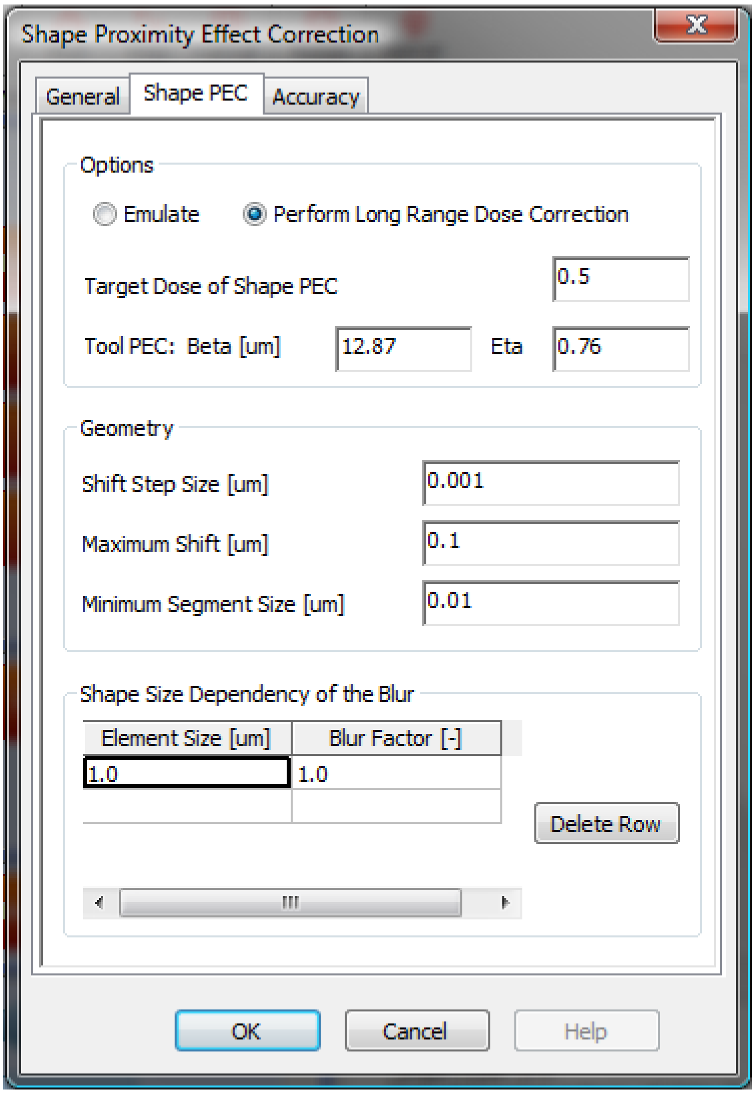
Alpha [um] 0.01

☑ Gamma1 [um] 0.2 Nue1 0.1

☑ Gamma2 [um] 0.01 Nue2 0.1

Beam Size Convolution

Beam FWHM [um] 0.010



Shape Proximity Effect Correction

Options

○ Emulate ● Perform Long Range Dose Correction

Target Dose of Shape PEC: 0.5

Tool PEC: Beta [um]: 12.87 Eta: 0.76

Geometry

Shift Step Size [um]: 0.001

Maximum Shift [um]: 0.1

Minimum Segment Size [um]: 0.01

Shape Size Dependency of the Blur

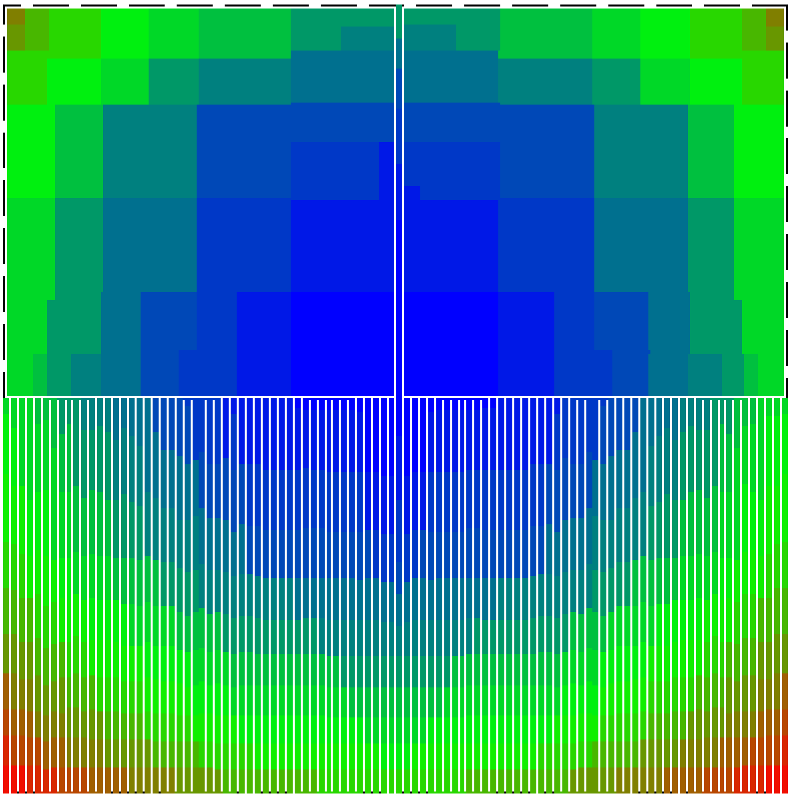
Element Size [um]	Blur Factor [-]
1.0	1.0

The Nue$_1$ and Nue$_2$ parameter are higher than the ones computed from the electron scattering function. This is to represent the influence of the process.

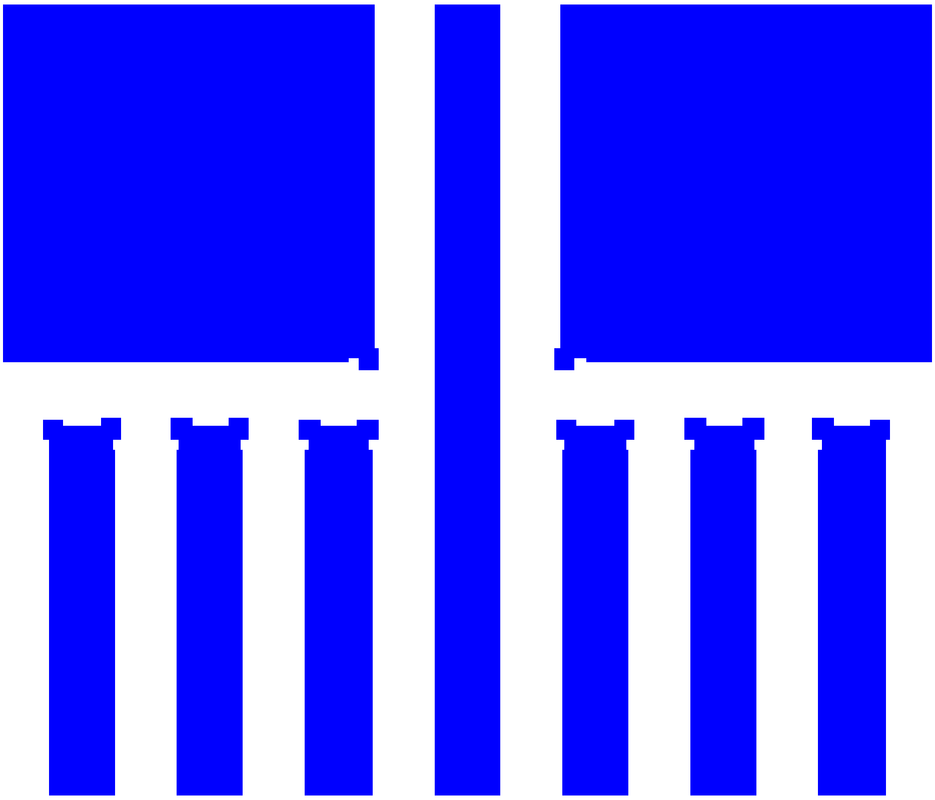
The minimum segment size is set to a low value to enable corrections at the corners of the elements. If this value is set too high then the corrections will be not very strong or not appear at all. This is due to the fact that the required edge shifts would not be doable with the set parameters.

As for the correction in the above example, the result would look like this for different regions of the layout.

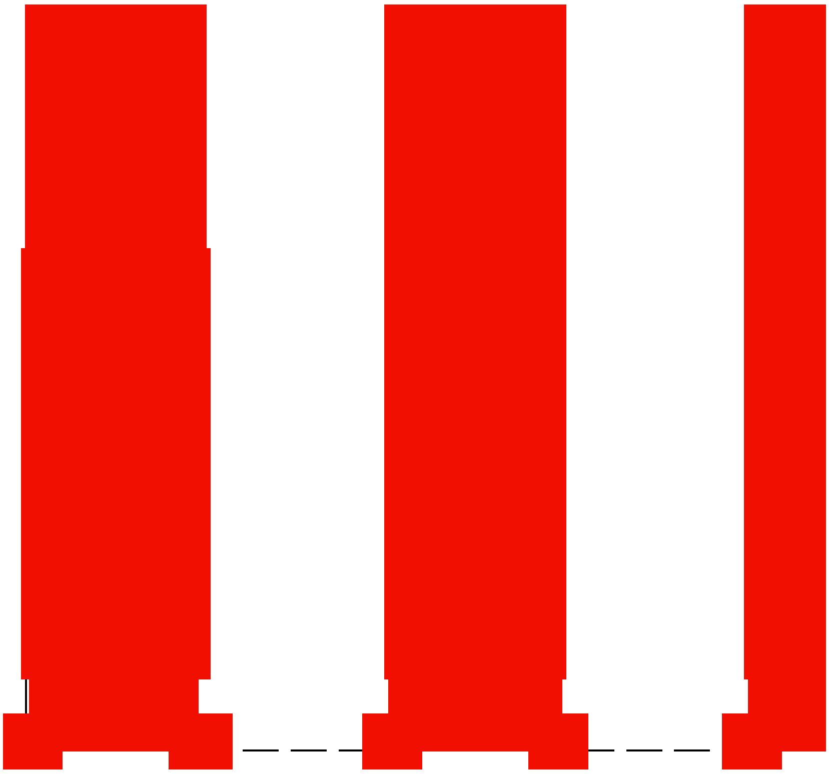
The dose modulation is coming from the long range terms of a GaAs substrate coated with HSQ resist.



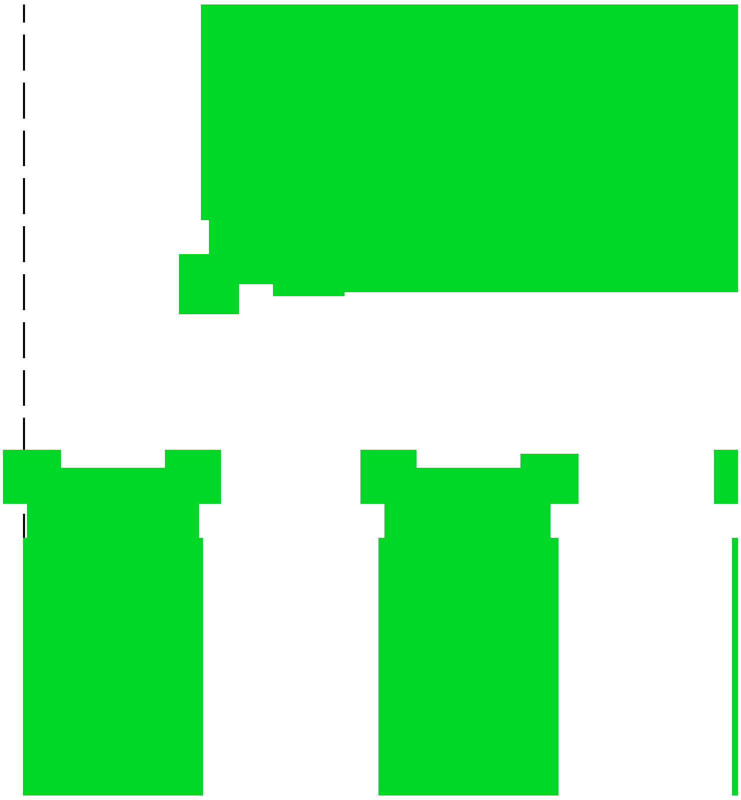
The center of the layout:



The lower left corner:



and the left corner center:



10.4 VIEWER

Contained herein are application examples for Layout VIEWER. For a more detailed description on its functionality, go to Layout VIEWER GUI.

10.4.1 Selection

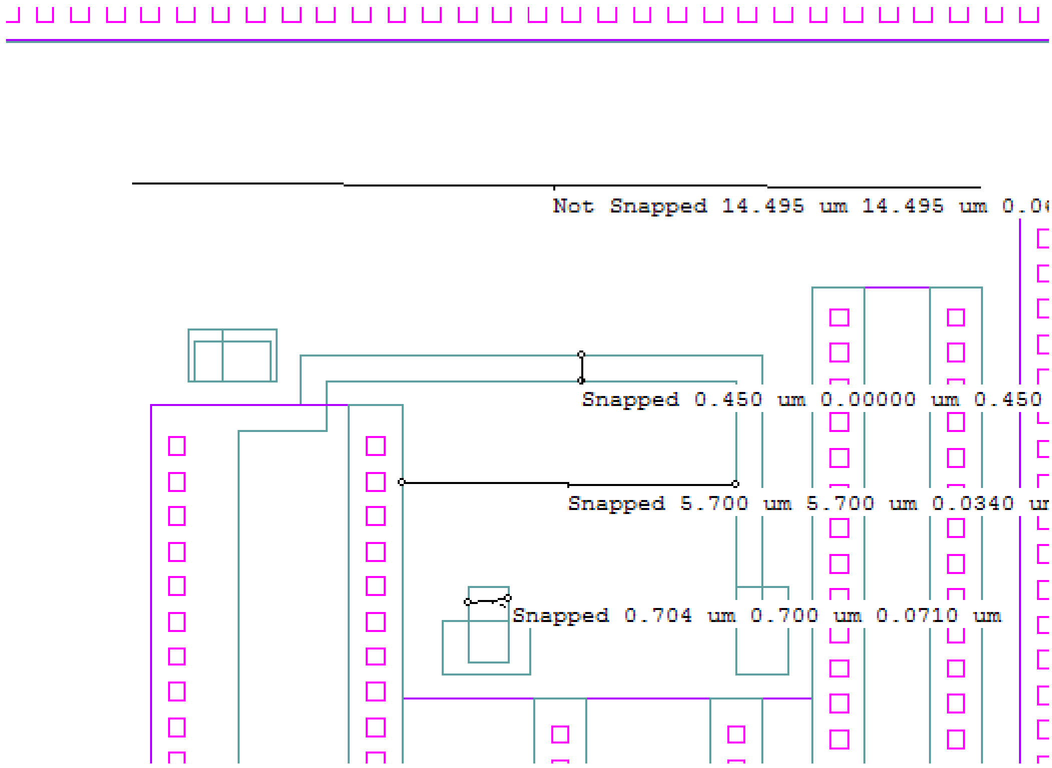
Layer selection is done with the menu at left side. Check boxes allow you to select and deselect layer individually. With the button All, all layers will be selected. Use Hide and Show to control their visibility.

10.4.2 Measuring

Measure Function:

To make a measurement, draw a line with two right mouse button clicks (beginning/end).

- Snapped measurement between two polygon lines – between the two clicks the length is measured
- Unsnapped measurement of the line length – if the start or end click is a defined number of pixels away from the layout border, the measurement will not snap to an edge. (See here for Snap Range).

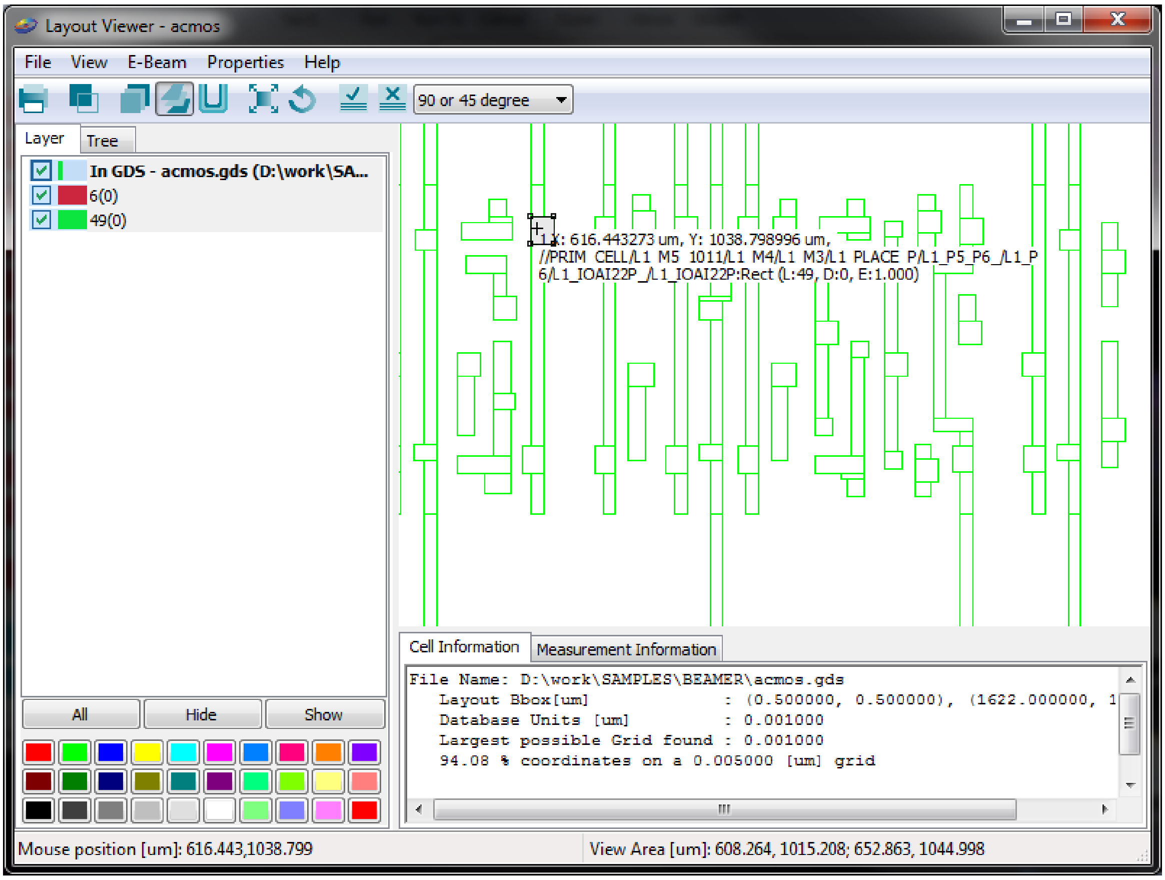


```
Not Snapped 14.495 um 14.495 um 0.0
Snapped 0.450 um 0.00000 um 0.450
Snapped 5.700 um 5.700 um 0.0340 ur
Snapped 0.704 um 0.700 um 0.0710 um
```

10.4.3 Picking

Pick Functions:

Use a right mouse click at a position to pick all polygons at this position. A pick also highlights the picked element with a grey transparent box and marks the polygon vertices with small boxes. Doing multiple picks will cause Layout VIEWER to measure between these points.



1 X: 616.443273 um, Y: 1038.798996 um,
//PRIM_CELL/L1_M5_1011/L1_M4/L1_M3/L1_PLACE_P/L1_P5_P6_/L1_P
6/L1_IOAI22P_/L1_IOAI22P:Rect (L:49, D:0, E:1.000)

```
File Name: D:\work\SAMPLES\BEAMER\acmos.gds
   Layout Bbox[um]             : (0.500000, 0.500000), (1622.000000, 1
   Database Units [um]         : 0.001000
   Largest possible Grid found : 0.001000
   94.08 % coordinates on a 0.005000 [um] grid
```

Mouse position [um]: 616.443,1038.799

View Area [um]: 608.264, 1015.208; 652.863, 1044.998

11 Python Scripting

BEAMER operations can be done using the Python scripting language. This provides a flexible environment for the user to run BEAMER in a command-line driven, programming environment. Scripts can be generated that are parameterized and can easily be reused. In Windows Python 2.6.5 is required. For Linux version 2.4.3 is required.

