

SILVACO

DevEdit

User's Manual

SILVACO, Inc.

4701 Patrick Henry Drive, Bldg. 1
Santa Clara, CA 95054

Phone (408) 567-1000

Web: www.silvaco.com

May 10, 2010

DevEdit
User's Manual
Copyright 2010

SILVACO, Inc.
4701 Patrick Henry Drive, Bldg. #2
Santa Clara, CA 95054

Phone: (408) 567-1000
Web: www.silvaco.com

The information contained in this document is subject to change without notice.

SILVACO, Inc. MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

SILVACO, Inc. shall not be held liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright laws of the United States. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of SILVACO, Inc.

ACCUCELL, ACCUCORE, ACCUMODEL, ACCUTEST, ATHENA, ATHENA 1D, ATLAS, BLAZE, C-INTERPRETER, CATALYSTDA, CATALYSTAD, CELEBRITY, CELEBRITY C++, CIRCUIT OPTIMIZER, CLARITYRLC, CLEVER, DECKBUILD, DEVEDIT, DEVEDIT3D, DEVICE3D, DISCOVERY, EDA OMNI, EDIF WRITER, ELITE, EXACT, EXPERT, EXPERTVIEWS, FERRO, GATEWAY, GIGA, GIGA3D, GUARDIAN, GUARDIAN DRC, GUARDIAN LVS, GUARDIAN NET, HARMONY, HIPEX, HIPEX-C, HIPEX-NET, HIPEX-RC, HYPERFAULT, LASER, LED, LISA, LUMINOUS, LUMINOUS3D, MAGNETIC, MAGNETIC3D, MASKVIEWS, MC DEPO/ETCH, MC DEVICE, MC IMPLANT, MERCURY, MIXEDMODE, MIXEDMODE3D, MOCASIM, MODELLIB, NOISE, NOMAD, OLED, OPTOLITH, ORGANIC DISPLAY, ORGANIC SOLAR, OTFT, PROMOST, QUANTUM, QUANTUM3D, QUEST, REALTIMEDRC, RESILIENCE, S-PISCES, S-SUPREM3, S-SUPREM4, SCOUT, SDDL, SFLM, SiC, SILOS, SMARTLIB, SMARTSPICE, SMARTSPICE SEE, SMARTSPICERF, SMARTVIEW, SIMULATION STANDARD, SOLVERLIB, SPAYN, SPDB, SPIDER, SPRINT, STELLAR, TCAD DRIVEN CAD, TCAD OMNI, TFT, TFT3D, THERMAL3D, TONYPLOT, TONYPLOT3D, TWISTER, TWISTERFP, VCSEL, UTMOST, UTMOST III, UTMOST IV, UTMOST IV- FIT, UTMOST IV-MEASURE, UTMOST IV- OPTIMIZATION, VICTORY, VICTORYCELL, VICTORYDEVICE, VICTORYPROCESS, VICTORYSTRESS, VIRTUAL WAFER FAB, VWF, VWF AUTOMATION TOOLS, VWF INTERACTIVE TOOLS, VWF MANUFACTURING TOOLS, and VYPER are trademarks of SILVACO, Inc.

All other trademarks mentioned in this manual are the property of their respective owners.

Copyright © 1984 - 2010, SILVACO, Inc.

How to Read this Manual

Style Conventions		
Font Style/Convention	Description	Example
•	This represents a list of items or terms.	<ul style="list-style-type: none"> • Bullet A • Bullet B • Bullet C
1. 2. 3.	This represents a set of directions to perform an action.	To open a door: <ol style="list-style-type: none"> 1. Unlock the door by inserting the key into keyhole. 2. Turn key counter-clockwise. 3. Pull out the key from the keyhole. 4. Grab the doorknob and turn clockwise and pull.
→	This represents a sequence of menu options and GUI buttons to perform an action.	File→Open
Courier	This represents the commands, parameters, and variables syntax.	HAPPY BIRTHDAY
New Century Schoolbook Bold	This represents the menu options and buttons in the GUI.	File
<i>New Century Schoolbook Italics</i>	This represents the equations.	$abc=xyz$
Note:	This represents the additional important information.	Note: Make sure you save often while running an experiment.
NEW CENTURY SCHOOLBOOK IN SMALL CAPS	This represents the names of the SILVACO products.	ATHENA and ATLAS.

Table of Contents

Chapter 1

Introduction	1-1
1.1: What is DevEdit	1-1
1.1.1: The Problem	1-1
1.1.2: The Solution	1-1
1.1.3: When to Use DevEdit	1-1
1.1.4: When Not to Use DevEdit	1-1
1.1.5: Getting Started	1-2
1.2: Base Window	1-3
1.2.1: Layout and Functionality	1-3
1.2.2: Control Panel	1-4
1.2.3: Main Panel Controls	1-4
1.2.4: Control Windows	1-4
1.3: FILE CONTROL	1-5
1.3.1: Using Devedit	1-5
1.3.2: Loading a SILVACO Standard Structure File	1-5
1.3.3: Writing to a File	1-5
1.3.4: Difference - SILVACO Standard vs. Devedit	1-6
1.3.5: Loading a Command File	1-6
1.3.6: Default Files	1-6

Chapter 2

Tutorial	2-1
2.1: Goal And Purpose Of Creating A New Mesh	2-1
2.2: EXAMPLE 1 - CREATE A NEW STRUCTURE	2-2
2.2.1: Work Area	2-2
2.2.2: Defining Regions	2-3
2.2.3: Defining Impurities	2-8
2.2.4: Mesh Creation	2-9
2.2.5: Saving The File	2-12
2.3: EXAMPLE 2 - REMESHING AN EXISTING STRUCTURE	2-14
2.3.1: Loading The Structure	2-14
2.3.2: Obtain Existing Structure	2-14
2.3.3: Structure Editing	2-14
2.3.4: Display	2-15
2.3.5: Mesh Creation	2-16
2.3.6: Saving The File	2-21
2.4: Advanced Features	2-25
2.4.1: 3D Structures	2-25
2.4.2: Combining Two ATHENA Structures into a Single Device	2-27
2.4.3: Stretch and Cut	2-28
2.4.4: Circular Devices	2-29
2.4.5: Summary	2-30