For details on the scripting language Python, visit the official website at http://www.python.org/.

11.1 General

Python scripting uses a class concept. Each module's content at first defines an object like myImport. In subsequent modules the object is referenced, so that a heal function would have a name like myHeal, and would use myImport as input.

Each module function can have default definitions. This is useful when dealing with modules that have a large set of parameters. For these a one time default is defined and then when calling the module you only need to adjust those parameters. Here is a sample using the Bias module:

```
bias_defaults( { 'Bias' : '0.100000', 'CornerExtension' : '1.000000',
               'TargetLayer' : '0', 'TargetDatatype' : '0', 'Mode' : 'X-
               Y', 'HierarchicalProcessing' : 'true' }  )
```

Here we see that the default target layer is 0. The defaults are defined by using the function name with a _default attached to it and the import object is omitted. Now using bias to process a layout but put the result on a different layer I just need to use:

```
myBias = myLB.bias ( myImport, {'TargetLayer' : 2})
```

This result will now be stored on layer 2 with a 0.1um bias applied to it.

11.2 Getting Starting

To use BEAMER within Python the environment needs to be set. The namespace and module need to be referenced in the Python environment. The library is located in the BEAMER installation directory within the binary folder called LBpy (with the OS typical extensions). The best method to do this is to use the sys.path.append option.

Windows:
sys.path.append('c:/program files/GenISys/BEAMER4.6.5/bin')

Linux:
sys.path.append('/homes/BEAMER_USER/BEAMER4.6.5/lib')

This loads the needs libraries to your Python script.

11.3 Samples

Below is a sample script written in Python. This script contains comments on the steps executed. A key point of Python is that objects (variables) are used to define new results. An imported file is stored in the object myImport. In a subsequent heal operation the object myHeal uses myImport for data.

Here is a sample:

```
# load a system library, this script needs an argument containing the path to the L
# part of the binaries

import sys

# add the path to the BEAMER to the system paths python uses
# FOR LINUX USE the BEAMER lib directory: sys.path.append('/home/BEAMER_USER/BEAMER
sys.path.append(/usr/myself/bin/GenISys/BEAMER/bin)

# load the BEAMER library
# for Python 2.4.3 you can use BEAMERpy23
# for Python 2.6.x you can use BEAMERpy26
# directly pointing to the required library
import BEAMERpy

# get an instance of an BEAMER
lb = BEAMERpy.GBEAMER()

# import an file
imp = lb.import_gds( { 'FileName' : '../layout_files/variousfigs.gds' } )

# heal the file
heal = lb.heal( imp )

# export to gpf
exp = lb.export_gpf( heal,{'FileName' : '../layout_files/variousfigs.gpf' })
```

11.4 Command Reference

This is a list of the currently available commands in the BEAMER Python interface.

11.4.1 number of cores

To define the number of cores to be used by the Python script please use the command set_number_of_threads at the beginning of your Python script.

```
import BEAMERpy
lb = BEAMERpy.GBEAMER()
lb.set_number_of_threads(16)
```

The example is set for 16 cores to be used.

11.4.2 bias

instance methods

```
GObject bias( **gobject**, { 'SoftFrame' : 0.3, 'Bias' : 0,
             'CornerExtension' : 1, 'TargetLayer' : '0(0)', 'Mode' : 'X-
             Y', 'HierarchicalProcessing' : True, 'LayerAssignment' :
             'AllLayer', 'SelectedLayerSet' : '*', 'BiasLayerList' : [] }
             )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
SoftFrame	float	
Bias	float	
CornerExtension	float	
TargetLayer	int	
HierarchicalProcessin	bool	true, false
Mode	enumerator	X-Y, X, Y
LayerAssignment	enumerator	‘AllLayer’,’PerLayer’,’LayerSpecific’
SelectedLayerSet		Per Layer
BiasLayerList		Layer Specific

11.4.3 bool_and

instance methods

```
GObject bool_and( **left_gobject**, **right_gobject**, { 'TargetLayer' :
'5(0)', 'SoftFrame' : 0.3 } )
```

The first two parameters must be GObjects. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
(left_/right_)**gobje	GObject	
TargetLayer	int	
SoftFrame	float	[um]

11.4.4 bool_not

instance methods

```
GObject bool_not( **gobject**, { 'TargetLayer' : '6(0)',
                 'HierarchicalProcessing' : True, 'SoftFrame' : 0.3,
                 'SelectedLayerSet' : '*', 'LayerAssignment' : 'AllLayer'
                 } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
TargetLayer	int	
HierarchicalProcessin	bool	
SoftFrame	float	[um]
SelectedLayerSet		Per Layer
LayerAssignment		‘AllLayer’, ‘PerLayer’

11.4.5 bool_or

instance methods

```
GObject bool_or( **left_gobject**, **right_gobject**, { 'TargetLayer' :
                '3(0)', 'SoftFrame' : 0.3 } )
```

The first two parameters must be GObjects. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values

(left_/right_)**gobje	GObject	
TargetLayer	int	
SoftFrame	float	[um]

11.4.6 bool_xor

instance methods

```
GObject bool_xor( **left_gobject**, **right_gobject**, { 'TargetLayer' :
                  '7(0)', 'SoftFrame' : 0 } )
```

The first two parameters must be GObjects. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
(left_/right_)**gobje	GObject	
TargetLayer	int	
SoftFrame	float	[um]

parameter	type	description/values
(left_/right_) **gobject**	GObject	
SoftFrame	float	
TargetLayer	int	
TargetDatatype	int	

11.4.7 ebeam

instance methods

```
GObject e_beam( **gobject**, { 'ExposureDose' : 1, 'Alpha' : 0.005,
               'Beta' : 30, 'Eta' : 0.6, 'Gamma1' : 0.3, 'Gamma2' : 2,
               'Nue1' : 0.01, 'Nue2' : 0.01, 'BeamSize' : 0.01,
               'BeamStepSize' : 0.01, 'Med1Correction' : True,
               'Med2Correction' : True,
               'UseFileParameterForShotSimulation' : False,
               'SimulateGaussianShots' : True, 'UseNumericalPSF' : False,
               'PSFFileName' : '', 'ManualResolution' : True, 'PixelSize'
               : 0.025, 'UseWholeLayout' : True, 'RegionLayerSet' : 100,
               'ExtractResistContours' : True, 'DiffusionLength' : 0,
               'PositiveResist' : True, 'Try1D' : True, 'LineExtension' :
               0.1, 'EtchBias' : 0, 'ResistEffects' : True,
               'ResistEffectInfluenceRange' : 1, 'BlurVariabiliy' : True,
               'EtchEffects' : True, 'EtchInfluenceRange' : 1,
               'EtchVisibleDirectionInside' : True, 'EtchEffects2' :
               True, 'EtchInfluenceRange2' : 1, 'EtchEffects3' : True,
               'EtchInfluenceRange3' : 1, 'AnalysisFileName' : '',
               'ThresholdList' : [0.500000], 'RegionList' : [],
               'ElementSizeBlurList' : [[1.000000, 1.000000]],
               'DensityDevRateList' : [[0.000000, 1.000000], [1.000000,
               1.000000]], 'ThresholdList' : [0.500000],
               'ElementSizeBlurList' : [[1.000000, 1.000000]],
               'EtchRateList' : [[0.000000, 1.000000], [1.000000,
               1.000000]], 'EtchRateList2' : [[0.000000, 1.000000],
               [1.000000, 1.000000]], 'EtchRateList3' : [[0.000000,
               1.000000], [1.000000, 1.000000]], 'EBeamType' : 'Simple
               Manhattan' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
ExposureDose	float	
Alpha	float	
Beta	float	
Eta	float	
Gamma1	float	
Gamma2	float	
Nue1	float	
Nue2	float	
BeamSize	float	
BeamStepSize	float	
PixelSize	float	
UseNumericalPSF	bool	'True','False'
PSFFileName	string	
ManualResolution	bool	'True','False'
UseWholeLayout	bool	'True','False'
RegionLayerSet	string	
ExtractResistContours	bool	'True','False'
DiffusionLength	float	
PositiveResist	bool	'True','False'
Try1D	bool	'True','False'
LineExtension	float	
EtchBias	float	
Med1Correction	bool	'True','False'
Med2Correction	bool	'True','False'
UseFileParameterForShotSimulati	bool	'True','False'
SimulateGaussianShots	bool	'True','False'
ResistEffects	bool	'True','False'

ResistEffectInfluenceRange	float	
ThresholdList	List	
RegionList	List	[[Xmin1,Ymin1,Xmax1,Ymax1], [X
ElementSizeBlurList	List	[[ElementSize1, Blur1], [Eleme
DensityDevRateList	List	[[Density1, DevelopmentRate1],
BlurVariabiliy	bool	‘True’,’False’
EtchEffects	bool	‘True’,’False’
EtchEffects2	bool	‘True’,’False’
EtchEffects3	bool	‘True’,’False’
EBeamType		‘SimpleManhattan’
EtchInfluenceRange	float	
EtchInfluenceRange2	float	
EtchInfluenceRange3	float	
EtchVisibleDirectionInside	bool	‘True’,’False’

11.4.8 export

11.4.8.1 export_cel

instance methods

```
GObject export_cel( **gobject**, { 'FileName' : './celfile.CEL',
                    'FieldSize' : 600, 'DotNumber' : 60000, 'BaseDoseTime'
                    : 2.5, 'HierarchicFileStructure' : False,
                    'GenerateDoseInfo' : True, 'FractureMode' :
                    'Conventional', 'ReplaceFiles' : False } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**
FieldSize	float	
DotNumber	int	
BaseDoseTime	float	
HierarchicFileStructur	bool	‘True’, ‘False’
GenerateDoseInfo	bool	‘True’, ‘False’
FractureMode	enumerator	‘Conventional’, ‘LRFT’
ReplaceFiles	bool	‘True’, ‘False’

11.4.8.2 export_cif

instance methods

```
GObject export_cif( **gobject**, { 'FileName' : '<filepath>' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**

11.4.8.3 export_cnf

instance methods

```
GObject export_cnf( **gobject**, { 'FileName' : ./CNFfile.cnf',
                    'ExtentMode' : 'Default', 'LowerLeftX' : 0,
                    'LowerLeftY' : 0, 'UpperRightX' : 0, 'UpperRightY' :
                    0, 'AddressUnits' : 1.25, 'FrameDataVolume' :
                    134217728, 'FrameCellLocationLimit' : 4000000,
                    'ApproximationAccuracy' : 10, 'Or2Compression' : True,
                    'ArrayCompression' : True, 'FrameHeight' : 512,
                    'CellWidth' : 128, 'CellHeight' : 128, 'CellMargin' :
                    0.25, 'BlockWidth' : 1024, 'BlockHeight' : 128,
                    'FractureMode' : 'LRFT' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**
ExtentMode	enumerator	‘Default’, ‘Minimum’, ‘User’
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
AddressUnits	float	
FrameDataVolume	int	
FrameCellLocationLimit	int	
ApproximationAccuracy	float	
Or2Compression	bool	‘True’, ‘False’
ArrayCompression	bool	‘True’, ‘False’
FrameHeight	float	

CellWidth	float	
CellHeight	float	
CellMargin	float	
BlockWidth	float	
BlockHeight	float	
FractureMode	enumerator	‘Conventional’, ‘LRFT’

11.4.8.4 export_fre

instance methods

```
GObject export_fre( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                  \22\22222.fre', 'ExtentMode' : 'Default', 'LowerLeftX'
                  : 0, 'LowerLeftY' : 0, 'UpperRightX' : 0,
                  'UpperRightY' : 0, 'FieldSizeX' : 650, 'FieldSizeY' :
                  650, 'Resolution' : 0.01, 'MainfieldResolution' :
                  0.01, 'VRU' : 1, 'NormalizeToOne' : False,
                  'PreserveDoseRange' : True, 'ExtendedLogfile' : False,
                  'BssAlignment' : 'NONE', 'Voltage' : 20,
                  'FractureMode' : 'LRFT', 'EnableCompaction' : True,
                  'CurveTolerance' : 1, 'RegionList' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**
ExtendMode	enumerator	‘Default’, ‘Minimum’, ‘User’, ‘Symmetric’
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
FieldSizeX	float	
FieldSizeY	float	
VRU	float	
FieldOverlapX	float	
FieldOverlapY	float	
Resolution	float	
MainFieldResolution	float	
BeamStepSize	float	
Voltage	float	
CurveTolerance	float	
RegionLayerSet	string	

BSSAlignment	enumerator	‘NONE’, ‘Y_ONLY’, ‘X_AND_Y’
NormalizeToOne	bool	‘True’, ‘False’
PreserveDoseRange	bool	‘True’, ‘False’
ExtendedLogfile	bool	‘True’, ‘False’
EnableCompaction	bool	‘True’, ‘False’
FractureMode	enumerator	‘Conventional’, ‘LRFT’, ‘Curved’
RegionList	list	[[X1, Y1], [X2, Y2],...]

11.4.8.5 export_gdr

instance methods

```
GObject export_gdr( **gobject**, { 'FileName' : '<filepath>' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**

11.4.8.6 export_gds

instance methods

```
GObject export_gdr( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                   \22\GDSfile.gds', 'LayerSet' : '*', 'FlattenLeaves' :
                   0, 'SubtreeSelected' : False, 'MaximumPolygonVertices'
                   : 1024 } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	Gobject	
FileName	string	**filepath**
DoseMapping	enumerator	‘Property’, ‘Layer’, ‘Datatype’, Voi
LayerSet	string	
FlattenLeaves	float	
SubtreeSelected	bool	‘True’, ‘False’
MaximumPolygonVertices	int	

11.4.8.7 export_gpf

instance methods

```
GObject export_gpf( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                   \22\GPFfile.gpf', 'ExtentMode' : 'Minimum',
                   'LowerLeftX' : 0, 'LowerLeftY' : 0, 'UpperRightX' : 0,
                   'UpperRightY' : 0, 'FormatType' : '5200 / 5000+ 20bit
                   HS 100kV', 'GridResolution' : 0.01,
                   'MainFieldResolution' : 0.001, 'SubFieldResolution' :
                   0.0005, 'BeamStepSize' : 0.01, 'MainfieldSizeX' :
                   1040, 'MainfieldSizeY' : 1040, 'SubfieldSizeX' :
                   4.525, 'SubfieldSizeY' : 4.525, 'Compaction' : True,
                   'MainfieldPlacement' : 'Fixed', 'YTrapezoids' : True,
                   'NormalizeToOne' : True, 'TrapezoidDoseCorrection' :
                   True, 'ParallelogramCompaction' : True,
                   'OptimizeWritingOrder' : False,
                   'WritingOrderSeleciton' : '*', 'SequenceFile' : '',
                   'FractureMode' : 'LRFT', 'RegionLayerSet' : '',
                   'BeamStepSizeFracturing' : True, 'CurveTolerance' : 1,
                   ' AreaSelection ' : 'SelectedThenRemainder',
                   'FieldOverlapX' : 0, 'FieldOverlapY' : 0,
                   'OverlapMethod' : 'Standard', 'InterleavingSize' : 0,
                   'InterlockLayer' : '*', 'MultipassMode' : 1,
                   'MainfieldOffsetX' : 0.5, 'MainfieldOffsetY' : 0.5,
                   'SubfieldOffsetX' : 0, 'SubfieldOffsetY' : 0,
                   'MultipassLayer' : '*', 'UserDosePass' : [],
                   'FieldOverlapX' : 0, 'FieldOverlapY' : 0,
                   'OverlapMethod' : 'Standard', 'InterleavingSize' : 0,
                   'InterlockLayer' : '*', 'MultipassMode' : 1,
                   'MainfieldOffsetX' : 0.5, 'MainfieldOffsetY' : 0.5,
                   'SubfieldOffsetX' : 0, 'SubfieldOffsetY' : 0,
                   'MultipassLayer' : '*', 'UserDosePass' : [],
                   'RegionList' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
`**gobject**`	`GObject`	
`FileName`	`string`	`**filepath**`
`ExtentMode`	`enumerator`	`‘Default’, ‘Minimum’, ‘User’, ‘`
`LowerLeftX`	`float`	
`LowerLeftY`	`float`	
`UpperRightX`	`float`	
`UpperRightY`	`float`	
`FormatType`	`string`	
`GridResolution`	`float`	
`BeamStepSize`	`float`	

FieldOverlapX	float	
FieldOverlapY	float	
MainfieldSizeX	float	
MainfieldSizeY	float	
SubfieldSizeX	float	
SubfieldSizeY	float	
SubfieldNumDACBits	int	
SubfieldMaxMSF	int	
AreaSelection	string	
SequenceFile	string	
MainfieldNumDACBits	int	
MainfieldResolution	float	
SubfieldResolution	float	
MaxSubfieldSize	float	
MaxMainfieldSize	float	
Voltage	enumerator	20 kV, 50 kV, 100 kV
FormatVersion	enumerator	1.39, 1.40
Optimization	enumerator	NONE, FILE_SIZE, WRITING_TIME
MainfieldPlacement	enumerator	RANDOM, MEANDER
IwflClassification	enumerator	MERGE_CLASSES, MULTI_EXPOSURE
YTrapezoids	bool	
NormalizeToOne	bool	
TrapezoidDoseCorrectio	bool	
ParallelogramCompactio	bool	
FractureMode	enumerator	‘Conventional’, ‘LRFT’, ‘Curved’
UserDosePass	list	
RegionList	list	[[X1, Y1], [X2, Y2],…]
MultipassLayer	string	
MainfieldOffsetX	float	
MainfieldOffsetY	float	
SubfieldOffsetX	float	
SubfieldOffsetY	float	
InterlockLayer	string	
InterleavingSize	int	
OverlapMethod	enumerator	‘Standart’, ‘Interleaving + ext
BeamStepSizeFracturing	bool	‘True’,’False’

11.4.8.8 export_j51

instance methods

```
GObject export_j51( **gobject**, { 'FileName' : '<filepath>',
                   'ExtentMode' : 'Default', 'LowerLeftX' : '0.000000',
                   'LowerLeftY' : '0.000000', 'UpperRightX' : '0.000000',
                   'UpperRightY' : '0.000000', 'FieldSizeX' :
                   '1600.000000', 'FieldSizeY' : '1600.000000',
                   'FieldOverlapX' : '0.000000', 'FieldOverlapY' :
                   '0.000000', 'MachineType' : 'JBX-5DII', 'Resolution' :
                   '25', 'MemorySize' : '256', 'SubfieldSizeX' :
                   '100.000000', 'SubfieldSizeY' : '100.000000',
                   'EOSMode' : '1', 'Frequency' : '2', 'CenterToField' :
                   'false', 'NormalizeDose' : 'false',
                   'PreserveDoseRange' : 'true', 'UseCompaction' :
                   'true', 'CorrectLines' : 'false',
                   'ReservedSingleLineClasses' : '0', 'FractureMode' :
                   'Mode1' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
FieldSizeX	float	
FieldSizeY	float	
FieldOverlapX	float	
FieldOverlapY	float	
MachineType	enumerator	JBX-5DII, JBX-5DII(U), JBX-5DII(F), JBX-5000LS, JBX-5FE, J
Resolution	int	
MemorySize	int	
SubfieldSizeX	float	
SubfieldSizeY	float	
EOSMode	int	1, 2,..., 8
Frequency	int	2, 6, 12 (in MHz)
CenterToField	bool	
NormalizeDose	bool	
PreserveDoseRange	bool	

UseCompaction	bool	
CorrectLines	bool	
ReservedSingleLineClasse	int	
FractureMode	enumerator	‘Conventional’, ‘LRFT’

11.4.8.9 export_ldb

instance methods