Chapter 3

Editing Functions	3-1
3.1: STRUCTURE EDITING	3-1
3.1.1: Zooming	3-1
3.1.2: Panning	3-1
3.1.3: Editing Summary	3-1
3.1.4: Selecting The Resolution	3-1
3.2: EDITING REGIONS	3-2
3.2.1: Adding a Region	3-2
3.2.2: Selecting Region Material	3-2
3.3: DRAWING REGIONS	3-3
3.3.1: Setting Base Impurity	3-4
3.3.2: Deleting Regions	3-4
3.3.3: Modify Regions	3-5
3.3.4: Deleting Boundary Points	3-5
3.4: DOPING DEFINITION	3-6
3.4.1: Defining an Impurity Source Line	3-6
3.4.2: Defining an Impurity Source Box	3-7
3.4.3: Doping Source Attributes	3-7
3.4.4: Deleting Source Objects	3-7
3.4.5: 3D Doping	3-7
3.5: MESHBUILD MESHING	3-8
3.5.1: Boundary Conditioning	3-8
3.5.2: Limitations	3-9
3.5.3: Mesh Constraints	3-9
3.5.4: Adaptive Meshing	3-10
3.5.5: Refinement	3-10
3.5.6: Manually Refining The Mesh	3-11
3.5.7: Manually Relaxing A Mesh	3-11
3.5.8: Tensor Product Mesh	3-11
3.5.9: Work Area Resizing	3-11
3.5.10: Defining 3D Structures	3-11
3.6: CREATING A NEW MESH	3-12
3.6.1: Setting The Mesh Controls	3-12
3.6.2: Base Mesh Parameters	3-12
3.6.3: Refining On Impurities	3-12
3.6.4: Mesh Constraints	3-12
3.6.5: Final Meshing	3-12
3.6.6: Saving the SILVACO Standard Structure File	3-12
3.7: IMPURITIES	3-13
3.7.1: Viewing Impurities	3-13
3.7.2: Impurity Definition	3-13
3.7.3: Impurities Loaded From A Structure	3-13
3.7.4: Add Impurity Mode	3-14
3.7.5: Defining An Impurity Source Area	3-14
3.7.6: Defining Impurity Roll-off Direction	3-15
3.8: ROLL-OFF FUNCTION	3-19
3.8.1: Analytic Functions	3-19
3.8.2: Analytic Functions (Dist)	3-20
3.8.3: Doping Profiles	3-21

3.8.4: Join Function 2D	3-22
3.8.5: Join Function 3D	3-23
3.8.6: Deleting Impurities	3-24
3.8.7: Editing Impurities	3-24
3.8.8: Combining Impurity Rolloffs	3-24
Chapter 4	
Statements	4-1
4.1: Overview	4-1
4.1.1: Cards And Parameters	4-1
4.2: BASE.MESH	4-3
4.3: BOUNDARY.CONDITIONING	4-4
4.4: CONSTRAINT.MESH	4-6
4.5: CUT	4-10
4.6: DEPOSIT	4-11
4.7: FLIP	4-12
4.8: IMPURITY	4-13
4.9: IMPURITY REFIN	4-17
4.10: INITIALIZE	4-19
4.11: JOIN	4-20
4.12: MESH	4-21
4.13: MIRROR	4-22
4.14: MOVE	4-23
4.15: PROFILE	4-24
4.16: QUIT	4-25
4.17: REFIN	4-26
4.18: REGION	4-27
4.19: RENUMBER.REGIONS	4-29
4.20: SOURCE	4-30
4.21: STRETCH	4-31
4.22: STRUCTURE	4-34
4.23: SUBSTRATE	4-35
4.24: WORK.AREA	4-36
4.25: Z.PLANES	4-37
4.26: GENERIC PARAMETER - BOOLEAN TYPE	4-38
4.27: GENERIC PARAMETER - COLOR	4-39
4.28: GENERIC PARAMETER - IMPURITY	4-40
4.28.1: Possible Values	4-40
4.29: GENERIC PARAMETER - MATERIAL	4-48
4.29.1: Possible Values	4-48
4.30: GENERIC PARAMETER - PATTERN	4-54

This page is intentionally left blank.

1.1: What is DevEdit

DEVEDIT is a device structure editor. It can be used to generate a new mesh on an existing structure and can be used to create or modify a device. These devices can then be used by SILVACO 2-D and 3-D simulators. DEVEDIT can be used as a simulator under DECKBUILD or through a Graphical User Interface (GUI).

Note: The GUI is not available on Windows.

1.1.1: The Problem

A limitation of device simulators prior to DEVEDIT was inadequate or poor structure meshes. A mesh containing too many obtuse triangles or an insufficient number of triangles (too coarse a grid) may provide an inaccurate result or no result at all. A mesh containing too many triangles (too fine a grid) can result in excessive simulator processing time. Since the time most simulators use grows geometrically with the number of triangles (or grid points), it is critical to keep the number of triangles down to a reasonable number. Using simulators, such as SSUPREM4, to create non-uniform meshes tend to be very time consuming and require a great deal of effort.

1.1.2: The Solution

DEVEDIT resolves these problems by allowing structures to be created or read into DEVEDIT in the form of SILVACO Standard Structure Files. The mesh contained in the file can then be replaced using the MESHBUILD algorithm. You can refine the mesh by setting parameters that describe critical areas or by simply pointing to the areas, which require refinement.

In the process of creating a structure, you can define a device by drawing it on the screen. DEVEDIT can also perform analytic implants using built-in equations or cut lines from other simulators. Constraints are then placed on the mesh to describe the critical areas of the device.

1.1.3: When to Use DevEdit

Use DEVEDIT when you want to perform the following operations:

- Defining a device interactively on the screen for subsequent device or process simulations.
- Remeshing a device structure between process simulation and device test simulations, when the process simulator does not create a good grid for the device simulator.
- Remeshing a device structure during a process or device simulation, when the mesh is no longer adequate for the next simulation step.

1.1.4: When Not to Use DevEdit

You should not use DEVEDIT to perform the following operations:

- Replacing numerical process simulations where accuracy is required.
- Meshing 1D device structures.

1.1.5: Getting Started

You can run DEVEDIT from the UNIX prompt or from DECKBUILD. There are two file types that DEVEDIT can read: SILVACO standard structure file format (common to all SILVACO simulators) and command format (a list of DEVEDIT commands which create a structure). The structure file format contains information, such as triangles, impurity values, borders. The command format is normally used when starting a new device mesh. It contains the list of instructions that describe the current state of mesh development. Although the DEVEDIT command file is usually smaller than the structure file, it contains information, such as the state of DEVEDIT, impurity equations, and meshing modes.

Startup

You can start DEVEDIT one of two modes: 2D and 3D.

For 2-D mode, type:

```
devedit
```

For 3-D mode, type:

```
devedit3d
```

To start DEVEDIT in GUI mode, use one of the following commands for a UNIX prompt.

Command	Description
<code>devedit &</code>	This starts DEVEDIT with no active device.
<code>devedit fred.str &</code>	This starts DEVEDIT with the SILVACO Standard structure <code>fred.str</code> loaded.
<code>devedit fred.de &</code>	This starts DEVEDIT with the command file <code>fred.de</code> loaded.
<code>devedit3d &</code>	This starts DEVEDIT in 3-D mode with no active device.
<code>devedit3d fred3d.de &</code>	This starts DEVEDIT with the 3-D command file <code>fred.de</code> loaded. DEVEDIT does not currently support loading 3-D structure files.

Note: Make sure you set your DISPLAY environmental variable to an active X window screen.

1.2: Base Window

1.2.1: Layout and Functionality

The DEVEDIT base window display (Figure 1-1) is made up of several sections:

- **Control Buttons** - These menu buttons are used to control all DEVEDIT actions.
- **Main Canvas** - This area is used to show a graphical representation of the device.
- **Main Panel** - Displays a list of the regions in the current device and allows a region to be selected for region specific commands. The list contains all regions by name, number, legend, and (for color terminals) color.
- **User Added Impurities List** - Displays a list of the user-added impurities as they are selected from the **Control panel Impurities** menu, or read from a command file (impurities read from SILVACO standard structure files do not appear in this list). This list also allows impurities to be selected for impurity specific commands.

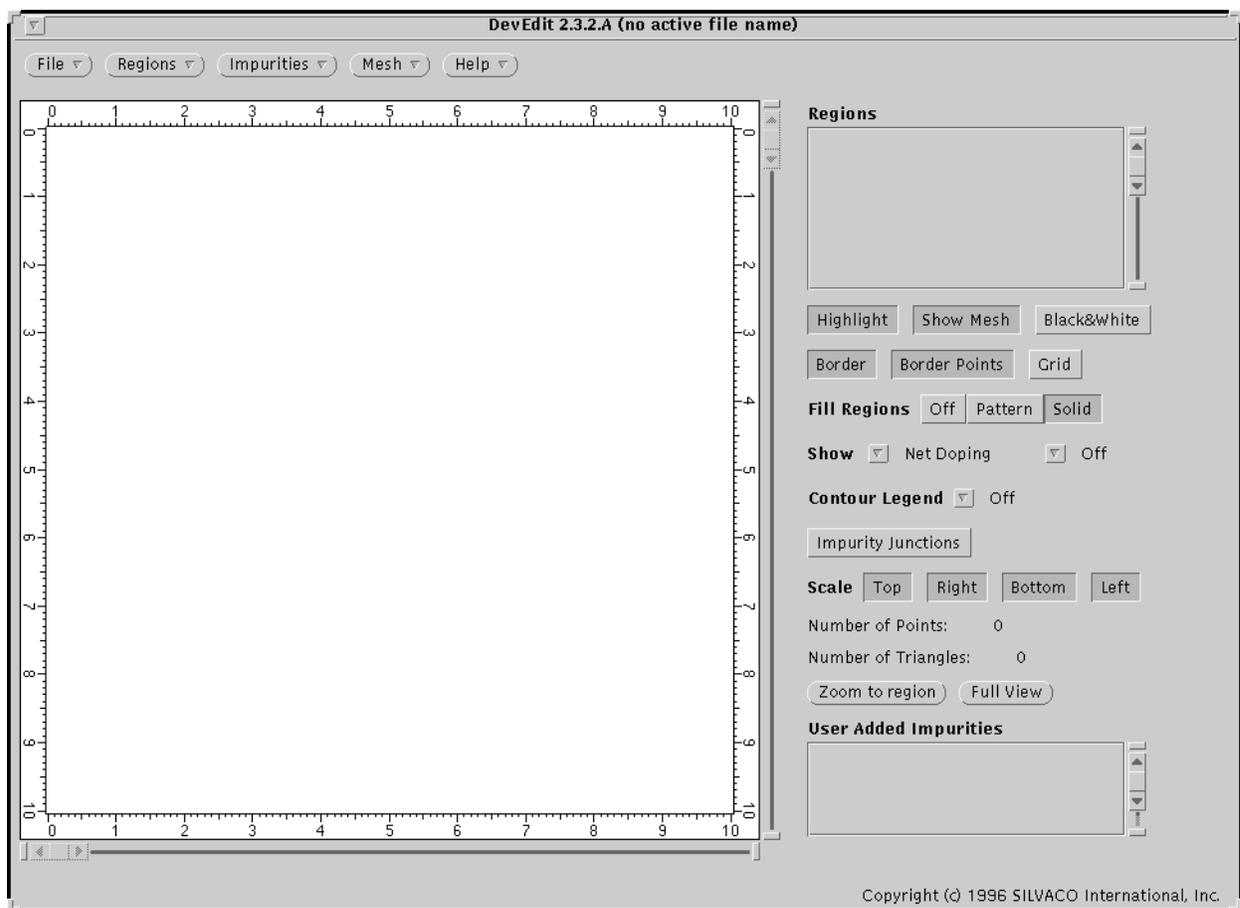


Figure 1-1:Base Window Display

1.2.2: Control Panel

Commands are performed by selecting the appropriate option from one of the following menus or buttons displayed on the control panel.

- **Files** - Contains a menu of load/save file operations.
- **Regions** - Contains a menu of various material region control options.
- **Impurities** - Contains various options for control of impurities.
- **Mesh** - Contains a menu of various mesh control options.
- **Help** - Provides interactive help information.

1.2.3: Main Panel Controls

Controls located between the **Regions** panel and **User Added Impurities** panel control how regions and impurities are displayed. They perform the following functions:

- **Highlight** - Permits highlighting selected material region in red.
- **Show Mesh** - If a mesh exists, it can be shown or not shown to speed up display.
- **Black & White** - Allows making a black and white picture on a color screen.
- **Border** - Shows a black border around regions. It can be turned off so regions are not separated by a border.
- **Border Points** - Shows key points used to create the mesh.
- **Grid** - Shows intersect points between scales on main panel.
- **Fill Regions** - How regions should be filled in.
- **Off** - Regions are not filled.
- **Pattern** - A fill pattern is used to fill region. Ordinarily used only in Black & White mode.
- **Solid** - The region's color is used to fill region. This option is not shown on Black & White terminals.
- **Show Net Doping** - Permits selection of how doping is displayed (normally, coarse or off is selected).
- **Net Doping Legend** - Permits positioning of a legend in one of eight positions on the main panel.
- **Impurity Junctions** - Shows where the junctions are (Only shown when a mesh exists).
- **Scale** - Permits display (or not) of scale on each border of the main panel.
- **Number of Points** - Displays number of points in current structure.
- **Number of Triangles** - Displays number of triangles in current structure.
- **Zoom to Region** - Permits selection of a region on the Regions list as full view on main panel.
- **Full View** - Selects full view of work area (entire device).

1.2.4: Control Windows

During the process of DEVEDIT operation, a number of auxiliary panes can stack up obscuring the Main panel. To regain access to the Main panel, sequentially remove the auxiliary panes by choosing the **Cancel**, **Done**, or **Apply** button (as appropriate) for each panel.

1.3: FILE CONTROL

1.3.1: Using Devedit

Using DEVEDIT is largely intuitive. As a starting point, a screen appears displaying a space where you can edit a device. Menus and options appear on the right hand side of the screen. They are selected from the **Main Menu** button options along the top of the main screen. Each action may change the control panel on the right side of the main screen. Select the required menu options from the series of buttons and menus, and press **Apply** or **Done** on the control panel when complete.

1.3.2: Loading a SILVACO Standard Structure File

You can load a SILVACO Standard Structure file at any time into DEVEDIT. A SILVACO Standard Structure File may have been created by any 2-D simulator. A SILVACO standard structure can be loaded from the time DEVEDIT was invoked (see Invoking DEVEDIT from the UNIX prompt in Section 1.1.5: “Getting Started”) or by using the **File menu**. Under the **File menu** button on the left side of the DEVEDIT screen, a number of menu options allow the basic SILVACO standard structure file I/O control of DEVEDIT. To load a file, select the **Load menu** option. A popup window appears indicating a **Directory name**, a **filename** and a **Filter type**. Two types of file can be loaded into DEVEDIT. One, a SILVACO Standard Structure File that contains a description of the entire structure to be loaded. Two, a Command file that contains the complete list of instructions developed as operations performed to create a structure. If you are starting from a 2-D simulator only, you will not have a Command file to load. If starting from a point previously run in DEVEDIT, a Command file may exist. Command files can be distinguished by the filename extension .de.