```
GObject export_ldb( **gobject**, { 'FileName' : '<filepath>' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**

11.4.8.10 export_meb

instance methods

```
GObject export_meb( **gobject**, { 'FileName' : '<filepath>',
                   'ExtentMode' : 'Default', 'LowerLeftX' : '0.000000',
                   'LowerLeftY' : '0.000000', 'UpperRightX' : '0.000000',
                   'UpperRightY' : '0.000000', 'StripeHeight' : '512',
                   'AddressUnits' : '0.500000', 'MaskInfo' : 'Layout
                   BEAMER optimizing MEBES compiler', 'FractureMode' :
                   'Mode1' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
StripeHeight	int	
AddressUnits	float	
MaskInfo	string	
FractureMode	enumerator	‘Conventional’, ‘LRFT’

11.4.8.11 export_mgs

instance methods

```
GObject export_mgs( **gobject**, { 'FileName' : '<filepath>' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
`**gobject**`	`GObject`	
`FileName`	`string`	`**filepath**`

11.4.8.12 export_png

instance methods

```
GObject export_png( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                  \22\PNGFile.png', 'PixelPitch' : 1, 'ExposureDose' :
                  1, 'ApertureDimension' : 0, 'BeamBlur' : 0,
                  'Annotation' : 'ann', 'NumOfGreyScaleValues' : 256,
                  'RelGreyScaleValue' : 1, 'ApplyCornerSharpening' :
                  True, 'CornerCorrectionAveraging' : False,
                  'DitheringRange' : 1, 'AreaDither' : True } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
`**gobject**`	`GObject`	
`FileName`	`string`	`**filepath**`
`PixelPitch`	`float`	
`ExposureDose`	`float`	
`ApertureDimension`	`float`	
`BeamBlur`	`float`	
`Annotation`	`string`	
`NumOfGreyScaleValues`	`int`	
`RelGreyScaleValue`	`float`	
`ApplyCornerSharpening`	`bool`	`‘True’, ‘False’`
`CornerCorrectionAveragir`	`bool`	`‘True’, ‘False’`
`DitheringRange`	`int`	`1, 2, 4`
`AreaDither`	`bool`	`‘True’, ‘False’`

11.4.8.13 export_sdf

instance methods

```
GObject export_sdf( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                    \22\SDF_File.sdf', 'ExtentMode' : 'Default',
                    'LowerLeftX' : 0, 'LowerLeftY' : 0, 'UpperRightX' : 0,
                    'UpperRightY' : 0, 'Substrate' : 'GP1100mmx1100mm',
                    'WriteLens' : 4, 'StripeWidth' : 800,
                    'GenerateLicFiles' : False, 'ReplaceFiles' : False,
                    'FractureMode' : 'Conventional', 'CurveTolerance' :
                    1 } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**
ExtendMode	enumerator	‘Default’, ‘Minimum’, ‘User’
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
Substrate	enumerator	
WriteLens	int	2, 4, 5, 10, 20
StripeWidth	int	
GenerateLicFiles	bool	‘True’, ‘False’
ConvertCircles	bool	‘True’, ‘False’
PixelDensity	int	
PixelSubresolution	int	
BlockSize	int	
MirrorStripeY	bool	‘True’, ‘False’
ReplaceFiles	bool	‘True’, ‘False’
FractureMode	enumerator	‘Conventional’, ‘LRFT’, ‘Curved’
CurveTolerance	int	

11.4.8.14 export_txl

instance methods

```
GObject export_txl( **gobject**, { 'FileName' : '<filepath>' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**

11.4.8.15 export_v30

instance methods

```
GObject export_v30( **gobject**, { 'FileName' : '<filepath>',
                    'ExtentMode' : 'Default', 'LowerLeftX' : '0.000000',
                    'LowerLeftY' : '0.000000', 'UpperRightX' : '0.000000',
                    'UpperRightY' : '0.000000', 'FieldSizeX' :
                    '1000.000000', 'FieldSizeY' : '1000.000000',
                    'FieldOverlapX' : '0.000000', 'FieldOverlapY' :
                    '0.000000', 'MachineType' : 'JBX-9300FS (100kV)',
                    'Resolution' : '1', 'MemorySize' : '256', 'EOSMode' :
                    '1', 'CenterToField' : 'false', 'NormalizeDose' :
                    'false', 'PreserveDoseRange' : 'true', 'UseCompaction'
                    : 'true', 'CorrectLines' : 'false',
                    'ReservedSingleLineClasses' : '0', 'FractureMode' :
                    'Mode1' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
FieldSizeX	float	
FieldSizeY	float	
FieldOverlapX	float	
FieldOverlapY	float	
MachineType	string	
Resolution	string(int)	

MemorySize	int	
SubfieldSizeX	float	
SubfieldSizeY	float	
EOSMode	string(int)	
Frequency	string(int)	
CenterToField	bool	'True', 'False'
NormalizeDose	bool	'True', 'False'
PreserveDoseRange	bool	'True', 'False'
UseCompaction	bool	'True', 'False'
CorrectLines	bool	'True', 'False'
ReservedSingleLineClasse	int	
FractureMode	enumerator	'Conventional', 'LRFT', 'Curved'

11.4.8.16 export_vep

instance methods

```
GObject export_vep( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                    \22\VEPfile.vep', 'ExtentMode' : 'Default',
                    'LowerLeftX' : 0, 'LowerLeftY' : 0, 'UpperRightX' : 0,
                    'UpperRightY' : 0, 'FieldSizeX' : 1300, 'FieldSizeY' :
                    1300, 'Resolution' : 0.01, 'MainfieldResolution' :
                    0.005, 'VRU' : 1, 'RegionLayerSet' : '',
                    'NormalizeToOne' : False, 'PreserveDoseRange' : True,
                    'ExtendedLogfile' : False, 'BssAlignment' : 'NONE',
                    'FieldOrder' : 'Fixed', 'ClockConversion' : 'Dose',
                    'LensType' : 'VB6-UHR EWF', 'DACRange' : 18,
                    'SubfieldDacBits' : 12, 'SubfieldAlignment' :
                    'CornerToCenter', 'CenterToSubfield' : False,
                    'SubfieldFracturing' : False, 'Voltage' : 100,
                    'FractureMode' : 'LRFT', 'EnableCompaction' : True,
                    'CurveTolerance' : 1, 'AreaSelection' :
                    'SelectedThenRemainder', 'TraversalPath' :
                    'MinimizeDistances', 'FieldOverlapX' : 0,
                    'FieldOverlapY' : 0, 'OverlapMethod' : 'Standard',
                    'InterleavingSize' : 0, 'InterlockLayer' : '*',
                    'MultipassMode' : 1, 'MainfieldOffsetX' : 0.5,
                    'MainfieldOffsetY' : 0.5, 'SubfieldOffsetX' : 0,
                    'SubfieldOffsetY' : 0, 'MultipassLayer' : '*',
                    'UserDosePass' : [], 'RegionList' : [] } )
```

The first parameter must be a Gobject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	Gobject	

FileName	string	**filepath**
ExtentMode	enumerator	‘Default’, ‘Minimum’, ‘User’, ‘Symme
LensType	enumerator	‘VB6-HR’, ‘VB6-UHR’, ‘VB6-HR EWF’, ‘
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
FieldSizeX	float	
FieldSizeY	float	
FieldOverlapX	float	
FieldOverlapY	float	
Resolution	float	
MainfieldResolution	float	
VRU	float	
BeamStepSize	float	
RegionLayerSet	string(csv)	
NormalizeToOne	bool	‘True’, ‘False’
PreserveDoseRange	bool	‘True’, ‘False’
ExtendLogFile	bool	‘True’, ‘False’
BssAlignment	string	
FieldOrder	string	
ClockConversion	string	
DACRange	float	
SubfieldDacBits	int	
SubfieldAlignment	string	
CenterToSubfield	bool	‘True’, ‘False’
SubfieldFracturing	bool	‘True’, ‘False’
EnableCompaction	bool	‘True’, ‘False’
Voltage	float	
FractureMode	enumerator	‘Conventional’, ‘LRFT’, ‘Curved’
CurveTolerance	float	
AreaSelection	string	
TraversalPath	string	
OverlapMethod	enumerator	‘Standart’, ‘Interleaving + extra fi
InterleavingSize	float	
InterlockLayer	string	
MultipassMode	int	
MainfieldOffsetX	float	

MainfieldOffsetY	float	
SubfieldOffsetX	float	
SubfieldOffsetY	float	
MultipassLayer	string	
UserDosePass	list	
RegionList	list	[[X1,Y1],[X2,Y2],…]

11.4.8.17 export_wfl

instance methods

```
GObject export_wfl( **gobject**, { 'FileName' : 'C:\Users\bengi\Desktop
                  \22\IWFLfile.iwfl', 'ExtentMode' : 'Default',
                  'LowerLeftX' : 0, 'LowerLeftY' : 0, 'UpperRightX' : 0,
                  'UpperRightY' : 0, 'FormatType' : '45HRV 50kV',
                  'GridResolution' : 0.01, 'MainFieldResolution' : 0.01,
                  'SubFieldResolution' : 0.01, 'BeamStepSize' : 0.01,
                  'MainfieldSizeX' : 320, 'MainfieldSizeY' : 320,
                  'SubfieldSizeX' : 2.49, 'SubfieldSizeY' : 2.49,
                  'Compaction' : True, 'IwflClassification' :
                  'MERGE_CLASSES', 'NormalizeToOne' : False,
                  'TrapezoidDoseCorrection' : False,
                  'ParallelogramCompaction' : False,
                  'OptimizeWritingOrder' : False,
                  'WritingOrderSeleciton' : '*', 'SequenceFile' : '',
                  'FractureMode' : 'LRFT', 'BeamStepSizeFracturing' :
                  False, 'CurveTolerance' : 1, 'FieldOverlapX' : 0,
                  'FieldOverlapY' : 0, 'OverlapMethod' : 'Standard',
                  'InterleavingSize' : 0, 'InterlockLayer' : '*',
                  'MultipassMode' : 1, 'MainfieldOffsetX' : 0,
                  'MainfieldOffsetY' : 0, 'SubfieldOffsetX' : 0,
                  'SubfieldOffsetY' : 0, 'MultipassLayer' : '*',
                  'UserDosePass' : [], 'RegionList' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
FileName	string	**filepath**
ExtendMode	enumerator	‘Default’, ‘Minimum’, ‘User’,
FormatType	enumerator	
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	
GridResolution	float	

BeamStepSize	float	
FieldOverlapX	float	
FieldOverlapY	float	
MainfieldSizeX	float	
MainfieldSizeY	float	
SubfieldSizeX	float	
SubfieldSizeY	float	
SubfieldNumDACBits	int	
SubfieldMaxMSF	int	
MainfieldNumDACBits	int	
MainfieldResolution	float	
SubfieldResolution	float	
MaxSubfieldSize	float	
MaxMainfieldSize	float	
Voltage	enumerator	20 kV, 50 kV, 100 kV
Optimization	enumerator	NONE, FILE_SIZE, WRITING_TIME
IwflClassification	enumerator	MERGE_CLASSES, MULTI_EXPOSURE
YTrapezoids	bool	'True', 'False'
NormalizeToOne	bool	'True', 'False'
TrapezoidDoseCorrection	bool	'True', 'False'
ParallelogramCompaction	bool	'True', 'False'
OptimizeWritingOrder	bool	'True', 'False'
BeamStepSizeFracturing	bool	'True', 'False'
FractureMode	enumerator	'Conventional', 'LRFT', 'Curve
MultipassLayer	string	
UserDosePass	list	
RegionList	list	[[X1,Y1],[X2,Y2],…]
MainfieldOffsetX	float	
MainfieldOffsetY	float	
SubfieldOffsetX	float	
SubfieldOffsetY	float	
OverlapMethod	enumerator	'Standart', 'Interleaving + extra
InterleavingSize	float	
InterlockLayer	string	
MultipassMode	int	
WritingOrderSelecito	string	

n		
SequenceFile	string	
CurveTolerance	float	

11.4.9 extract

instance methods

```
GObject extract( **gobject**, { 'ExtentMode' : 'Default', 'ExtractMode'
               : 'InstancesExtract', 'CellName' : '', 'LayerSet' : '*',
               'RegionLayer' : '', 'RegionBehavior' : 'Clip',
               'AllExceptForRegions' : False, 'ExtractBoxes' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
ExtendMode	string	‘Default’,’Minimum’,’Region’,’User’
ExtractMode	string	‘InstancesExtract’, ‘CellExtract’,
CellName	string	Cell Instances - Cell Name
LayerSet	string	Layer
RegionLayer	string	
RegionBehavior		‘Clip’, ‘Within’, ‘Touching’
AllExeptForRegions	bool	‘True’, ‘False’
ExtractBoxes	list	X min [um], X max [um], Y min [um],
LowerLeftX	float	
LowerLeftY	float	
UpperRightX	float	
UpperRightY	float	

11.4.10 fda

instance methods

```
GObject fda( **gobject**, { 'LayerDoseTableFile' : '**filename**',
           'AssignmentType' : 'Assign', 'LayerDoseList' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
LayerDoseTableFile	string	*filename*

AssignmentType	Enumerator	‘Assign Value’, ‘Multiply Value’
LayerDoseList	List	[[Layer 1,Rel. Dose 1], [Layer 2, Rel.

11.4.11 filter

instance methods

```
GObject filter( **gobject**, { 'SelectionList' : [{'GlobalMode' : 'AND',
           'TargetLayer' : 'KeepLayer', 'TargetLayerName' : '1',
           'RuleList' : [{'Attribute' : 'Width', 'Mode' : 'OR',
           'Criteria1' : '>', 'Criteria2' : '<', 'Value1' : '1',
           'Value2' : '0.2'}, {'Attribute' : 'Height', 'Criteria1' :
           '>', 'Value1' : '1'}]}, {'GlobalMode' : 'AND', 'TargetLayer'
           : 'AssignLayer', 'TargetLayerName' : '1', 'RuleList' :
           [{'Attribute' : 'Area', 'Criteria1' : '>=', 'Value1' :
           '22'}]}] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
SelectionList	list	[<SelectionFilter1>,<SelectionFilter2>,.
GlobalMode	?	‘AND, ‘OR’
TargetLayer	?	‘Keep Layer’ , ‘Assign Layer’
TargetLayerName	string	Assign Layer
RuleList	list	[{Attribute:<Width, Height, Area, Heigh

11.4.12 fracture

instance methods

```
GObject fracture( **gobject**, { 'Resolution' : 0.001, 'BeamStepSize' :
                 1, 'CurveTolerance' : 1, 'FractureAxis' : 'X_AND_Y',
                 'FractureMode' : 'LRFT', 'BssFracturing' : True,
                 'FractureAngle' : 'AnyAngle', 'FractureTolerance' : 1,
                 'FractureType' : 'Hierarchical', 'FieldSizeX' : 1000,
                 'FieldSizeY' : 1000, 'SubfieldSizeX' : 0,
                 'SubfieldSizeY' : 0, 'TraversalDirection' : 'BottomUp',
                 'FieldOverlapX' : 0, 'FieldOverlapY' : 0,
                 'OverlapMethod' : 'Standard', 'InterleavingSize' : 0,
                 'InterlockLayer' : '*', 'MultipassMode' : 1,
                 'MainfieldOffsetX' : 0, 'MainfieldOffsetY' : 0,
                 'SubfieldOffsetX' : 0, 'SubfieldOffsetY' : 0,
                 'MultipassLayer' : '*', 'UserDosePass' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
`**gobject**`	GObject	
Resolution	float	[um]
BeamStepSize	int	1-10
CurveTolerance	int	
FractureAxis		‘X_AND_Y’, ‘X_ONLY’, ‘Y_ONLY’
FractureMode		‘Conventional’,’LRFT’,’Curved’
BssFracturing	bool	‘True’, ‘False’
FractureAngle		‘AnyAngle’, ‘45Degree’, ‘Rectangular’
FractureTolerance	int	1-10
FractureType		‘Hierarchical’, ‘Flat’, ‘FlatWithFields
FieldSizeX	float	[um]
FieldSizeY	float	[um]
SubFieldSizeX	float	[um]
SubFieldSizeY	float	[um]
TraversalDirection		‘BottomUp’, ‘TopDown’
FieldOverlapX	float	[um]
FieldOverlapY	float	[um]
OverlapMethod		‘Standart’, ‘Interleaving + extra field
InterleavingSize	float	[um]
InterlockLayer		
MultipassMode		‘1’, ‘2’, ‘4’, ‘Doseselective’
MainfieldOffsetX	float	
MainfieldOffsetY	float	
SubfieldOffsetX	float	
SubfieldOffsetY	float	
MultipassLayer		
UserDosePass	field	[<Dose larger> , < Number Passes>]

11.4.13 grid

instance methods

```
GObject grid( **gobject**, { 'DatabaseGrid' : 0.001,
           'SmoothingTolerance' : 0 } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
DatabaseGrid	float	[um]
SmoothingTolerance	float	[um]

11.4.14 heal

instance methods

```
GObject heal( **gobject**, { 'TargetLayer' : '1(0)', 'SoftFrame' : 0.3,
            'HierarchicalProcessing' : True, 'SelectedLayerSet' : '*',
            'LayerAssignment' : 'AllLayer', 'ProcessingMode' :
            'Healing' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
TargetLayer	int	
SoftFrame	float	[um]
HierarchicalProcessin	bool	‘True’, ‘False’
SelectedLayerSet	?	Per Layer
LayerAssignment	?	‘AllLayer’, ‘PerLayer’
ProcessingMode	enumerator	‘Healing’, ‘OverlapRemoval’

11.4.15 import

11.4.15.1 import_bmp

instance methods

```
GObject import_bmp( { 'PixelSize' : '1.000000', 'FileName' :
                  '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
PixelSize	float	
FileName	string	**filepath**

11.4.15.2 import_cel

instance methods

```
GObject import_cel( { 'Interface' : 'Read from File', 'PositionShiftX' :
                    0, 'PositionShiftY' : 0, 'FieldSize' : 600,
                    'DotNumber' : 60000, 'MaxErrorForConversion' : 0.001,
                    'FileName' : 'C:\Users\bengi\Desktop\22
                    \celfile.CEL' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
PositionShiftX	float	
PositionShiftY	float	
Interface	string	
FieldSize	float	
DotNumber	int	
MaxErrorForConversio	float	
FileName	string	**filepath**

11.4.15.3 import_cif

instance methods

```
GObject import_cif( { 'LayerSet' : '*', 'CIFUnits' : 'Standard',
                    'MaxErrorForConversion' : 0.001, 'FileName' : 'C:
                    \Users\bengi\Desktop\22\CIFfile.cif' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
LayerSet	string	
DatatypeSet	string	
CIFUnits	enumerator	‘Standard’, ‘Cadence’
MaxErrorForConversio	float	
FileName	string	**filepath**

11.4.15.4 import_cnf

instance methods

```
GObject import_cnf( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.5 import_dxf

instance methods

```
GObject import_dxf( { 'LayerSet' : '*', 'DatatypeSet' : '*', 'DXFUnits'
                    : 'um', 'DXFPolyMode' : 'ConvertToPolygon',
                    'MaxErrorForConversion' : '0.001000', 'SnappingRange'
                    : '0.000500', 'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
LayerSet	string	
DatatypeSet	string	
DXFUnits	enumerator	um , mm, mil
DXFPolyMode	enumerator	ConvertToPolygon, ConvertToSingleLine
MaxErrorForConvers	float	
SnappingRange	float	

11.4.15.6 import_fre

instance methods

```
GObject import_fre( {'FileName' : '<filepath>' })
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.7 import_gdr

instance methods