To load a SILVACO Standard Structure File, select **File→Load...** In the new window, set the filter to *.str unless you do not end all your SILVACO Standard Structure Files with .str. Set the current directory and file name if needed. Currently, the mesh in the SILVACO Standard Structure File is ignored. Therefore, you must put on a new mesh before saving the structure file.

1.3.3: Writing to a File

DEVEDIT can save both structures and Command files. Saving a structure means the object displayed on the screen is saved to a SILVACO Standard Structure File (*.str). This is done only after a mesh has been generated. Saving a command file means that the complete sequence of events required to arrive at the current structure are saved while in DEVEDIT. The command file is a complete object oriented history of events required to create the current structure. To save a file, type in or select a file name. To save as a set of commands, click on the **Save Commands** button. To save a structure file, click on the **Save Structure** button. You can save over the previously loaded/saved file by selecting the **Save** option instead of the **Save as...** option.

In 3D mode, there is also a **Save Structure (Tetrahedra)** button. In this case, the **Save Structure** button saves prisms appropriate for ATLAS. The **Save Structure (Tetrahedra)** button saves tetrahedrons appropriate for VICTORYDEVICE.

There is also a special file option called **Save generic mesh commands....** This is like saving a command deck but only commands related to generic mesh commands are saved. The idea behind this is for you to load a structure, set some meshing options like refine on doping and minimum mesh size and then mesh the structure. When this is done to your liking, the mesh commands can then be saved using **Save generic mesh commands....** Then, you can load a different structure and apply the same mesh options by using **Source (command sequence)** on the previously saved file.

1.3.4: Difference - SILVACO Standard vs. Devedit

A SILVACO Standard Structure File and a DEVEDIT file (Command) differ. A SILVACO Standard Structure File constrains some information about each region, a mesh describing the device, and a list of impurities at each point. This file type must be used to pass the structure to a simulator. A DEVEDIT file contains the DEVEDIT cards (instructions) needed to create a device, the analytic impurities functions, mesh creation cards, and DEVEDIT mode settings. It does not currently support impurities (including doping and results) read in from a structure file. Therefore if a structure file is read in, a structure file must be written out. Otherwise, all the impurities not added during the current editing session are lost. Subsequent releases will rectify this problem.

When starting a new device mesh, you should save it in both card deck and structure file formats. When saved in the command format, all original information is retained, such as formulas, region colors, mesh constraints. Structure files are necessary for use by ATLAS. When a file is saved in the structure file format, only the resulting values of calculations are retained, pre-emptying the possibility of future modification of the original DEVEDIT commands.

It is especially important to save both a Command File and a Structure file when creating 3D devices. The loading of impurities and Z planes is not supported when loading 3D structure files. Therefore, this information will be lost if only the structure file is saved. Therefore, you need to save a structure file for ATLAS/VICTORY use and Command file for reloading into DEVEDIT.

1.3.5: Loading a Command File

Pull down the **File** menu and select **Load...** In the new window, set the filter to **LOADING A COMMAND FILE - *.de]*.de** unless you do not end all your DEVEDIT Command files with .de. Set the current directory and file name if needed.

1.3.6: Default Files

In addition to SILVACO Standard Structure Files and DEVEDIT Command files, there are also DEVEDIT Default Option files. These files are used by DEVEDIT to reset DEVEDIT options when loading a SILVACO Standard Structure File or when starting DEVEDIT with no file. If you are loading a DEVEDIT command, most options will return to the active settings when that command was last saved.

2.1: Goal And Purpose Of Creating A New Mesh

This tutorial discusses aspects of mesh creation in DEVEDIT. It begins by discussing the goal of an efficient mesh for device simulation. This tutorial uses the Graphical User Interface (GUI), which is currently not available on Windows.

There are two examples. The first example (Section 2.2: “EXAMPLE 1 - CREATE A NEW STRUCTURE”) demonstrates how to create a structure. The second example (Section 2.3: “EXAMPLE 2 - REMESHING AN EXISTING STRUCTURE”) illustrates how to create a mesh in an existing structure. These two examples illustrate the basic usage of DEVEDIT. Not all features, however, are discussed in this tutorial. We recommend that you read both examples. Finally, in this tutorial, some advanced features (Section 2.4: “Advanced Features”) are mentioned. The remainder of the DEVEDIT manual should be used as a reference.

Specifying a good mesh is a crucial issue in device simulation. There is a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine mesh that resolves the structure in solutions. Numerical efficiency is greater when fewer points are used. The critical areas to resolve are difficult to generalize, since they depend on the technology, transport phenomena, and bias conditions. A generalization is that critical areas tend to coincide with reverse-biased metallurgical junctions. Typical areas that require fine mesh includes:

- high electric fields at the drain/channel junction in MOSFETs.
- the transverse electric field beneath the MOSFET gate.
- recombination effects around the emitter/base junction in BJTs.
- areas of high impact ionization.
- around heterojunctions.

The cpu time required to obtain a solution is typically in proportion to

N^a

where N is the number of nodes and a varies from 2 to 3, depending on the complexity of the problem.

Therefore, it is most efficient to allocate a fine grid only in critical areas and a coarser grid elsewhere. Additional factors to consider in a mesh are:

- avoid obtuse triangles in semiconductor regions, particularly in current path and high field areas.
- avoid abrupt discontinuities in mesh density.
- avoid thin triangles. The ratio of longest to shortest edge in a mesh triangle should be in the order of 10 but not 100.
- use several mesh layers in a material layer, particularly for very thin material or doping layers.
- for most simulations, 2000 - 3000 mesh points are adequate.

These principles can generally ensure accurate solutions with quick convergence times. Poor meshes can lead to inaccurate answers, poor convergence times or even lack of convergence, leaving you without a solution. A significant number of device simulation problems are caused by not adhering to the above principles.

2.2: EXAMPLE 1 - CREATE A NEW STRUCTURE

To start DEVEDIT, enter:

```
devedit &
```

in a terminal window. This starts DEVEDIT in the GUI (opposed to batch) mode. Note the work panel on the right side of DEVEDIT, this home panel returns after other functions have been applied. In this first example, a recessed AlGaAs/InGaAs HEMT is created.

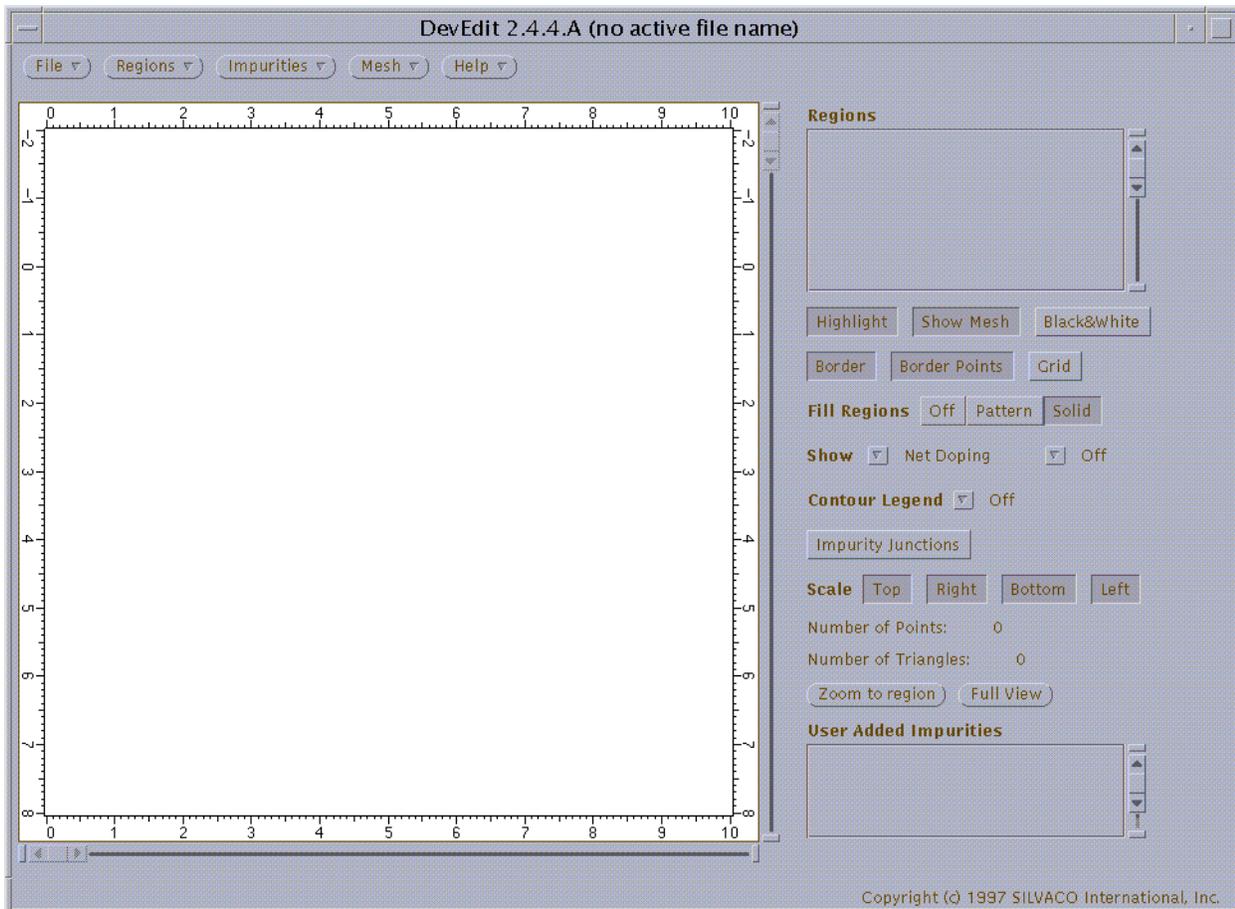


Figure 2-1: DevEdit Graphics User Interface

2.2.1: Work Area

DEVEDIT starts with a work area that uses a default setting. To change the size of this work area, use the **Menu** (right) mouse button over **Regions** and select **Resize Work Area...** A new panel will then appear on the right. You can change the Depth (y) and Length (x) minimum and maximum to desired values. For this tutorial, set $y.min$ to -0.05 , $y.max$ to 0.5 , $x.min$ to zero, and $x.max$ to 4.5 . Then, press Return after typing each value and click on **Apply**. Careful choice of the device in the coordinate system can prevent confusion at later stages.

Note: After entering values into the text boxes in DEVEDIT, press Return for the value to be accepted.



Figure 2-2: Resize Work Area Menu

2.2.2: Defining Regions

Adding a Region

Creating a device begins with adding regions. Each region consists of one material, although an area of a single material can consist of more than one region. Under the **Regions** pull-down menu, choose **Add region....** The menu on the right of DEVEDIT changes with the title **Add Region**. DEVEDIT allows you to create geometrical shapes either by using the mouse or by typing the coordinates of the region.

Using the Mouse

Use the left mouse button and click on the location (0, 0.05), noting that the location of the mouse is displayed in the X and Y locators. Continue to choose the points (4.5, 0.05), (4.5, 0.5), (0, 0.5). If a mistake was made in the locations of the points, press the middle mouse button to remove the point.

Using the Keyboard

Alternatively, type the X and Y locations of each point in the bottom of the right panel, press return after each number, and then click on **Add**. The point will then appear in the work area. You can also change the location of a point by selecting the point in the **Polygon** box (using the left mouse button, so the point is highlighted), changing the X or Y locations by entering the correct location, pressing Return, then clicking on **Replace**.