```
GObject import_gdr( { 'LayerSet' : '*', 'DatatypeSet' : '*', 'FileName'
                    : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
LayerSet	string	
DatatypeSet	string	

11.4.15.8 import_gds

instance methods

```
GObject import_gds( { 'LayerSet' : '*', 'ImportZeroWidthPaths' : False,
                    'LineWidth' : 0, 'ImportBoxes' : True, 'FileName' :
                    'C:\Users\bengi\Desktop\22\22.gds' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
LayerSet	string	
DatatypeSet	string	
ImportZeroWidthPat	bool	'True','False'
LineWidth	float	
ImportBoxes	bool	'True','False'

11.4.15.9 import_gpf

instance methods

```
GObject import_gpf( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.10 import_j01

instance methods

```
GObject import_j01( { 'Resolution' : 0.1, 'MaxErrorForConversion' : 0.1,
                    'UseInternalScale' : True, 'FileName' : 'C:\Users
                    \bengi\Desktop\22\JEOLfile.j01' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
Resolution	float	
MaxErrorForConvers	float	
UseInternalScale	bool	'True','False'

11.4.15.11 import_j51

instance methods

```
GObject import_j51( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.12 import_j52

instance methods

```
GObject import_j52( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.13 import_jb

instance methods

```
GObject import_jb( { 'LayerSet' : '*', 'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
LayerSet	string	

11.4.15.14 import_ldb

instance methods

```
GObject import_ldb( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.15 import_meb

instance methods

```
GObject import_meb( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.16 import_mic

instance methods

```
GObject import_mic( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.17 import_png

instance methods

```
GObject import_png( { 'PixelPitch' : 1, 'PixelSize' : 1,
                    'DoseValueForGreyScale' : 0.003922, 'FileName' : 'C:
                    \Users\bengi\Desktop\22\PNGFile.png' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
PixelPitch	float	
PixelSize	float	
DoseValueForGreyScal	float	

11.4.15.18 import_sdf

instance methods

```
GObject import_sdf( { 'PixelSizeNm' : 200, 'StripeWidth' : 800,
                    'MaxCircleError' : 1, 'MirrorStripeY' : False,
                    'FileName' : 'C:\Users\bengi\Desktop\22/SDF_File
                    \0.sdf' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values

FileName	string	**filepath**
PixelSizeNm	int	
StripeWidth	int	
MaxCircleError	float	
MirrorStripeY	bool	'True','False'

11.4.15.19 import_txl

instance methods

```
GObject import_txl( { 'LayerSet' : '*', 'FileName' : 'C:\Users\bengi
                    \Desktop\22\TXLfile.txl' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**
LayerSet	string	
DatatypeSet	string	

11.4.15.20 import_vep

instance methods

```
GObject import_vep( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.15.21 import_wfl

instance methods

```
GObject import_wfl( {'FileName' : '<filepath>' } )
```

All parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
FileName	string	**filepath**

11.4.16 mapping

instance methods

```
GObject mapping( **gobject**, { 'AliasTableFile' : '**filename**',
           'AliasList' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
AliasTableFile	string	**filename**
AliasList	list	[<layer>,<alias>]

11.4.17 merge

instance methods

```
GObject merge( [ **gobject1**, **gobject2**, ... ],
                   { 'DoseClassBehaviour' : 'UnlimitedNumber',
                   'NumberDoseClasses' : '256' } )
```

parameter	type	description/values
**Array of gobjects	Array(of GObject	
DoseClassBehaviour	Enumerator	UnlimitedNumber,LimitedNumber
NumberDoseClasses	integer	

11.4.18 minus

instance methods

```
minus( **left_gobject**, **right_gobject**, { 'SoftFrame' : 0.3 } )
```

The two parameters must be GObjects. This module has no other parameters.

parameter	type	description/values
left/right gobject	GObject	
SoftFrame	float	[um]

11.4.19 p_xor

instance methods

```
GObject p_xor( **gobject**, { 'TargetLayer' : '4(0)', 'SoftFrame' : 0,
             'HierarchicalProcessing' : True, 'SelectedLayerSet' : '*',
                'LayerAssignment' : 'AllLayer' } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
HierarchicalProcessin	bool	'True', 'False'
TargetLayer	int	All Layers - Target Layer
SelectedLayerSet		Per Layer - Layers
SoftFrame	float	[um]
LayerAssignment		'AllLayers', 'PerLayer'

11.4.20 pec

```
GObject pec( **gobject**, { 'UserdefinedDoseClassFile' : '**filename**',
           'MinFractureSizeMode' : 'Userdefined', 'Alpha' : 0.005,
           'Beta' : 30, 'Eta' : 0.6, 'Gamma1' : 0.3, 'Gamma2' : 2,
           'Nue1' : 0.01, 'Nue2' : 0.01, 'FoggingCorrection' : False,
           'FoggingSigma' : 20000, 'FoggingWeight' : 0.05,
           'FoggingNeighborhoodDensity' : 0, 'BlurVariabiliy' : False,
           'DoseClassMode' : 'UserdefinedDoseClasses',
           'MaxNumOfDoseClasses' : 256, 'Accuracy' : 1, 'BeamSize' :
           0.01, 'UserDefinedSeparationValue' : True, 'SeparationValue'
           : 0.1, 'FractureGrid' : 0.01, 'MinFractureSize' : 0.1,
           'MinDoseFactor' : 0.1, 'MaxDoseFactor' : 10, 'PitchX' : 0,
           'PitchY' : 0, 'RepetitionX' : 1, 'RepetitionY' : 1,
           'SmallestFeature' : 0.05, 'LayerListForCorrection' : '*',
           'PSFFileName' : '', 'UseNumericalPSF' : True,
           'MidRangeActivationThreshold' : 2, 'SingleLineBeamWidth' : 0,
           'AutoSRRegionDetection' : True, 'PeriodicLayout' : False,
           'ManualResolution' : True, 'IncludeSRCorrection' : True,
           'HierarchicShortRangePEC' : True, 'ShortRangePECUseGPU' :
           False, 'Med1Correction' : False, 'Med2Correction' : False,
           'NewSRAlgorithm' : False, 'EtchEffects' : False,
           'EtchInfluenceRange' : 1, 'EtchVisibleDirectionInside' :
           True, 'EtchEffects2' : False, 'EtchInfluenceRange2' : 1 } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
UserdefinedDoseClassFile	string	**filename**
MinFractureSizeMode		'Automatic', 'User defined'
Alpha	float	[um]
Beta	float	[um]

Eta	float	
Gamma1	float	[um]
Gamma2	float	[um]
Nue1	float	
Nue2	float	
FoggingCorrection	bool	‘True’, ‘False’
FoggingSigma	int	
FoggingWeight	float	
FoggingNaighborhoodDensity	int	
BlurVariabiliy	bool	‘True’, ‘False’
DoseClassMode	enumerator	‘Accuracy’, ‘FixedClassNumber’, ‘
MaxNumOfDoseClasses	int	
Accuracy	float	
BeamSize	float	
UserDefinedSeparationValue	bool	‘True’, ‘False’
SeperationValue	float	
SeperationValue	float	
MinDoseFactor	float	
MaxDoseFactor	float	
MidRangeActivationThreshol	int	
SingleLineBeamWidth	int	
NumOfDoseClasses	int	
FractureGrid	float	
MinFractureSize	float	
MinFigureSize	float	
MaxDoseFactor	float	
PitchX	float	
PitchY	float	
RepetitionX	int	
RepetitionY	int	
SmallestFeature	float	
LayerListForCorrection	string	Layers for SR correction
PSFFileName	string	**filename**
UseNumericalPSF	bool	‘True’, ‘False’
AutomaticResolution	bool	‘True’, ‘False’
AutoSRRegionDetection	bool	‘True’, ‘False’
PeriodicLayout	bool	‘True’, ‘False’
ManualResolution	bool	‘True’, ‘False’

FastApproximation	bool	‘True’, ‘False’
IncludeSRCorrection	bool	‘True’, ‘False’
Med1Correction	bool	‘True’, ‘False’
Med2Correction	bool	‘True’, ‘False’
NewSRAlgorithm	bool	‘True’, ‘False’
HierarchicShortRangePEC	bool	‘True’, ‘False’
ShortRangePECUseGPU	bool	‘True’, ‘False’
EtchEffects	bool	‘True’, ‘False’
EtchInfluenceRange	int	
EtchVisibleDirectionInside	bool	‘True’, ‘False’
EtchEffects2	bool	‘True’, ‘False’
EtchInfluenceRange2	bool	‘True’, ‘False’

11.4.21 pec_3d

instance methods

```
GObject pec_3d( **gobject**, { ‘LayerDoseTableFile’ : ‘**filename**’,
                 ‘ContrastCurveFile’ : ‘**filename**’,
                 ‘UserdefinedDoseClassFile’ : ‘**filename**’,
                 ‘MinFractureSizeMode’ : ‘Userdefined’, ‘Alpha’ : 0.005,
                 ‘Beta’ : 30, ‘Eta’ : 0.6, ‘Gamma1’ : 0.3, ‘Gamma2’ : 2,
                 ‘Nue1’ : 0.01, ‘Nue2’ : 0.01, ‘FoggingCorrection’ :
                 True, ‘FoggingSigma’ : 20000, ‘FoggingWeight’ : 0.05,
                 ‘FoggingNeighborhoodDensity’ : 0, ‘BlurVariabiliy’ :
                 True, ‘DoseClassMode’ : ‘Accuracy’,
                 ‘MaxNumOfDoseClasses’ : 256, ‘Accuracy’ : 1, ‘BeamSize’
                 : 0.01, ‘UserDefinedSeparationValue’ : True,
                 ‘SeparationValue’ : 0.1, ‘FractureGrid’ : 0.01,
                 ‘MinFractureSize’ : 0.1, ‘MinDoseFactor’ : 0.1,
                 ‘MaxDoseFactor’ : 10, ‘PitchX’ : 0, ‘PitchY’ : 0,
                 ‘RepetitionX’ : 1, ‘RepetitionY’ : 1, ‘SmallestFeature’
                 : 0.05, ‘LayerListForCorrection’ : ‘*’,
                 ‘WorkingRangeMin’ : 0, ‘WorkingRangeMax’ : 1,
                 ‘SurfacePECDefinesThickness’ : True, ‘IsSurfacePEC’ :
                 True, ‘BaseDose’ : 300, ‘OriginalResistThickness’ : 1,
                 ‘PSFFileName’ : ‘’, ‘UseNumericalPSF’ : False,
                 ‘MidRangeActivationThreshold’ : 2,
                 ‘SingleLineBeamWidth’ : 0, ‘AutoSRRegionDetection’ :
                 True, ‘PeriodicLayout’ : True, ‘ManualResolution’ :
                 True, ‘IncludeSRCorrection’ : True,
                 ‘HierarchicShortRangePEC’ : False,
                 ‘ShortRangePECUseGPU’ : False, ‘Med1Correction’ : True,
                 ‘Med2Correction’ : True, ‘NewSRAlgorithm’ : False,
                 ‘EtchEffects’ : False, ‘EtchInfluenceRange’ : 1,
                 ‘EtchVisibleDirectionInside’ : True, ‘EtchEffects2’ :
                 False, ‘EtchInfluenceRange2’ : 1 } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

gobject	Gobject	
LayerDoseTableFile	string	**filename**
UserdefinedDoseClassFile	string	**filename**
ContrastCurveFile	string	**filename**
MinFractureSizeMode		‘Automatic’, ‘User defined’
Alpha	float	[um]
Beta	float	[um]
Eta	float	
Gamma1	float	[um]
Gamma2	float	[um]
Nue1	float	
Nue2	float	
FoggingCorrection	bool	‘True’, ‘False’
FoggingSigma	int	
FoggingWeight	float	
FoggingNaighborhoodDensity	int	
BlurVariabiliy	bool	‘True’, ‘False’
DoseClassMode	enumerator	‘Accuracy’, ‘FixedClassNumber’, ‘
MaxNumOfDoseClasses	int	
Accuracy	float	
BeamSize	float	
UserDefinedSeparationValue	bool	‘True’, ‘False’
SeperationValue	float	
SeperationValue	float	
MinDoseFactor	float	
MaxDoseFactor	float	
MidRangeActivationThreshol	int	
SingleLineBeamWidth	int	
NumOfDoseClasses	int	
FractureGrid	float	
MinFractureSize	float	
MinFigureSize	float	
MaxDoseFactor	float	
PitchX	float	
PitchY	float	
RepetitionX	int	
RepetitionY	int	

SmallestFeature	float	
LayerListForCorrection	string	Layers for SR correction
PSFFileName	string	**filename**
UseNumericalPSF	bool	‘True’, ‘False’
AutomaticResolution	bool	‘True’, ‘False’
AutoSRRegionDetection	bool	‘True’, ‘False’
PeriodicLayout	bool	‘True’, ‘False’
ManualResolution	bool	‘True’, ‘False’
FastApproximation	bool	‘True’, ‘False’
IncludeSRCorrection	bool	‘True’, ‘False’
Med1Correction	bool	‘True’, ‘False’
Med2Correction	bool	‘True’, ‘False’
NewSRAlgorithm	bool	‘True’, ‘False’
HierarchicShortRangePEC	bool	‘True’, ‘False’
ShortRangePECUseGPU	bool	‘True’, ‘False’
EtchEffects	bool	‘True’, ‘False’
EtchInfluenceRange	int	
EtchVisibleDirectionInside	bool	‘True’, ‘False’
EtchEffects2	bool	‘True’, ‘False’
EtchInfluenceRange2	bool	‘True’, ‘False’

11.4.22 pec_shape

instance methods

```
GObject pec_shape( **gobject**, { 'UserdefinedDoseClassFile' :
           '**filename**', 'MinFractureSizeMode' : 'Userdefined',
           'Alpha' : 0.005, 'Beta' : 30, 'Eta' : 0.6, 'Gamma1' : 0.3,
           'Gamma2' : 2, 'Nue1' : 0.01, 'Nue2' : 0.01,
           'FoggingCorrection' : False, 'FoggingSigma' : 20000,
           'FoggingWeight' : 0.05, 'FoggingNeighborhoodDensity' : 0,
           'BlurVariabiliy' : True, 'DoseClassMode' : 'Accuracy',
           'MaxNumOfDoseClasses' : 256, 'Accuracy' : 1, 'BeamSize' :
           0.01, 'UserDefinedSeparationValue' : True, 'SeparationValue'
           : 0.1, 'FractureGrid' : 0.01, 'MinFractureSize' : 0.1,
           'MinDoseFactor' : 0.1, 'MaxDoseFactor' : 10, 'PitchX' : 0,
           'PitchY' : 0, 'RepetitionX' : 1, 'RepetitionY' : 1,
           'SmallestFeature' : 0.05, 'LayerListForCorrection' : '*',
           'ShiftStepSize' : 0.001, 'MaxShift' : 0.1, 'MinSegmentSize' :
           0.5, 'TargetDoseOfShapePec' : 0.5, 'OverdoseFactor' : 1,
           'MachineEta' : 0.6, 'PSFFileName' : '', 'UseNumericalPSF' :
           False, 'MidRangeActivationThreshold' : 2,
           'SingleLineBeamWidth' : 0, 'AutoSRRegionDetection' : True,
           'PeriodicLayout' : False, 'ManualResolution' : False,
           'IncludeSRCorrection' : True, 'HierarchicShortRangePEC' :
```

```
False, 'ShortRangePECUseGPU' : False, 'Med1Correction' :
True, 'Med2Correction' : True, 'NewSRAlgorithm' : True,
'EtchEffects' : True, 'EtchInfluenceRange' : 1,
'EtchVisibleDirectionInside' : True, 'EtchEffects2' : True,
'EtchInfluenceRange2' : 1, 'EtchRateList' : [[0.000000,
1.000000], [1.000000, 1.000000]], 'EtchRateList2' :
[[0.000000, 1.000000], [1.000000, 1.000000]], 'EtchRateList3'
: [[0.000000, 1.000000], [1.000000, 1.000000]] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
UserdefinedDoseClassFile	string	**filename**
MinFractureSizeMode		‘Automatic’, ‘User defined’
Alpha	float	[um]
Beta	float	[um]
Eta	float	
Gamma1	float	[um]
Gamma2	float	[um]
Nue1	float	
Nue2	float	
FoggingCorrection	bool	‘True’, ‘False’
FoggingSigma	int	
FoggingWeight	float	
FoggingNaighborhoodDensity	int	
BlurVariabiliy	bool	‘True’, ‘False’
DoseClassMode	enumerator	‘Accuracy’, ‘FixedClassNumber’, ‘
MaxNumOfDoseClasses	int	
Accuracy	float	
BeamSize	float	
UserDefinedSeparationValue	bool	‘True’, ‘False’
SeperationValue	float	
SeperationValue	float	
MinDoseFactor	float	
MaxDoseFactor	float	
MidRangeActivationThreshol	int	
SingleLineBeamWidth	int	
NumOfDoseClasses	int	
FractureGrid	float	

MinFractureSize	float	
MinFigureSize	float	
MaxDoseFactor	float	
PitchX	float	
PitchY	float	
RepetitionX	int	
RepetitionY	int	
SmallestFeature	float	
LayerListForCorrection	string	Layers for SR correction
ShiftStepSize	float	
MaxShift	float	
MinSegmentSize	float	
TargetDoseOfShapePec	float	
OverdoseFactor	float	
MachineEta	float	
PSFFileName	string	**filename**
UseNumericalPSF	bool	'True', 'False'
AutomaticResolution	bool	'True', 'False'
AutoSRRegionDetection	bool	'True', 'False'
PeriodicLayout	bool	'True', 'False'
ManualResolution	bool	'True', 'False'
FastApproximation	bool	'True', 'False'
IncludeSRCorrection	bool	'True', 'False'
Med1Correction	bool	'True', 'False'
Med2Correction	bool	'True', 'False'
NewSRAlgorithm	bool	'True', 'False'
HierarchicShortRangePEC	bool	'True', 'False'
ShortRangePECUseGPU	bool	'True', 'False'
EtchEffects	bool	'True', 'False'
EtchInfluenceRange	int	
EtchVisibleDirectionInside	bool	'True', 'False'
EtchEffects2	bool	'True', 'False'
EtchInfluenceRange2	bool	'True', 'False'
EtchRateList	list	[[Etch Density 1, Etch Bias 1], [
EtchRateList2	list	[[Normalized Contrast 1, Etch Bia
EtchRateList3	list	[[Curvature 1, Etch Bias 1], [Cur

11.4.23 size

instance methods

```
GObject size( **gobject**, { 'Bias' : 0, 'CornerExtension' : 1,
                    'TargetLayer' : '0(0)', 'Mode' : 'X-Y',
                    'LayerAssignment' : 'AllLayer', 'SelectedLayerSet' :
                    '*', 'BiasLayerList' : [] } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
Bias	float	[um]
CornerExtension	float	
TargetLayer		All Layer - Target Layer
Mode		'X-Y', 'X', 'Y'
LayerAssignment		'AllLayer', 'PerLayer', 'LayerSpecific'
SelectedLayerSet		Per Layer - Layers
BiasLayerList	list	Layer Specific

11.4.24 transform

instance methods

```
GObject transform( **gobject**, { 'CoordinateOrigin' : 'LayoutOrigin',
                    'ScaleX' : 1, 'ScaleY' : 1, 'ShiftX' : 0, 'ShiftY' : 0,
                    'Rotation' : 0, 'ReflectX' : False, 'ReflectY' :
                    False } )
```

The first parameter must be a GObject. All other parameters are optional, named parameters and can occur in an arbitrary order.

parameter	type	description/values
gobject	GObject	
CoordinateOrigin	enumerator	'LayoutOrigin', 'Center', 'BottomLeft'
ScaleX	float	
ScaleY	float	
ShiftX	float	
ShiftY	float	
Rotation	float	
ReflectX	bool	'True', 'False'
ReflectY	bool	'True', 'False'

12 TextLib TXL

TXL is a ASCII based layout format supported by BEAMER. It provides an easy path to generate layout or to manipulate existing layouts by additions of elements.

TXL is using layout elements called features like paths, circles, boxes. These features are modified by attributes like width, layer, angles or magnification.

12.1 Header and Footer

The TXL format needs a header and footer that wraps around the structure definitions. An example TXL file is shown below. In bold at the top of the code in green is the header. In bold and red at the bottom of the code is the footer. The body of the code (in plain black text) is the pattern definition using two different structure commands that will be defined in a later section.

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT circle
     LAYER 0
     DATATYPE 0
     C 0.050 0.00,0.00 (0.000 360.000 360) ENDC
     ENDSTRUCT

     STRUCT photonic_crystal_array
     AREF circle (0,0) 1000 (0.150,0.000) 1000 (0.000,0.150)
     ENDSTRUCT

ENDLIB
```

The first line, `TEXTLIB 9.0.0`, is the identification of the file.