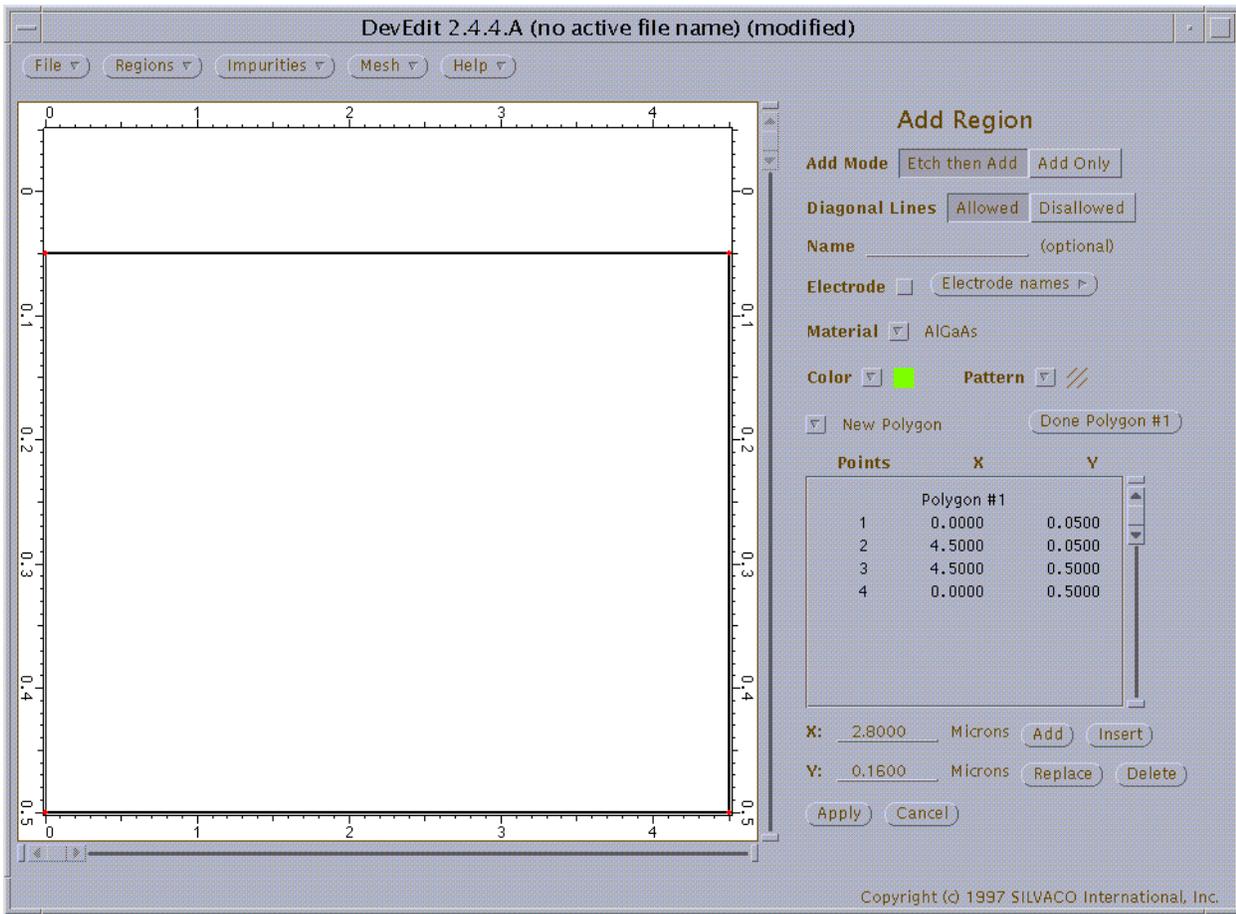


Figure 2-3: Add Region and Constructing a Device

Material Selection and Uniform Doping

After entering the corners of the region, you need to enter the material and doping. Use the right mouse button at the **Material** menu button, choose AlGaAs. Finally using the right mouse button at the **New Polygon**, select **Set Base Impurities**→**Doping Type**→**Generic Donors/Acceptors**, enter 1e14 by **Acceptors**, and press Return (Figure 2-4).

Add Region

Add Mode

Diagonal Lines

Name _____ (optional)

Electrode

Material ▾ AlGaAs

Color ▾ ■ **Pattern** ▾ //

Set Base Impurities

Doping Type ▾ Generic Donors/Acceptors

Acceptors:

Donors:

Net Doping:

Figure 2-4: Adding Uniform Doping Throughout a Region

Setting Mole (Composition) Fraction

Composition fractions for ternary and quaternary materials are defined in Appendix B of the ATLAS USER'S MANUAL. To enter the composition fraction for a material in DEVEDIT, again go to **Set Base Impurities** and under **Doping Type** and select **Composition Fractions**. Enter 0.2 for the **Comp. Fraction X** and press Return (Figure 2-5). The AlGaAs substrate region is now completely defined, with geometry, doping and composition fraction also defined. Once done, click on **Apply**.

Set Base Impurities

Doping Type ▾ Composition Fractions

Comp. Fraction X:

Comp. Fraction Y:

Figure 2-5: Setting the Mole Fraction

Modifying Regions

If you want to change a region or correct a mistake, such as clicking on **Apply** before entering all the settings (including doping and composition fraction), the easiest method to correct the problem is to modify the existing region. When DEVEDIT is displaying the home menu on the right, simply left click on the region name in the upper right box, then right click the **Region** pull-down menu and select **Modify Region**. The right display lists the existing information in a similar format to **Add Region**. When modifications are complete, click on **Apply**.

Next, add the GaAs cap regions. Again under **Regions**, select **Add Region**. Enter a polygon beginning at (0,0). Continue to choose the points (1.1, 0.0), (1.5, 0.05) and (0.0, 0.05). To add doping, select **Generic Donors/Acceptors** and enter $5e21$ by **Donors**.

Enter another polygon in the same manner to create the right cap region, at locations (2.9, 0), (4.5,0), (4.5, 0.05), (2.5, 0.05), GaAs, with the same doping.

Etch then Adding a Region

Next, an InGaAs layer is added. This step is performed in a similar way to adding a region, except the location is being etched from existing material (the AlGaAs substrate). The default setting for **Add Mode** is **Etch then Add**, meaning the new region displaces, or overwrites, any existing material region. The other mode, **Add Only**, adds a new region only where no other region has been defined. Add an InGaAs region from (0, 0.084), (4.5, 0.084), (4.5, 0.098), and (0, 0.098). The precision of these coordinates exceeds the default of the mouse setting. You have two options, either enter the coordinates of each point and click **Add** after each point, or use the mouse first to select approximate locations, then select the points in the **Polygon** box, and edit the points (click **Replace** after each point). Set the doping to $5e15$ donor concentration.

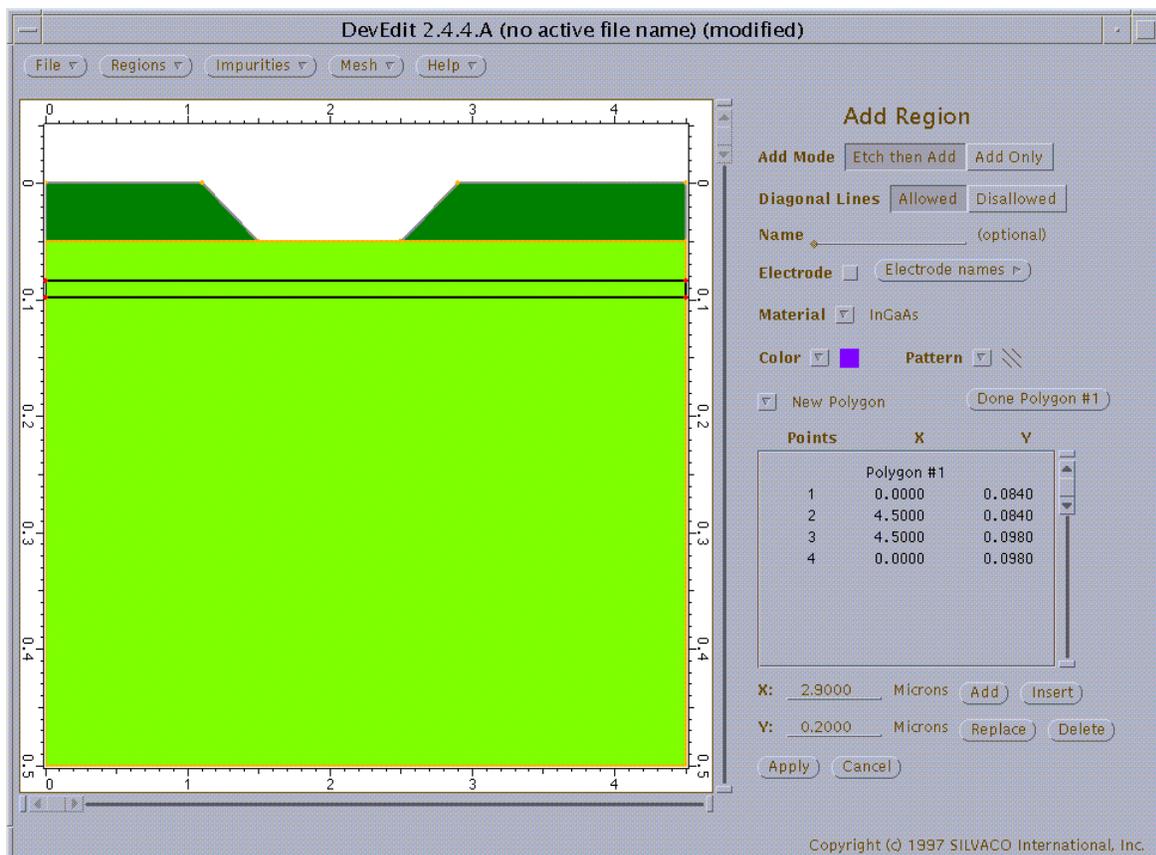


Figure 2-6: Etching, then Adding a Region - Overwriting a Material in an Existing Area

Add Electrodes

The last regions to add are the electrodes. In ATLAS, only the boundary of an electrode contacting the semiconductor is considered. Therefore, the height of the electrode and the mesh inside of the electrode are irrelevant. Add a gold region at (1.7, 0), (2.3, 0), (2.3, 0.05), (1.7, 0.05). To identify this region as an electrode, click on the button next to **Electrode**, then use the right mouse button to click **Electrode names**, and choose **gate**.