The `UNIT` line denotes which units the positional values snapped. Allowed arguments for the `UNIT` command are `NM`, `UM`, `MICRON`, `MM`, `METER`, `MIL`, `INCH`.

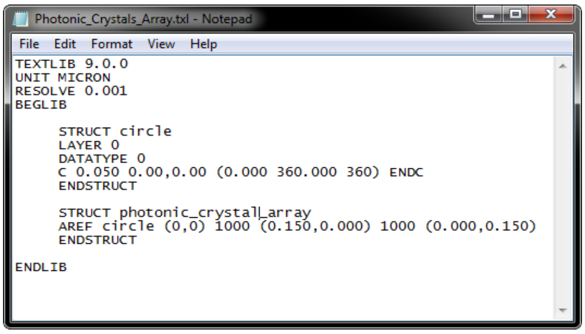
The `RESOLVE` is the design grid of the layout. Features need to be on a multiple of the `RESOLVE` to have a proper design.

`BEGLIB` denotes the starting of the lists of features that are inside the layout.

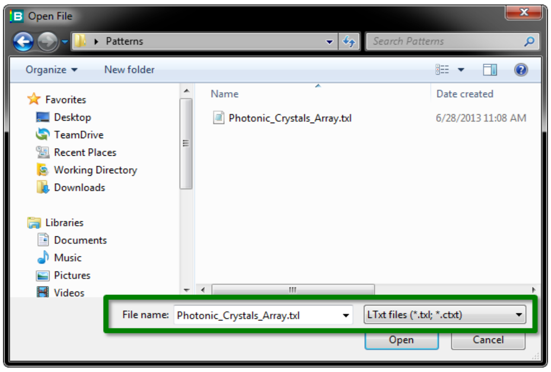
When defining a feature the `STRUCT` command is used. `STRUCT` <name> deciphers the cell that is now being designed `ENDSTRUCT` closes the cell. `STRUCT`s will be discussed later.

The footer of a TXL file is the `ENDLIB`. The parser knows with this command the end of the layout has been reached.

If the sample code above is saved to a text file like below left using Notepad in Windows, the data can imported right away into BEAMER via the Import module (see below right).


Photonic_Crystals_Array.txl - Notepad
File Edit Format View Help

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT circle
     LAYER 0
     DATATYPE 0
     C 0.050 0.00,0.00 (0.000 360.000 360) ENDC
     ENDSTRUCT

     STRUCT photonic_crystal_array
     AREF circle (0,0) 1000 (0.150,0.000) 1000 (0.000,0.150)
     ENDSTRUCT

ENDLIB
```

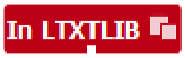


Photonic_Crystals_Array.txl 6/28/2013 11:08 AM

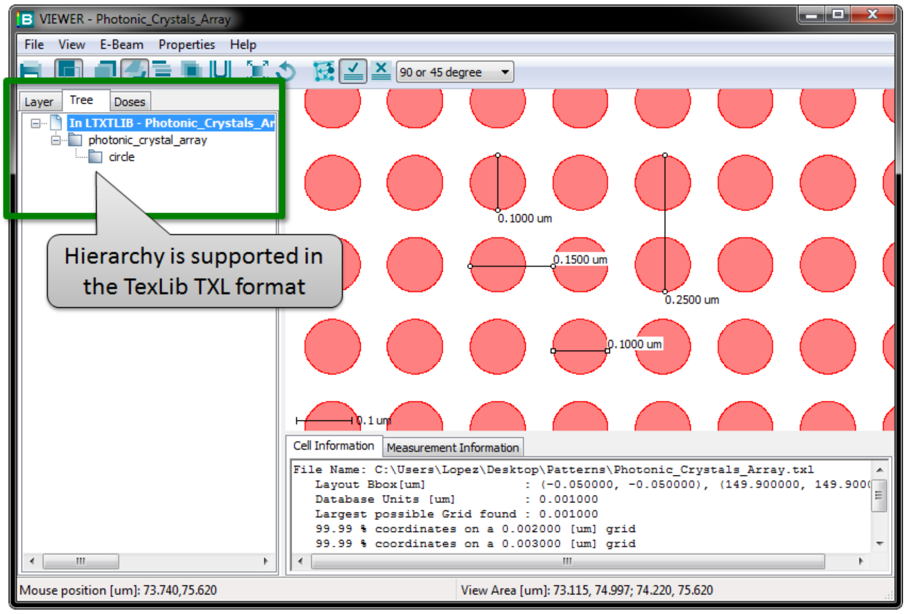
File name: Photonic_Crystals_Array.txl

LTxt files (*.txl; *.ctxt)

With the pattern imported, the Import module should look like:



Left click the icon on the module and the pattern will be loaded in the VIEWER:



Hierarchy is supported in the TexLib TXL format

```
File Name: C:\Users\Lopez\Desktop\Patterns\Photonic_Crystals_Array.txl
   Layout Bbox[um]              : (-0.050000, -0.050000), (149.900000, 149.900(
   Database Units [um]          : 0.001000
   Largest possible Grid found : 0.001000
   99.99 % coordinates on a 0.002000 [um] grid
   99.99 % coordinates on a 0.003000 [um] grid
```

Mouse position [um]: 73.740,75.620

View Area [um]: 73.115, 74.997; 74.220, 75.620

In the sample pattern, an array of circles are generated from a single cell called **circle**. The array serves as the topmost part of the hierarchy called **photonic_crystal_array**, that defines the pitch between the arrayed cell references.

12.2 STRUCT

When defining a pattern the `STRUCT` command is used. A `STRUCT` is simply a cell within a pattern. `STRUCT` <name> deciphers the cell that is now being designed `ENDSTRUCT` closes the cell. In the previous section, the code segment from the body of the code,

```
STRUCT circle
LAYER 0
DATATYPE 0
C 0.050 0.00,0.00 (0.000 360.000 360) ENDC
ENDSTRUCT
```

gave a good example of a simple `STRUCT`. Above, the name of the cell is **circle** (hence the line

STRUCT circle) on layer 0 with datatype 0, which in BEAMER will be seen as 0(0). The lines:

```
LAYER 0
DATATYPE 0
```

are attributes. These attributes are defined in a later section. Here the layer and datatype are both defined explicitly as 0, which in BEAMER is 0(0).

The circle is defined by the line:

```
C 0.050 0.00,0.00 (0.000 360.000 360) ENDC
```

This command is the circle command, which is defined in a later section. As stated, ENDSTRUCT closes the cell definition.

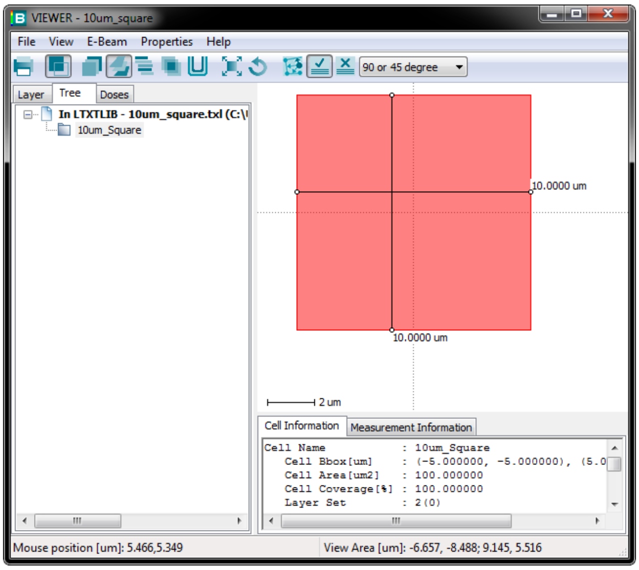
12.2.1 BOUNDARY

Rectangles, trapezoids and polygons can be defined by the boundary command. The boundary structure starts of with a B (for boundary) and ends with an ENDB. Between these a list of x,y coordinates describe the outline of the boundary. The boundary is always closed so the last point connects to the starting point. Here the name of the cell is called **10um_Square**.

A sample:

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT 10um_Square
     LAYER 2
     DATATYPE 0
     B -5.000,-5.000 5.000,-5.000 5.000,5.000 -5.000,5.000 ENDB
     ENDSTRUCT

ENDLIB
```

The pattern in the VIEWER looks as follows:


VIEWER - 10um_square
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
In LTXTLIB - 10um_square.txl (C:\
10um_Square
10.0000 um
10.0000 um
2 um
Cell Information Measurement Information

```
Cell Name              : 10um_Square
   Cell Bbox[um]       : (-5.000000, -5.000000), (5.0
   Cell Area[um2]      : 100.000000
   Cell Coverage[%]    : 100.000000
   Layer Set           : 2(0)
```

Mouse position [um]: 5.466,5.349 View Area [um]: -6.657, -8.488; 9.145, 5.516

Possible Attributes are LAYER, DATATYPE, ANGLE, MAG.

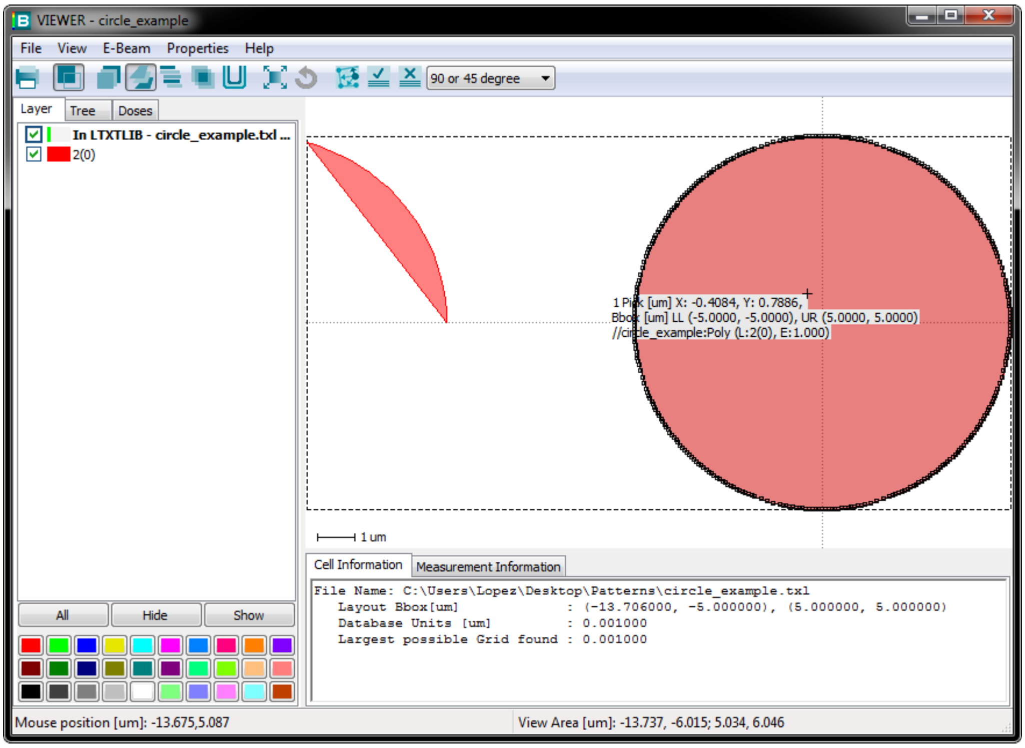
12.2.2 CIRCLE

The CIRCLE structure starts with an C and ends with an ENDC. The circle is then defined by a radius, the center point and optionally by start and end angles with the discrete number of points describing the circle. The attributes will put this cell (named **circle_example**) on layer/datatype 2(0).

A sample:

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT circle_example
     LAYER 2
     DATATYPE 0
     C 5.000 -15.000,0.000 (0.000 75.000 100) ENDC
     C 5.000   0.000,0.000 (0.000 360.000 360) ENDC
     ENDSTRUCT

ENDLIB
```

When importing this pattern into BEAMER, the resulting pattern is as follows:



```
1 Pick [um] X: -0.4084, Y: 0.7886,
Bbox [um] LL (-5.0000, -5.0000), UR (5.0000, 5.0000)
//circle_example:Poly (L:2(0), E:1.000)
```

```
File Name: C:\Users\Lopez\Desktop\Patterns\circle_example.txl
   Layout Bbox[um]               : (-13.706000, -5.000000), (5.000000, 5.000000)
   Database Units [um]           : 0.001000
   Largest possible Grid found : 0.001000
```

Mouse position [um]: -13.675,5.087

View Area [um]: -13.737, -6.015; 5.034, 6.046

In this pattern, an polygonal arc is generated from 0 to 75 degrees and is drawn with 100 vertices. The center of the pattern is offset by -15um to the left. This is defined by line:

```
C 5.000 -15.000,0.000 (0.000 75.000 100) ENDC
```

The full circle is defined by:

```
C 5.000   0.000,0.000 (0.000 360.000 360) ENDC
```

and is placed on the origin, defined by 360 vertices. Additional attributes that can be used with this command are LAYER, DATATYPE, ANGLE, MAG.

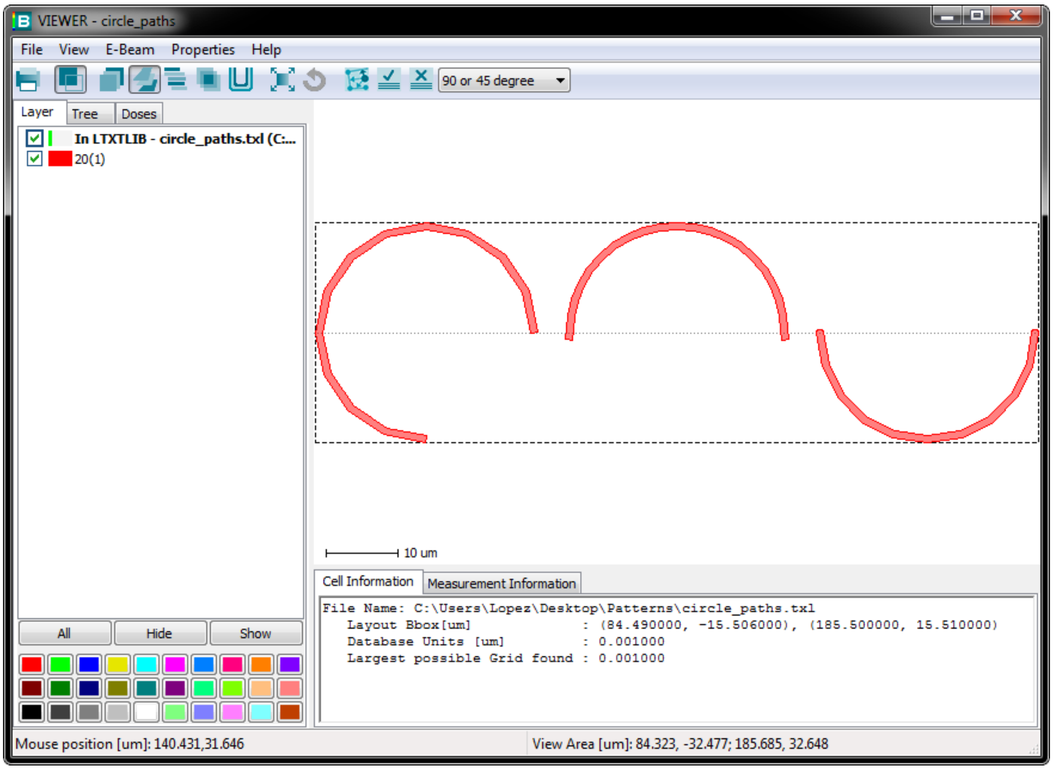
12.2.3 CIRCLEPATH

The CIRCLEPATH structure starts with an CP and ends with an ENDCP. It defines a circular path with a parameter set of radius, center point, start and end angles and number of segments in 360 degrees. The special commands are CPE for extended path, where the path is stretched beyond the defined endpoints and the CPR where a round cap is added to the path.

A sample:

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT circle_paths
     LAYER 20
     DATATYPE 1
     WIDTH 1
     CP  15 100,0   0 270    ENDCP
     CPE 15 135,0   0 180 40 ENDCP
     CPR 15 170,0 180 360 20 ENDCP
     ENDSTRUCT

ENDLIB
```

The code produces the following shapes on layer/datatype 20(1):



```
File Name: C:\Users\Lopez\Desktop\Patterns\circle_paths.txl
   Layout Bbox[um]             : (84.490000, -15.506000), (185.500000, 15.510000)
   Database Units [um]         : 0.001000
   Largest possible Grid found : 0.001000
```

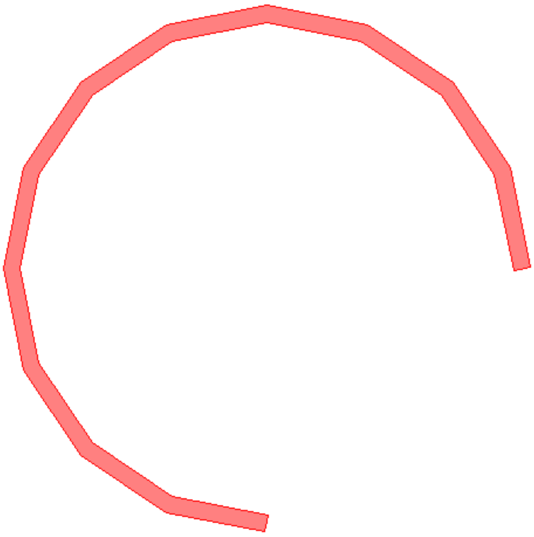
Mouse position [um]: 140.431,31.646

View Area [um]: 84.323, -32.477; 185.685, 32.648

The line:

```
     CP   15 100,0    0 270     ENDCP
```

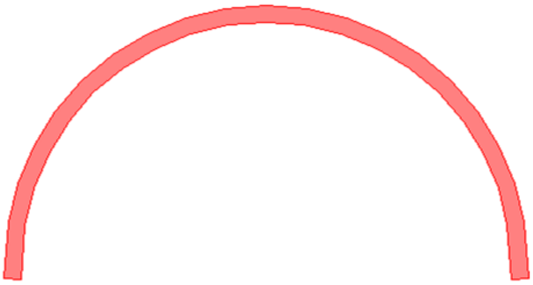
produces the pattern on the far left in the VIEWER above and to the right. Here the number of segments is not indicated, so by default, the number of segments used for a full circular path is 16.



The line:

```
     CPE 15 135,0    0 180 40 ENDCP
```

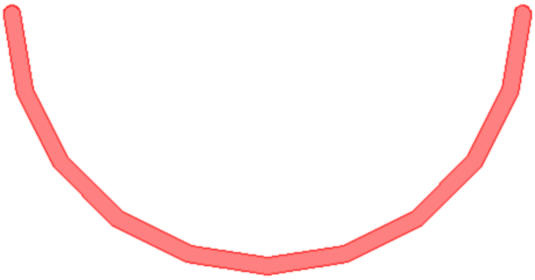
produces the pattern on the middle in the VIEWER above and to the right. Here the number of segment is 40 segments for a full circle. With the `CPE` command, the end points are extended beyond the defined endpoints.