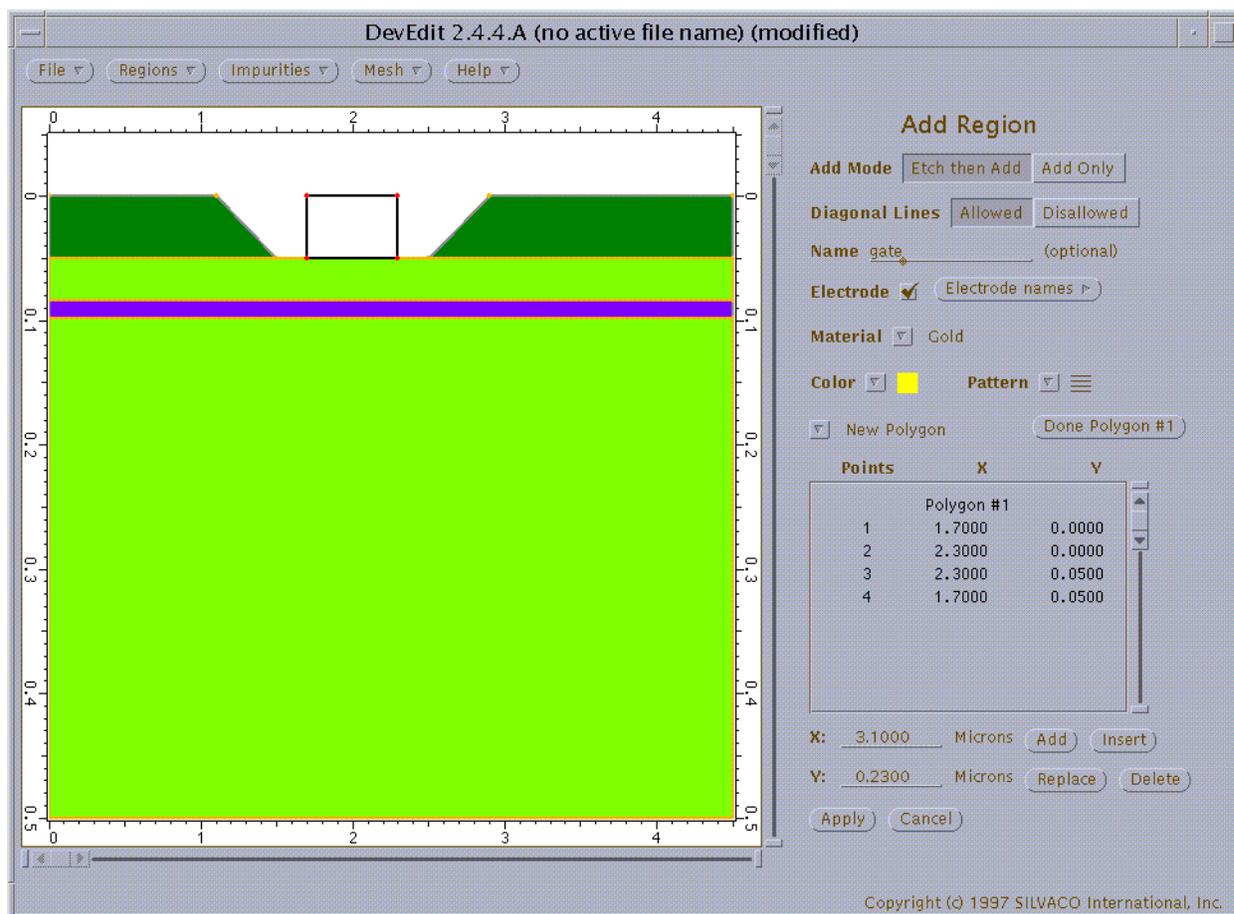


Figure 2-7: Adding a Gate Electrode

In a similar manner, add a source contact at (0, 0), (0, -0.01), (1, -0.01), and (1, 0). Add a drain contact at (3.5, 0), (3.5, -0.01), (4.5, -0.01), and (4.5, 0). Much of the geometry of the device is now complete. The last step is to add impurities.

2.2.3: Defining Impurities

Adding Impurities

If an impurity concentration is constant through a region, the preferred manner to define the dopant is in the **Region** definition as discussed above. For non-uniform impurities, include impurities that are in more than one region, then define the impurities separately as described below.

Under **Impurities**, select **Add impurity** and the menu for impurities will become available. Next to **Impurity**, select **Donors**. For **Start X** and **Y** values, use (0,0) and (1, 0.08). Note that only two points are selected, not four, and the opposite corners of a rectangle should be selected. A one-dimensional line is acceptable; it is considered a flat rectangle. This region is the impurity source region, and the impurity concentration is the peak concentration, uniformly throughout the region. Set the **Peak Concentration** to $5e+21$ and the **Reference Value** to $1e+20$. The **Reference Value** is used as a scaling factor in the Roll-Off Functions (see Section 3.8: "ROLL-OFF FUNCTION").

The **Roll-Off** functions calculate the vertical and horizontal roll-off from the source impurity region (the peak concentration) as a function of x , y , and in three dimensional structures, z . The **Join Function** calculates the dopant concentration when both an x and y distance exists. Set the **Join Function** to **Multiply**. Select the **Y Roll-off** as **Gaussian (Dist)** and **Distance** as 0.048. Select the **X Rolloff** as **Error Function** and **Constant** as 0.02. Then, click on **Apply**. Remember to press Return after entering the numbers. See Figure 2-8.