The line:

```
     CPR 15 170,0 180 360 20 ENDCP
```

produces the pattern on the far right in the VIEWER above and to the right. Here the number of segment is 20 segments for a full circle. With the `CPR` command, the end points are extended beyond the defined endpoints.



Additional attributes that can be used with this command are LAYER, DATATYPE, ANGLE, MAG.

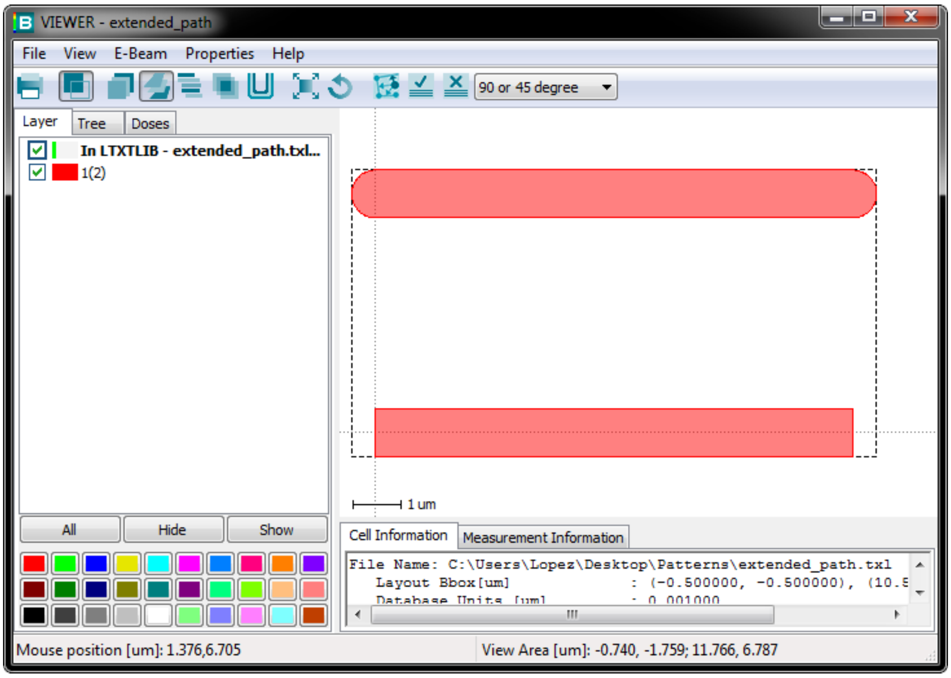
12.2.4 PATH

The PATH structure starts with a P and ends with an ENDP. A list of points describe the path the PATH is walking along. An extended option of the PATH command is the PR - ENDP where the path will have rounded end caps.

A sample:

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT extended_path
     LAYER 1
     DATATYPE 2
     WIDTH 1.000
     P  0.000,0.000 10.000,0.000 ENDP
     PR 0.000,5.000 10.000,5.000 ENDP
     ENDSTRUCT

ENDLIB
```

The resulting pattern looks like the following in BEAMER:



```
File Name: C:\Users\Lopez\Desktop\Patterns\extended_path.txl
   Layout Bbox[um]                   : (-0.500000, -0.500000), (10.5
   Database Units [um]               : 0.001000
```

Mouse position [um]: 1.376,6.705

View Area [um]: -0.740, -1.759; 11.766, 6.787

Additional attributes that can be used with this command are LAYER, DATATYPE, ANGLE, MAG, WIDTH.

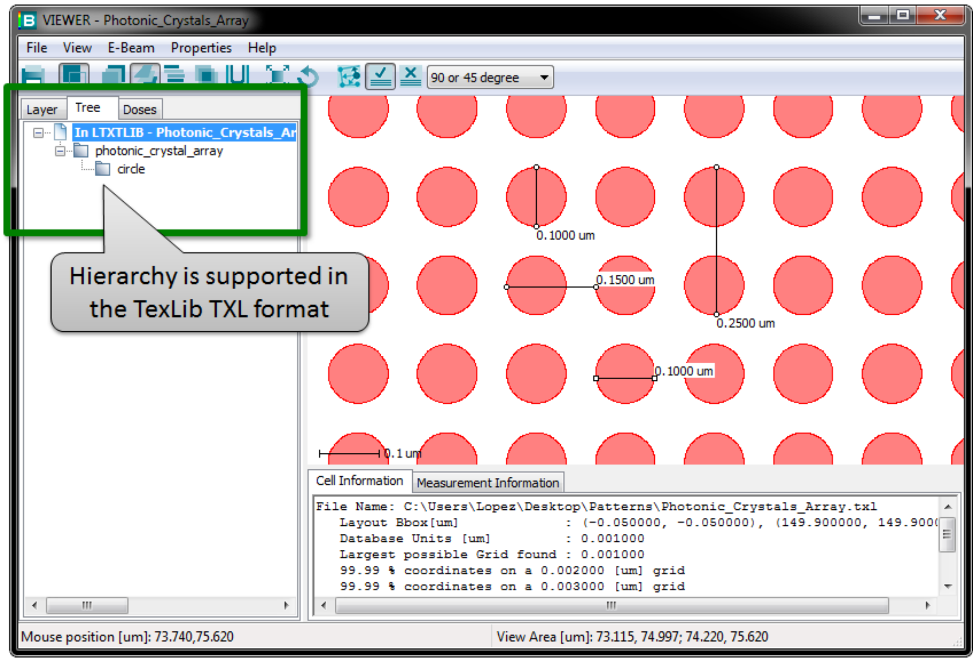
12.2.5 AREF

The AREF is a element inside a STRUCT calling other STRUCTs and position them and define their array properties. It first gives the position of the lower left as point, followed by repetitions in x and y

This sample is the same code from the section on the Header and Footer syntax:

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT circle
     LAYER 0
     DATATYPE 0
     C 0.050 0.00,0.00 (0.000 360.000 360) ENDC
     ENDSTRUCT

     STRUCT photonic_crystal_array
     AREF circle (0,0) 1000 (0.150,0.000) 1000 (0.000,0.150)
     ENDSTRUCT

ENDLIB
```

Here a cell called **circle** creates a 100nm diameter circle. Then `AREF` command in the cell named **photonic_crystal_array** arrays this cell to create a 1000 by 1000 photonic crystal array of the cell **circle**:



Hierarchy is supported in the TexLib TXL format

```
File Name: C:\Users\Lopez\Desktop\Patterns\Photonic_Crystals_Array.txl
   Layout Bbox[um]            : (-0.050000, -0.050000), (149.900000, 149.900(
   Database Units [um]        : 0.001000
   Largest possible Grid found : 0.001000
   99.99 % coordinates on a 0.002000 [um] grid
   99.99 % coordinates on a 0.003000 [um] grid
```

Mouse position [um]: 73.740,75.620

View Area [um]: 73.115, 74.997; 74.220, 75.620

Additional attributes that can be used with this command are LAYER, DATATYPE, ANGLE, MAG.

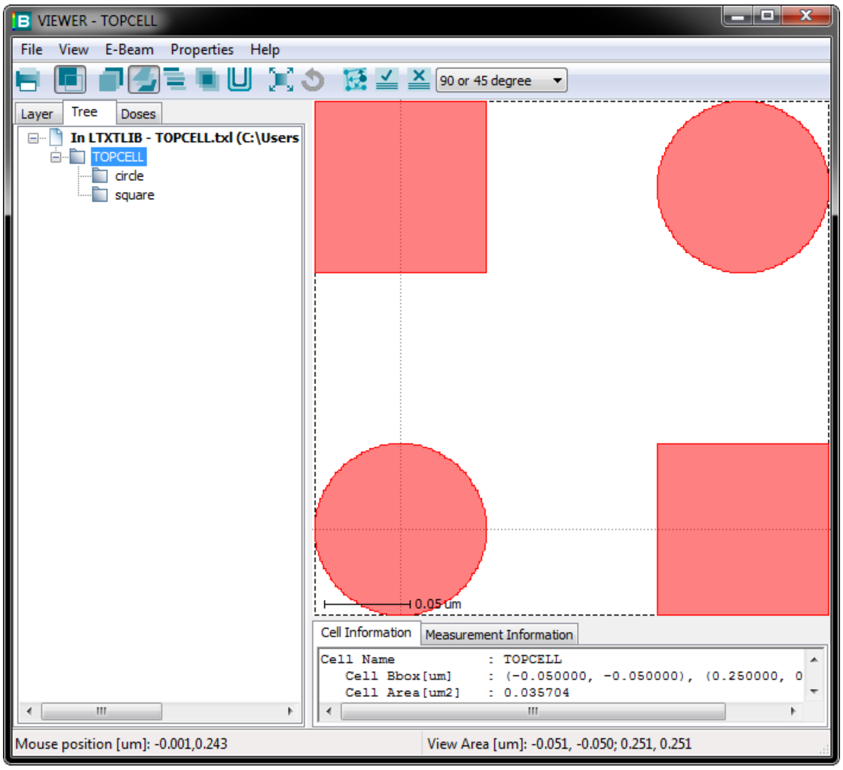
12.2.6 SREF

The SREF is an element inside a STRUCT to call other STRUCTs. It is simply calling the name of a defined cell and its position.

A sample:

```
TEXTLIB 9.0.0
UNIT MICRON
RESOLVE 0.001
BEGLIB

     STRUCT circle
     LAYER 0
     DATATYPE 0
     C 0.050 0.00,0.00 (0.000 360.000 360) ENDC
     ENDSTRUCT

     STRUCT square
     LAYER 0
     DATATYPE 0
     B -0.050,-0.050 0.050,-0.050 0.050,0.050 -0.050,0.050 ENDB
     ENDSTRUCT

     STRUCT TOPCELL
     SREF circle 0.2 0.2
     SREF circle   0   0
     SREF square 0.2   0
     SREF square   0 0.2
     ENDSTRUCT


ENDLIB
```

In BEAMER, the following pattern is produced. Here simple references to the defined cells circle and square are pulled in multiple times to generate the pattern below. Hierarchy is maintained and can be viewed in the Tree tab of the VIEWER.


VIEWER - TOPCELL
File View E-Beam Properties Help
90 or 45 degree
Layer Tree Doses
In LTXTLIB - TOPCELL.txt (C:\Users
TOPCELL
circle
square
0.05 um
Cell Information Measurement Information

```
Cell Name          : TOPCELL
   Cell Bbox[um]   : (-0.050000, -0.050000), (0.250000, 0
   Cell Area[um2]  : 0.035704
```

Mouse position [um]: -0.001,0.243

View Area [um]: -0.051, -0.050; 0.251, 0.251

Additional attributes that can be used with this command are LAYER, DATATYPE, ANGLE, MAG.

12.3 Attributes

12.3.1 ANGLE

The ANGLE defined the rotation of the features that follow the attribute.

A sample:

```
ANGLE 45
```

12.3.2 MAG

The MAG defined the magnification of the features that follow this attribute.

A sample:

```
MAG 1.5
```

12.3.3 LAYER

The LAYER attribute puts the following features in a specified layer of the design.

A sample:

```
LAYER 100
```

12.3.4 DATATYPE

The DATATYPE attribute puts the following features in a specified datatype of the design.

A sample:

```
DATATYPE 100
```

12.3.5 WIDTH

The WIDTH attribute assigns paths a specific width.

A sample:

```
WIDTH 1.00
```

13 Technical Support

13.1 License

GenISys uses the WIBU license software solution CodeMeter. The license is tied to a physical dongle with advanced security systems inside. To program the license a so called RaC file is used. These files are actually an encrypted image of the CodeMeter license sticks. The licenses provided by GenISys are always concurrent by default, meaning you can initialize a license server and access from any computer to that license server and run the software products provided by GenISys.

The following sections describe how to:
- Prepare and execute a license update
- How to setup a CodeMeter license server/client

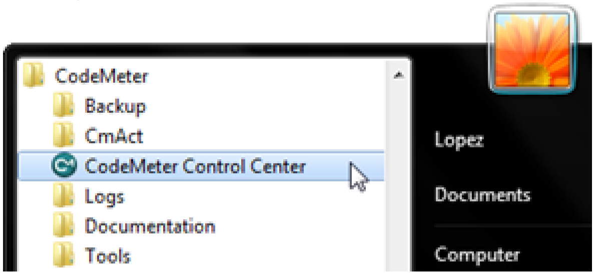
13.1.1 License Update

All GenISys products require a license update to keep your product running to the current release. As part of this process the RaC file is needed from the license stick. It must be sent to GenISys. Once processed, the RaU is returned to update the license server.

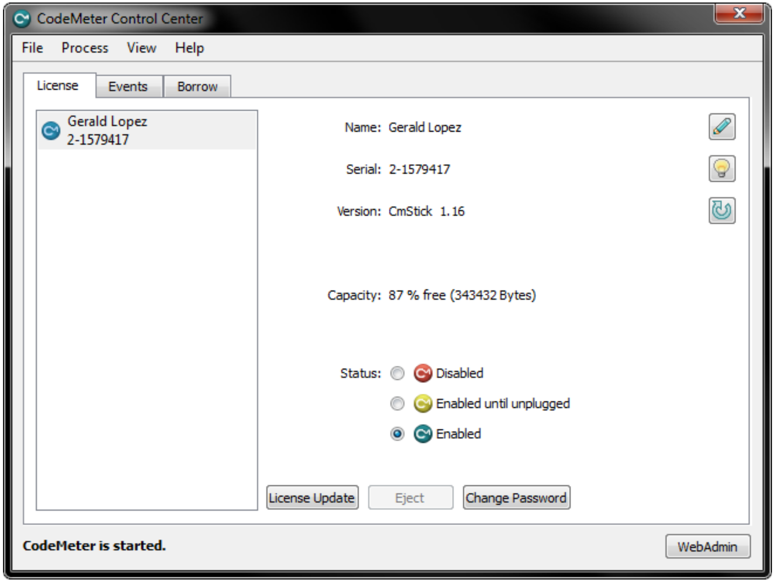
This section provides simple instructions on:
- Creating the RaC File
- Updating the CodeMeter License Server

13.1.1.1 Creating the RaC File

1. Start the Codemeter Control Center:
   a. In Linux, run the command CodeMeterCC
   b. In Windows, go to All Programs and then CodeMeter



2. Click the **License Update** button:



Name: Gerald Lopez

Serial: 2-1579417

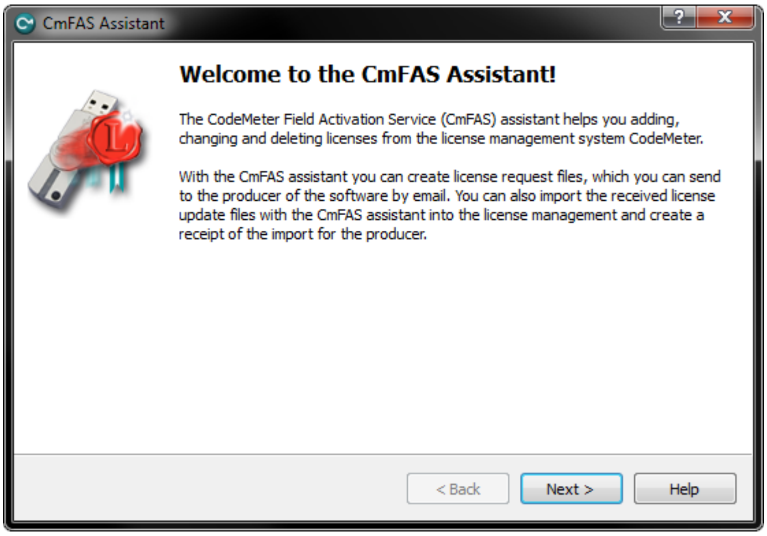
Version: CmStick 1.16

Capacity: 87 % free (343432 Bytes)

Status: ○ Disabled
○ Enabled until unplugged
● Enabled

CodeMeter is started.

3. Click **Next** to start the Field Activation Service:



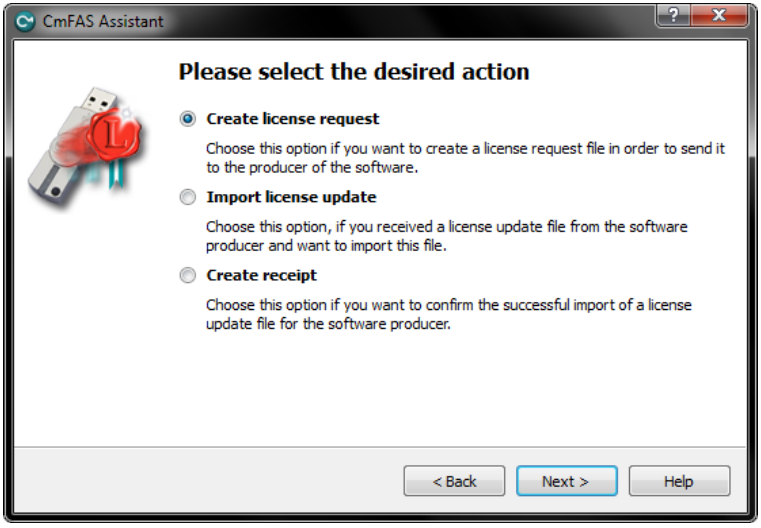
CmfAS Assistant

Welcome to the CmFAS Assistant!

The CodeMeter Field Activation Service (CmFAS) assistant helps you adding, changing and deleting licenses from the license management system CodeMeter.

With the CmFAS assistant you can create license request files, which you can send to the producer of the software by email. You can also import the received license update files with the CmFAS assistant into the license management and create a receipt of the import for the producer.

4. Select **Create license request** and click **Next**:



CmFAS Assistant

Please select the desired action

● Create license request

Choose this option if you want to create a license request file in order to send it to the producer of the software.

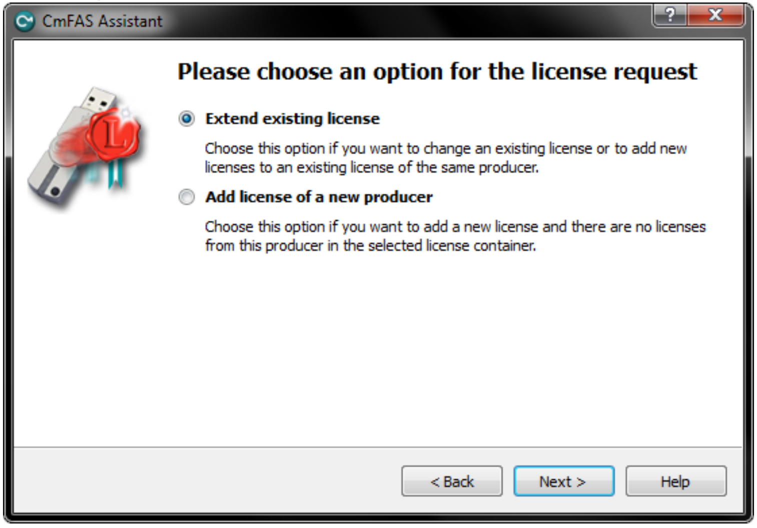
○ Import license update

Choose this option, if you received a license update file from the software producer and want to import this file.

○ Create receipt

Choose this option if you want to confirm the successful import of a license update file for the software producer.

5. Select **Extend existing license** and click **Next**:



CmFAS Assistant

Please choose an option for the license request

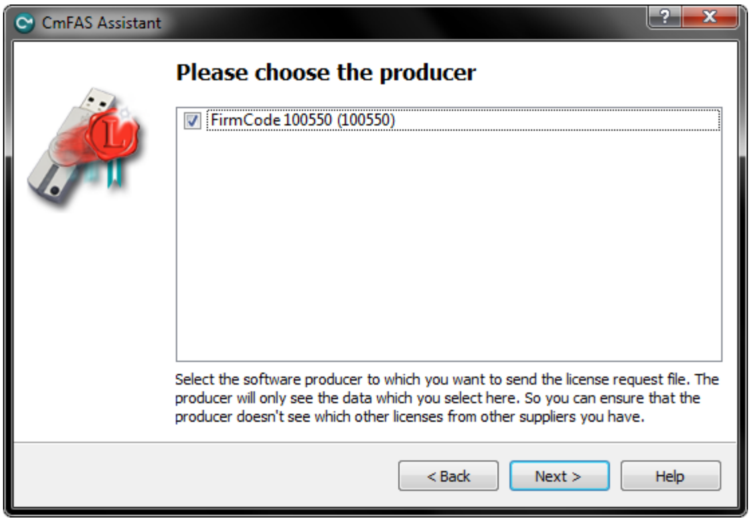
● Extend existing license

Choose this option if you want to change an existing license or to add new licenses to an existing license of the same producer.

○ Add license of a new producer

Choose this option if you want to add a new license and there are no licenses from this producer in the selected license container.

6. Select **FirmCode 100550 (100550)** and click **Next**:

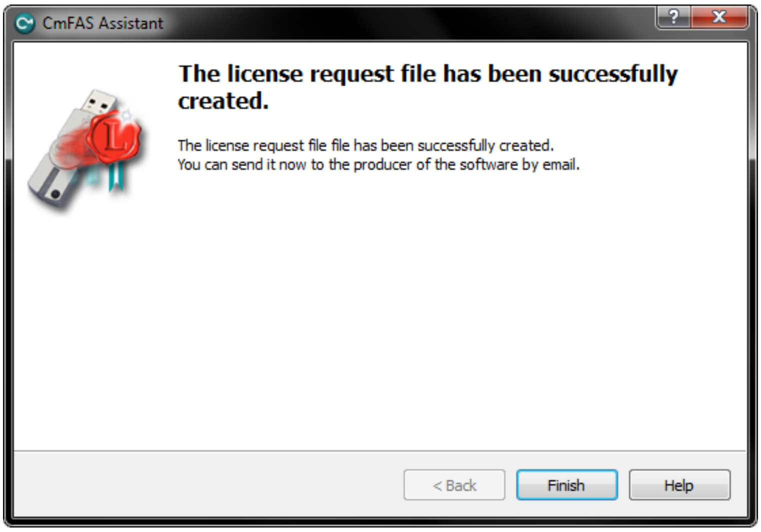


Please choose the producer

☑ FirmCode 100550 (100550)

Select the software producer to which you want to send the license request file. The producer will only see the data which you select here. So you can ensure that the producer doesn't see which other licenses from other suppliers you have.

7. Click **Finish**:



The license request file has been successfully created.

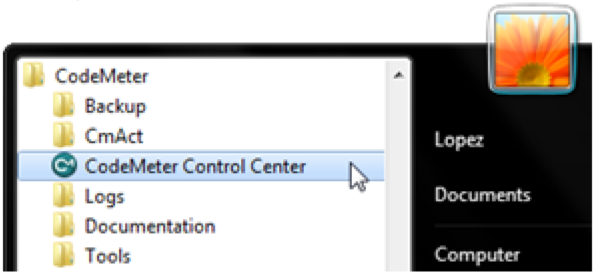
The license request file file has been successfully created.
You can send it now to the producer of the software by email.

8. E-mail the RaC file to your GenISys support engineer for processing and wait for the RaU file in our response.

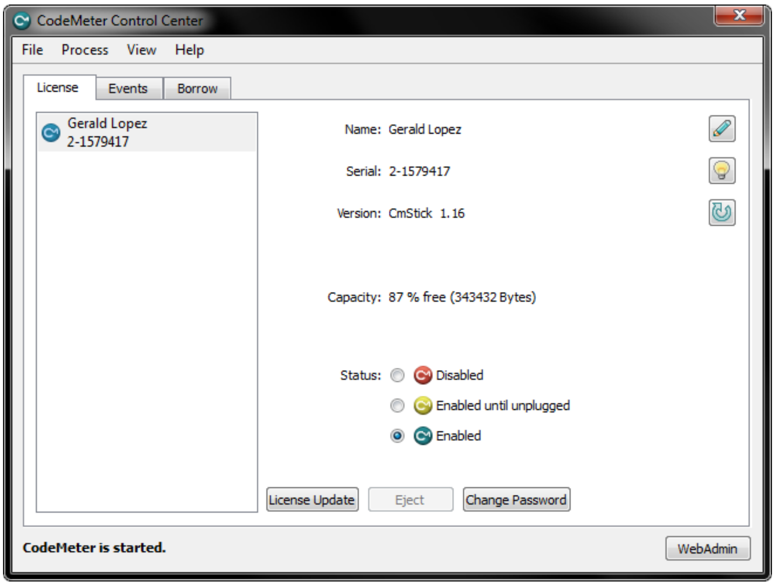
13.1.1.2 Updating the License with the RaU File

You will receive a “*.WibuCmRaU” file from Genlsys. We refer to this as the RaU file.

1. Start the Codemeter Control Center:
   a. In Linux, run the command CodeMeterCC
   b. In Windows, go to All Programs and then CodeMeter



2. Click the **License Update** button:



CodeMeter Control Center

Gerald Lopez
2-1579417

Name: Gerald Lopez

Serial: 2-1579417

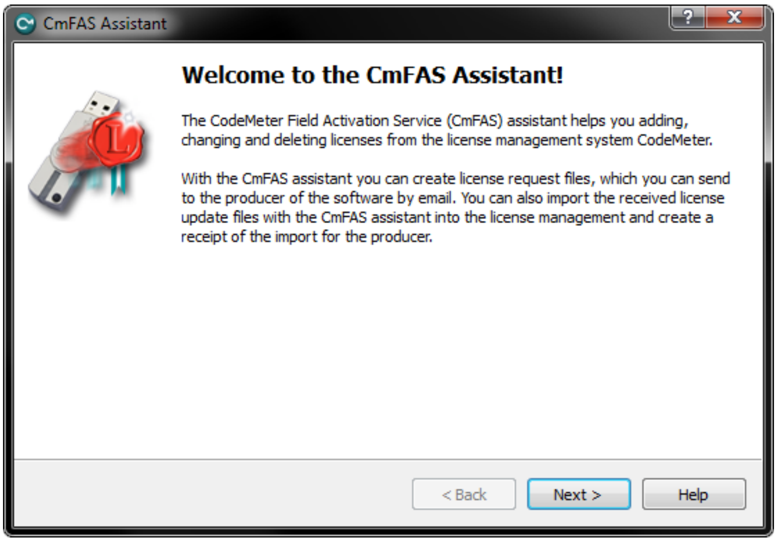
Version: CmStick 1.16

Capacity: 87 % free (343432 Bytes)

Status: ○ Disabled
○ Enabled until unplugged
● Enabled

CodeMeter is started.

3. Click **Next** to start the Field Activation Service:



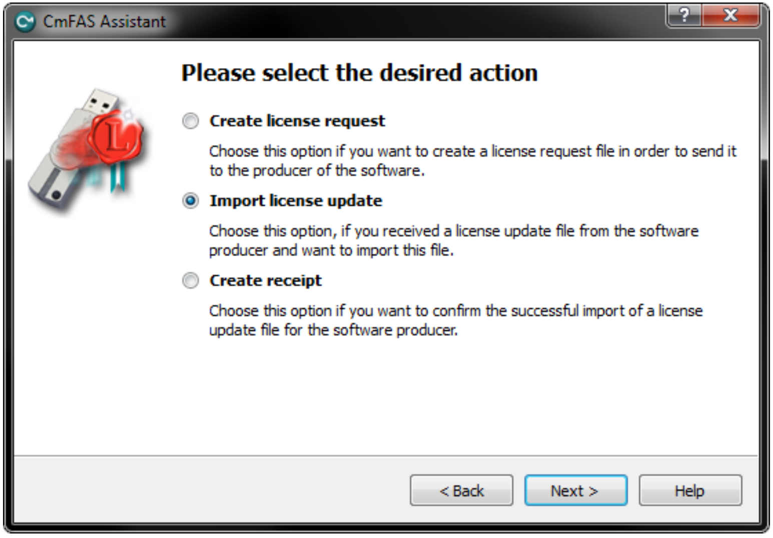
CmFAS Assistant

Welcome to the CmFAS Assistant!

The CodeMeter Field Activation Service (CmFAS) assistant helps you adding, changing and deleting licenses from the license management system CodeMeter.

With the CmFAS assistant you can create license request files, which you can send to the producer of the software by email. You can also import the received license update files with the CmFAS assistant into the license management and create a receipt of the import for the producer.

4. Select **Import license update** and click **Next**:



CmFAS Assistant

Please select the desired action

○ Create license request

Choose this option if you want to create a license request file in order to send it to the producer of the software.

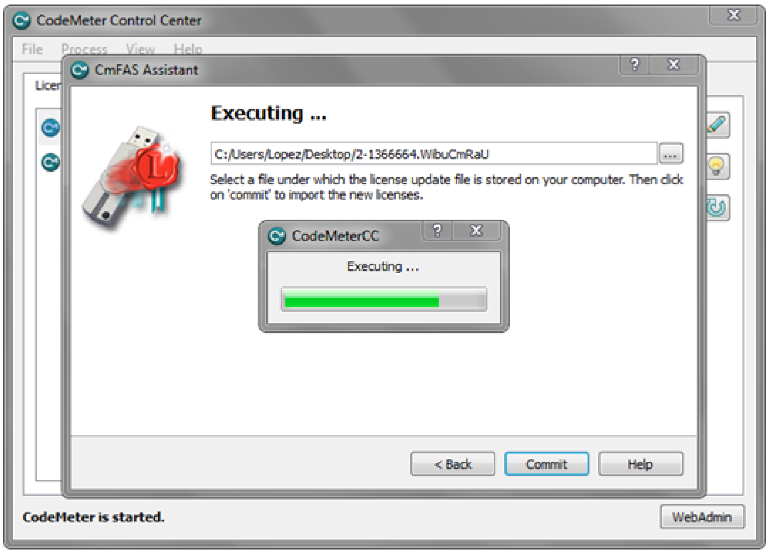
● Import license update

Choose this option, if you received a license update file from the software producer and want to import this file.

○ Create receipt

Choose this option if you want to confirm the successful import of a license update file for the software producer.

5. Click **Commit** and wait until the execution of the update is complete. It may take a few minutes to update.



Executing ...

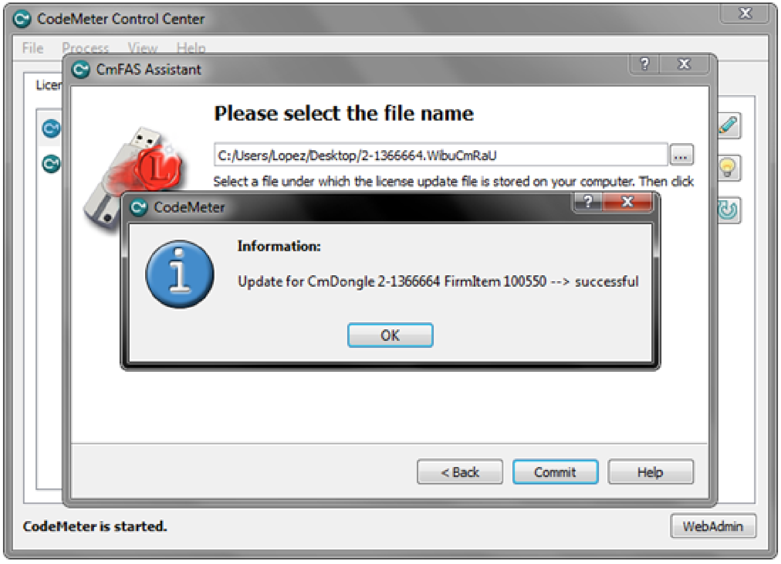
C:/Users/Lopez/Desktop/2-1366664.WibuCmRaU

Select a file under which the license update file is stored on your computer. Then click on 'commit' to import the new licenses.

Executing ...

CodeMeter is started.

6. Once executed, you will get a confirmation dialog stating that the update was successful. The license server update is complete.



Please select the file name

C:/Users/Lopez/Desktop/2-1366664.WibuCmRaU

Select a file under which the license update file is stored on your computer. Then click

Information:

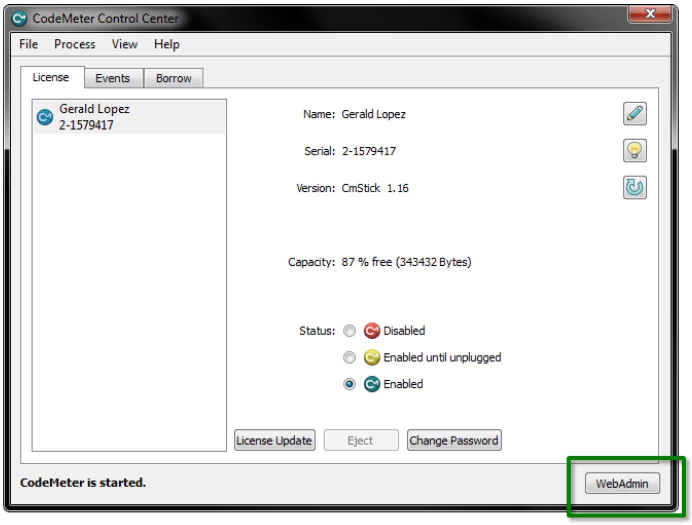
Update for CmDongle 2-1366664 FirmItem 100550 --> successful

CodeMeter is started.

Your license update should now be complete. Thank you for keeping your license current. If you have any questions, comments or concerns, please contact support at support@genisys-gmbh.com.

13.1.2 License Server Client Configuration

The configuration of the license server and client environment is done via the WebAdmin interface. This interface is started from within the CodeMeter Control Center using the WebAdmin button at the lower right. Execute the WebAdmin respectively on the server or client and complete the steps listed below.



Name: Gerald Lopez

Serial: 2-1579417

Version: CmStick 1.16

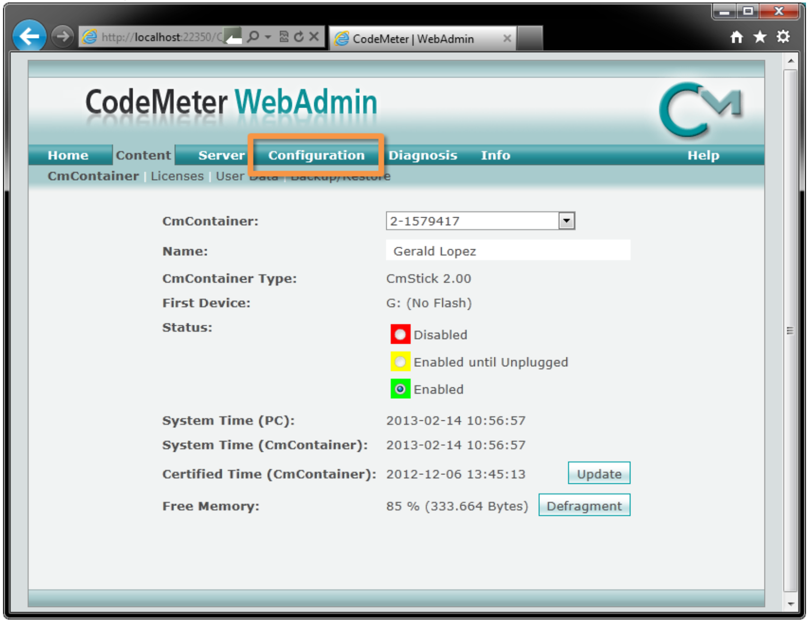
Capacity: 87 % free (343432 Bytes)

Status: ○ Disabled ○ Enabled until unplugged ● Enabled

CodeMeter is started.

13.1.2.1 Server

1. When the CodeMeter WebAdmin has opened in your default web browser, the start page should appear. Click on **Configuration**:


http://localhost:22350/
CodeMeter | WebAdmin
CodeMeter WebAdmin
Home Content Server Configuration Diagnosis Info Help
CmContainer Licenses User Data Backup/Restore

CmContainer: 2-1579417

Name: Gerald Lopez

CmContainer Type: CmStick 2.00

First Device: G: (No Flash)

Status: ○ Disabled ○ Enabled until Unplugged ● Enabled

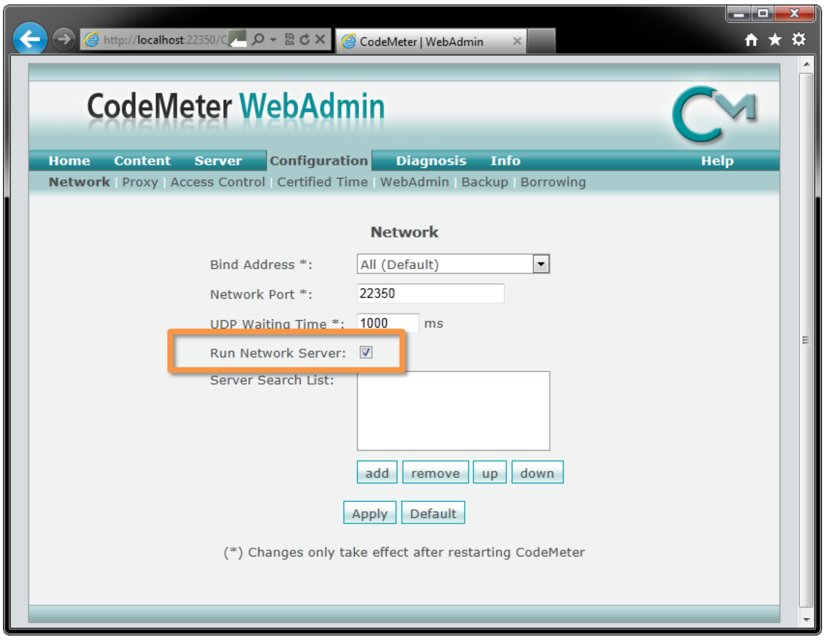
System Time (PC): 2013-02-14 10:56:57

System Time (CmContainer): 2013-02-14 10:56:57

Certified Time (CmContainer): 2012-12-06 13:45:13 Update

Free Memory: 85 % (333.664 Bytes) Defragment

2. In the Configuration,start the license server by enabling the check box to **Run Network Server** and click **Apply**:



Network

Bind Address *: All (Default)

Network Port *: 22350

UDP Waiting Time *: 1000 ms

Run Network Server: ☑

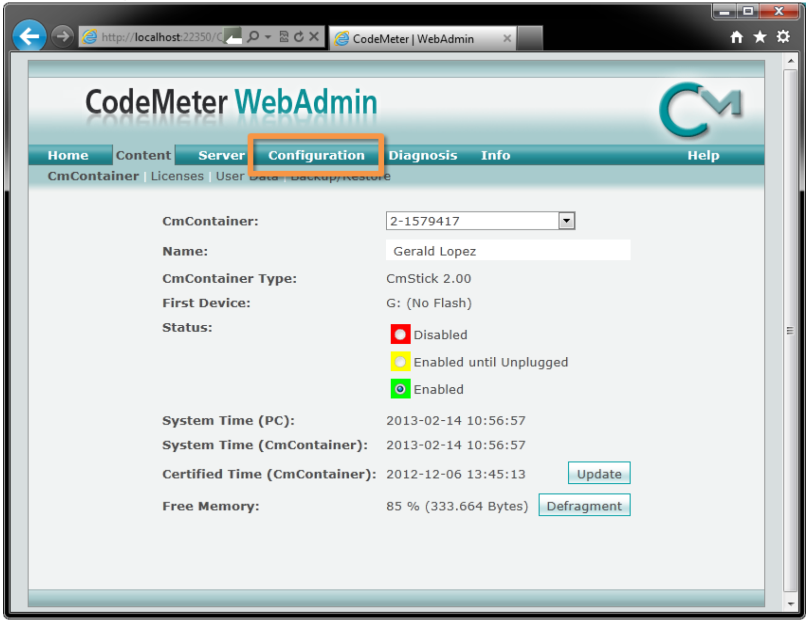
Server Search List:

(*) Changes only take effect after restarting CodeMeter

3. This completes the license server configuration.

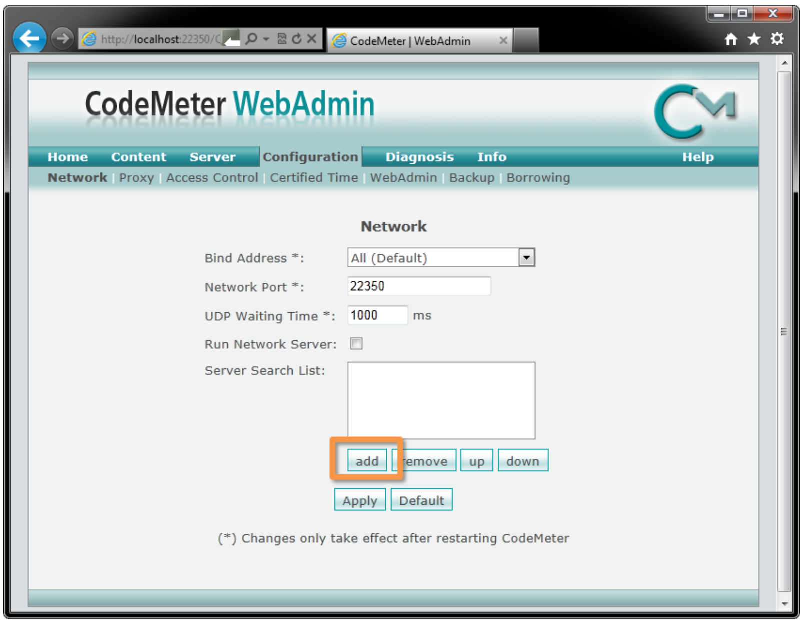
13.1.2.2 Client

1. When the CodeMeter WebAdmin has opened in your default web browser, the start page should appear. Click on **Configuration**:



CmContainer:	2-1579417
Name:	Gerald Lopez
CmContainer Type:	CmStick 2.00
First Device:	G: (No Flash)
Status:	○ Disabled ○ Enabled until Unplugged ● Enabled
System Time (PC):	2013-02-14 10:56:57
System Time (CmContainer):	2013-02-14 10:56:57
Certified Time (CmContainer):	2012-12-06 13:45:13
Free Memory:	85 % (333.664 Bytes)

2. Click the **add** button below the Server Search List:



Network

Bind Address *: All (Default)

Network Port *: 22350

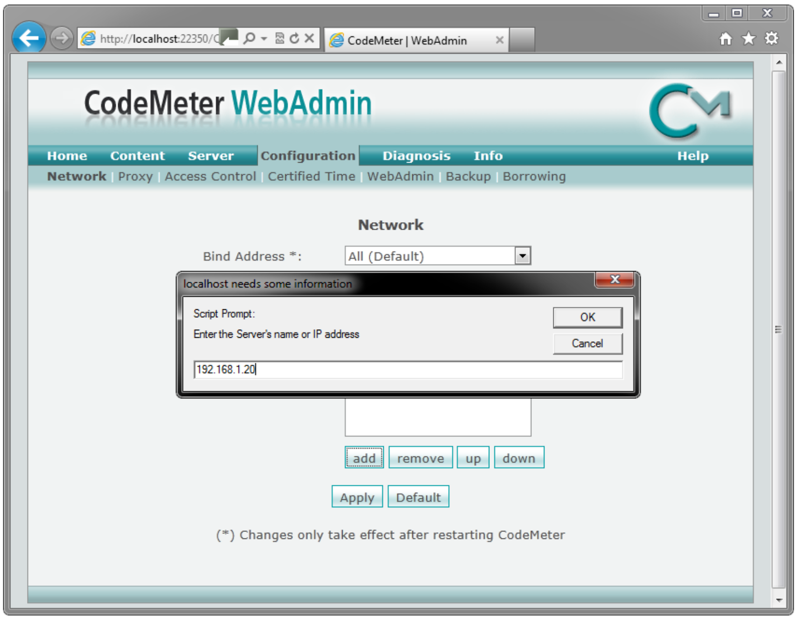
UDP Waiting Time *: 1000 ms

Run Network Server: ☐

Server Search List:

(*) Changes only take effect after restarting CodeMeter

3. Enter the IP address of the license server or its domain name and click **OK**:



Network

Bind Address *: All (Default)

localhost needs some information

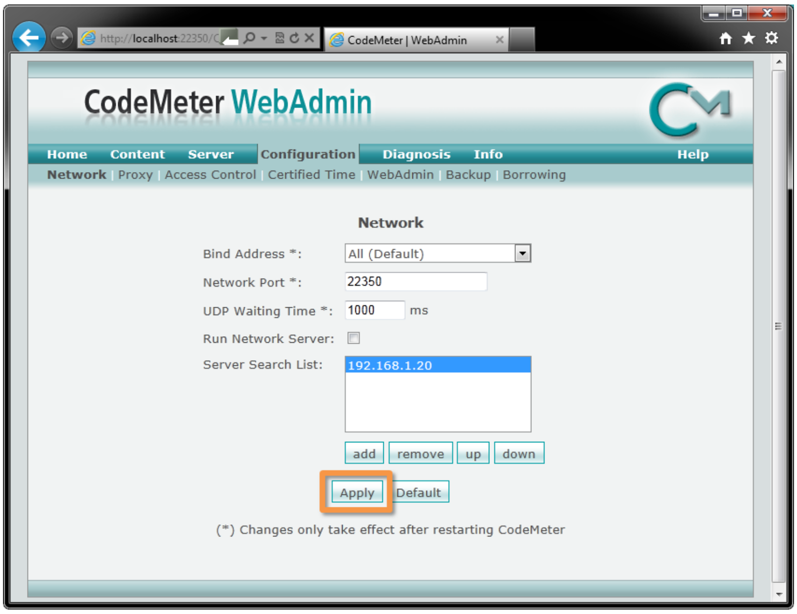
Script Prompt:

Enter the Server's name or IP address

192.168.1.20

(*) Changes only take effect after restarting CodeMeter

4. Click **Apply**:



Network

Bind Address *: All (Default)

Network Port *: 22350

UDP Waiting Time *: 1000 ms

Run Network Server: □

Server Search List: 192.168.1.20

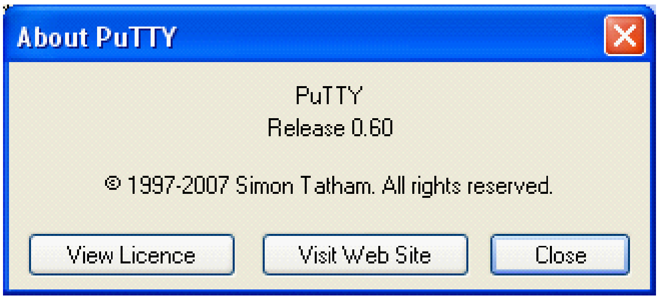
(*) Changes only take effect after restarting CodeMeter

5. The CodeMeter client has been configured and the license should now be accessible for the client computer.

13.2 Remote usage

Using VNC through an SSH tunnel

To use VNC to access the Unix machines remotely, you will need to come in through an **SSH** tunnel using PuTTY:



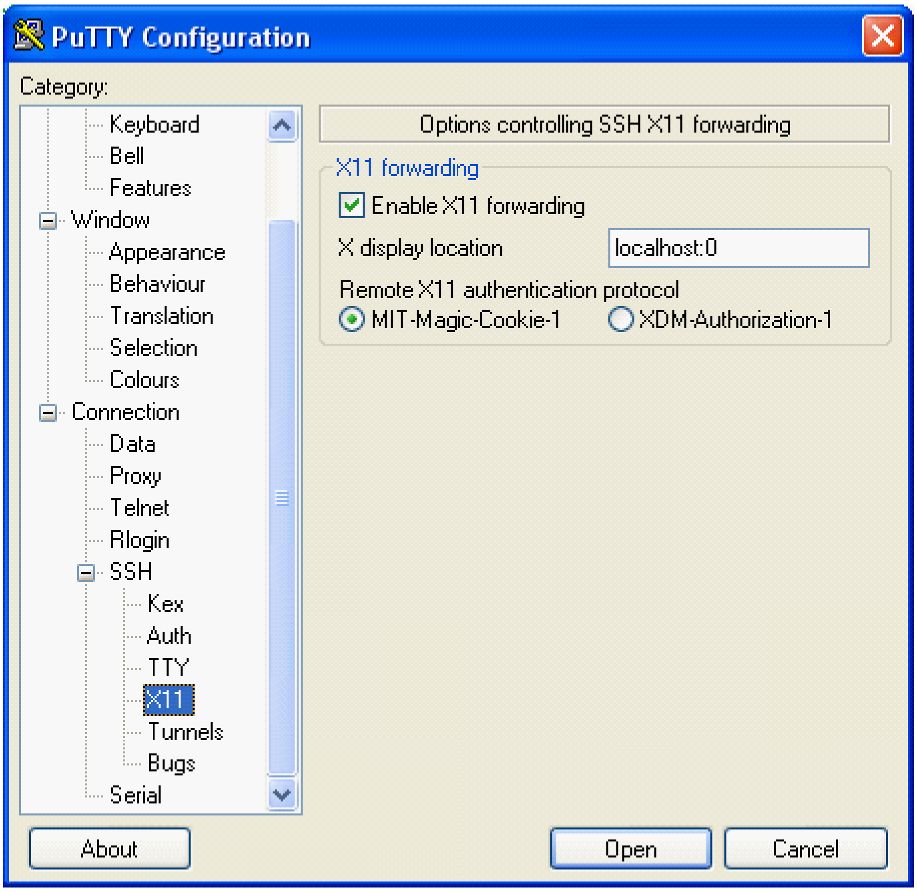
About PuTTY

PuTTY
Release 0.60

© 1997-2007 Simon Tatham. All rights reserved.

Start to make a connection as shown in the following picture (in this example galaxy.--.cnf..edu is the remote system name):

Next click [X11] and select „Enable X11 forwarding“ and „localhost:0“ as the X display location:



PuTTY Configuration

Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

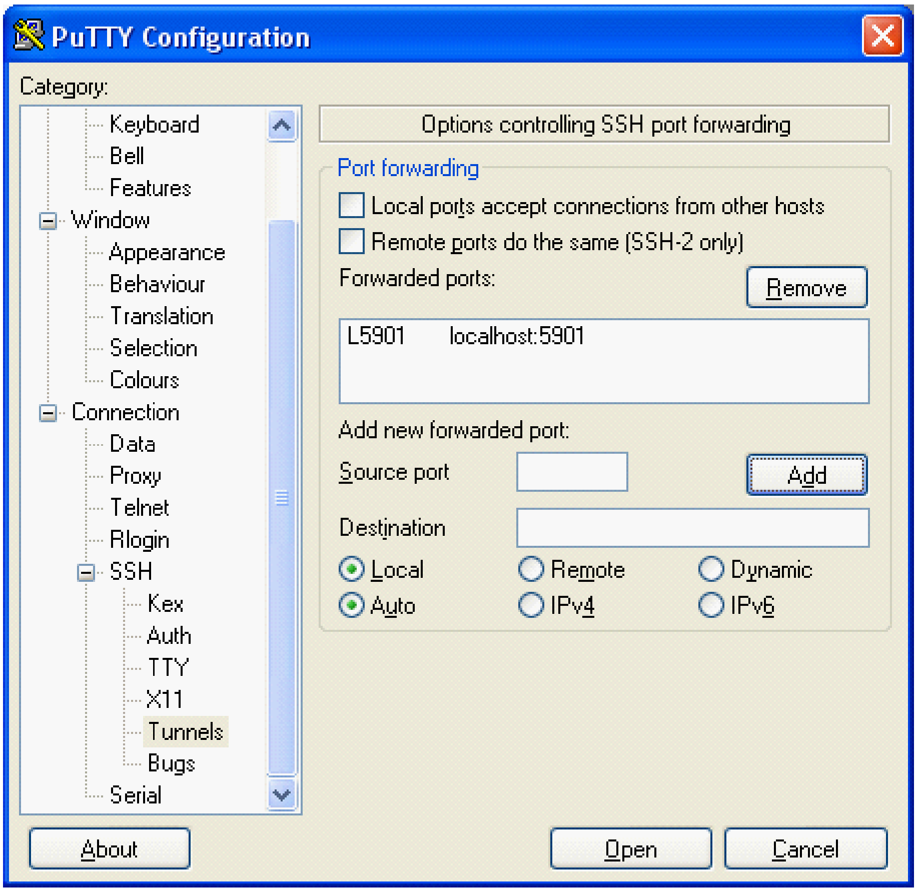
X display location: localhost:0

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

Next, click on [Tunnels] in the left-hand side of the Putty control screen, and fill in the Source port and Destination host:port descriptor, as shown below:

Next, click on the [Add] button to add this new tunnel. Putty will look like this:



PuTTY Configuration

Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

L5901 localhost:5901

Add new forwarded port:

Source port:

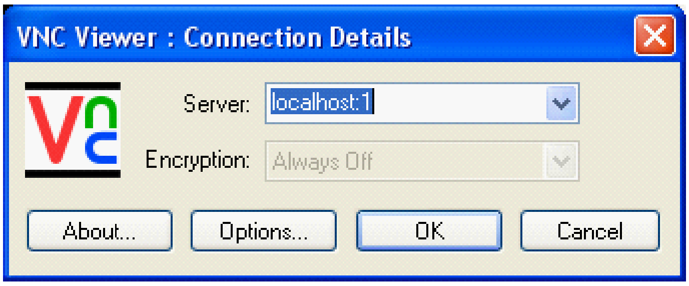
Destination:

● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

Next, click on the [Open] button, and log in to the remote system. Your tunnel is now active. You do not need to run anything else in this putty terminal window. Make sure that your xsession is **:1**, that you can use this connection to start the vncserver!

Finally, run VNCVIEWER on your windows host, and enter "localhost:1" as the server to connect to. VNC maps display 1 to port 5901, which SSH is tunneling through its own port (port 22).

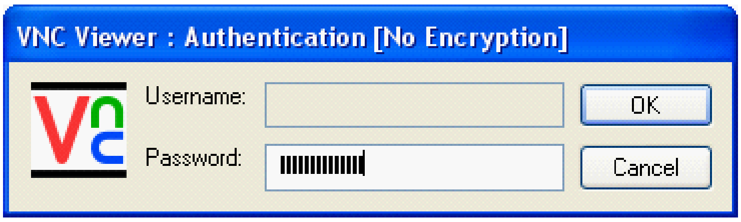


VNC Viewer : Connection Details

Server: localhost:1

Encryption: Always Off

[OK]



VNC Viewer : Authentication [No Encryption]

Username:

Password: ||||||||||||

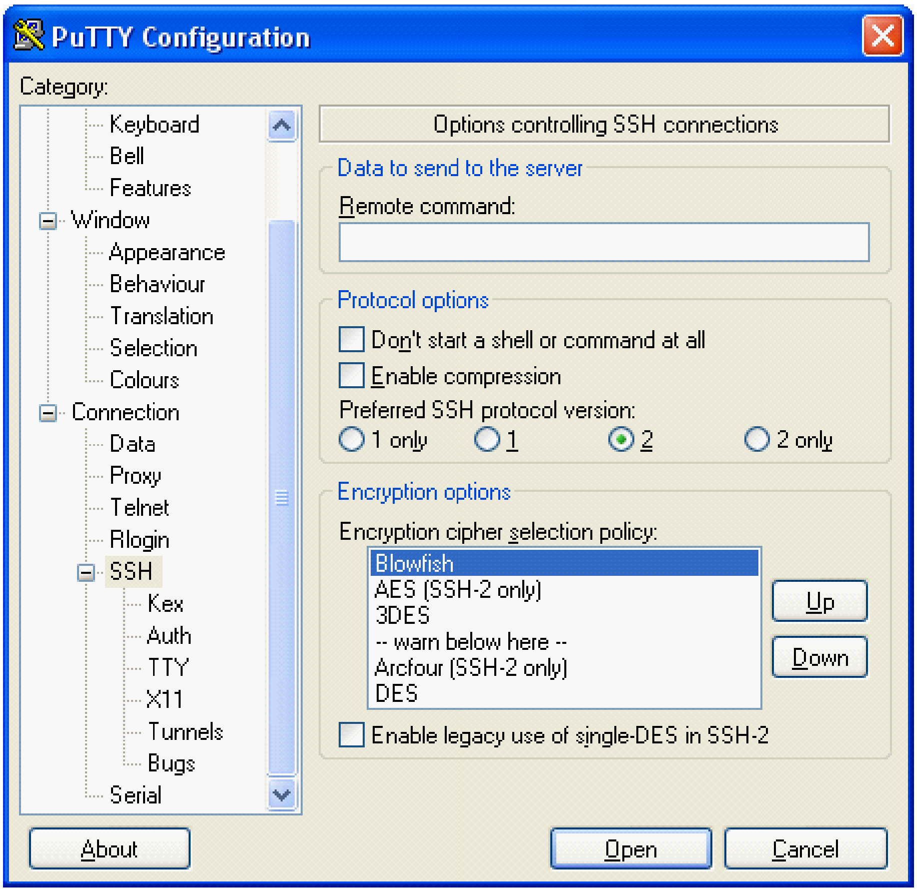
[OK]

When you are done do not *log out* of the session, rather just quit the vncveiwer on the local host and then kill the server (from the putty window):

host$ vncserver -kill :1

Modifications:
The default window manager is twm. If you want something else, e.g. gnome/metacity edit ~/ .vnc/xstartup: replace twm with gnome-session

To speed up things even more you can change the cipher algorithm to from SSH-2 to Blowfish:



Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

☐ Don't start a shell or command at all

☐ Enable compression

Preferred SSH protocol version:

○ 1 only ○ 1 ● 2 ○ 2 only

Encryption options

Encryption cipher selection policy:

Blowfish
AES (SSH-2 only)
3DES
-- warn below here --
Arcfour (SSH-2 only)
DES

☐ Enable legacy use of single-DES in SSH-2

13.3 Reporting a Problem

Whenever a problem occurs during the usage of the software, please report this to us. It would be helpful if the information below can be given to quickly identify and reproduce the problems observed:

- Description of the problem
- How it can be reproduced
- Files needed to reproduce the problem

- Version of the operating system and available HDD space and physical memory
- Version of BEAMER (Available in the Help Menu selection)

Please send this information to our support team at support@genisys-gmbh.com. Should the files be very big, please use our ftp server for the transfer.

13.4 Requesting Enhancements

We are always glad to enhance our software based on your requests.

Please contact our support team or our local sales manager for a discussion on how we can best meet your needs.

13.5 Training

To give you the best usage of the software we offer training either on site or via a remote computer connection.

Please contact our local sales manager to arrange the details.

13.6 Tips and Troubleshooting

13.6.1 FAT32 File System

The FAT32 file system doesn't support a file size larger than 4GB. You can run into this issue when exporting larger machine format files or during processing large temp files need to be created.

Try to avoid this file system type if possible. As an alternative, please use NTFS for your Windows platform. NTFS will allow file sizes larger than 4GB.

13.6.2 License Feature Not Found

In Linux, it can be observed that killing BEAMER while in module execution leads to a blocked license of that specific module. What happens is that BEAMER, executed via command line, is killed using CTRL+C. Inspection of the running processes will reveal that a BEAMER process is still running, and it is this zombie process that is blocking the specific license feature to be used.

Since formatters are separate license features, one might observe the aforementioned occurrence by finding that a specific formatter is missing from the pull down list.

To resolve this issue, kill the zombie process of BEAMER.

Index