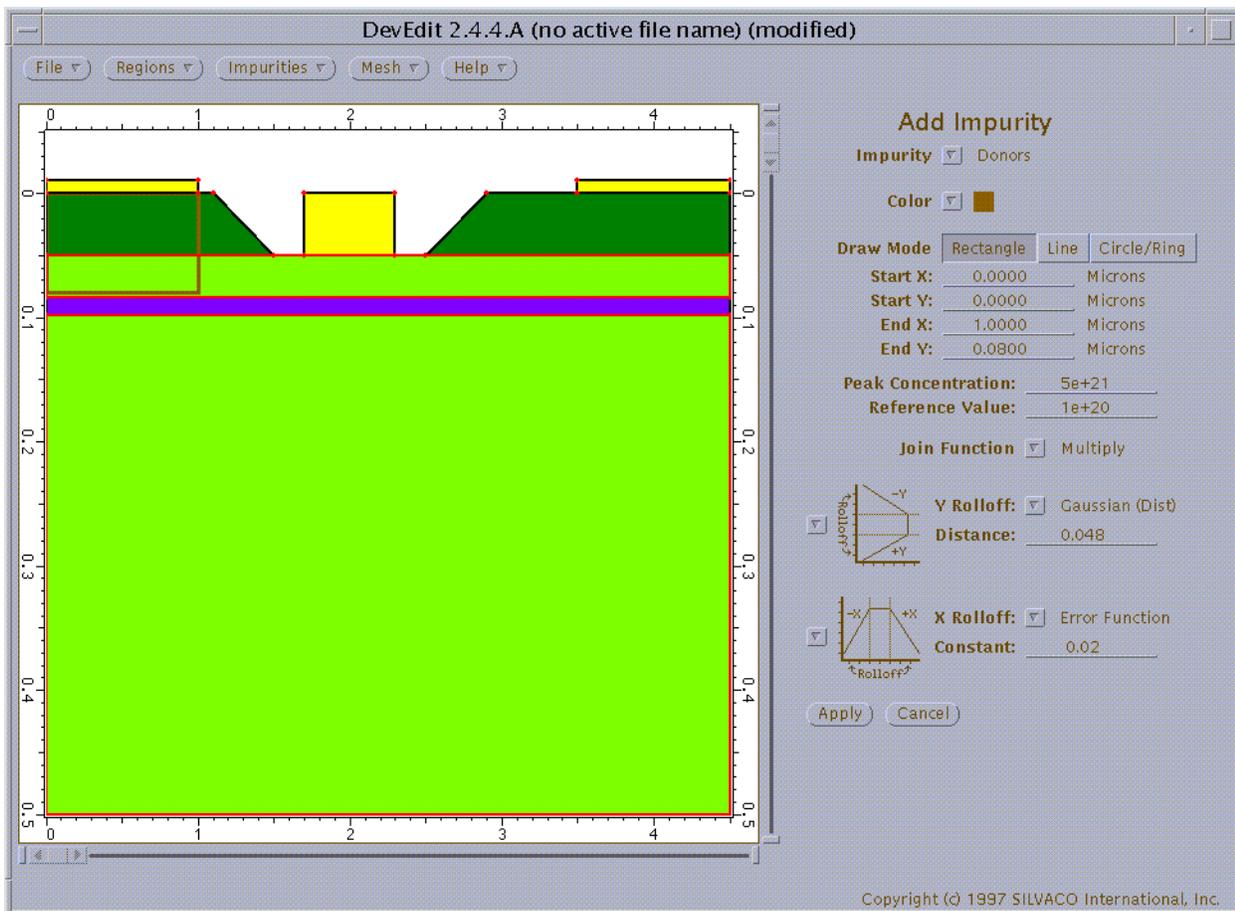


Figure 2-8: Add Impurity

Add another impurity region from (3.5, 0) to (4.5, 0.08) with similar roll-off properties.

Displaying the Doping

Back on the main menu, there are options to display, or **Show**, attributes of the structure. The first button defaults to **Net Doping**. The second button defaults to **Off**. Select the second button to **Fine**. This option shows a relatively fine gradation of doping throughout the structure. **Coarse**, **Medium**, and **Very Fine** display options also exist, but be alert that the finer the display, the longer the refresh time takes. Options under **Net Doping** includes separate options for donors, acceptors, and specific impurities. You can also select the location of the **Contour Legend**.

Modify Impurities

The impurity regions added are listed in the main menu at the bottom under **User Added Impurities**. Select the impurity of interest (left mouse button), then under the **Impurities** button, select **Modify impurity...** A new menu, similar to the **Add Impurity** menu, will become available. Using these principles and techniques, you can design any device of interest. The next concern is creating a mesh for the device.

2.2.4: Mesh Creation

MeshBuild

The **Mesh** pull-down menu has a **MeshBuild** command. This command reads the current boundary conditioning conditions, the base mesh, the geometry and impurities, and mesh constraints, and creates a mesh, using the mesh parameters available. The default parameters create a mesh adequate to define the geometry of the device, but little else. It is unsatisfactory to describe the details of the impurity distribution, for instance, or the material layers for device simulation in ATLAS.

Mesh Parameters

Under the **Mesh** pull-down menu, there is a command of **Mesh Parameters**. **Base Mesh Height** and **Width** may be used to create more symmetric meshes for some devices. For most devices, setting **Mesh Constraints** provide the same results and also allow more fine tuning. Therefore, these parameters were not used in this example. In devices with extremely detailed region borders, it may be turned **Off**. Click on **Cancel**. See Section 3.5.1: “Boundary Conditioning” for more information.

Refine on Quantities

This command allows you to refine the mesh on gradients of various quantities, including donors, acceptors, total and net doping, and molar composition. If the structure was imported from ATLAS, other quantities, including electrical field and potential, are available. This is a useful tool for refining the grid on areas that require a finer grid, namely, where gradients exist.

Right click on the **Add** button and select **Donors**. Then, right click on the **Mesh** pull-down menu and select **MeshBuild** again. **MeshBuild** operates by building a mesh with instructions of mesh generation, which have been modified when the **Refine on Quantities** was changed to include donor gradients.

Note: The mesh has changed to include a finer mesh where donor concentration gradients exist.

Click on **Done** at the bottom of the panel. This action returns you to the main DEVEDIT panel. If the existing panel is not the main panel, click on **Cancel** or **Done** on the existing panel. DEVEDIT returns you to the previous panel, and ultimately to the main panel. Near the bottom of the main panel the **Number of Points** and **Number of Triangles** are listed. These numbers are useful in order to gauge the total number of points with the more subjective interpretation of the quality of the grid.

You can go back to **Refine on Quantities**. The **Scale** defaults to **Logarithmic**, which is appropriate for dopant concentrations. Change the **Sensitivity**, then rebuild the mesh (**MeshBuild**), and observe the effect on the total number of points. The **Sensitivity** setting controls the extent of the gradient in which mesh points are added. A higher setting reduces the density of the mesh. A lower setting increases the mesh density. The **Transition** value is the minimum value that is considered for Meshbuilding purposes. For example, a gradient donor concentration from 10^9 to 10^8 is not considered when the **Transition** value is set to 10^{10} (the default).



Figure 2-9: Mesh Refinement on the Donor Gradient

Mesh Constraints

The **Mesh Constraints** Menu is the principal area for controlling mesh construction. In this section, you can control the maximum triangle ratio, the maximum and minimum height and width, either throughout the device, or selectively in given regions, in material types, or underneath regions or materials. Begin with selecting **Semiconductor Regions** in the **Material Types and Regions** box. Click on the box next to **Max. Height**, and set the value (either by typing or moving the slide bar) to 0.05 (all units of distance are microns). Similarly, set the **Max. Width** to 0.25. These values are valid for the mesh creation in all semiconductor regions. As mentioned previously, the mesh in insulators and electrodes are unimportant. Therefore, it is acceptable to leave the mesh arbitrarily large in these regions. By clicking on **MeshBuild** again, you will notice the mesh in the semiconductor regions now adheres to these criteria.

Select the upper AlGaAs region. Again, the same mesh criteria are available for control for this single layer. Change the **Max. Height** to 0.01. We strongly recommend that each material layer, or region, have several (such as four) mesh layers. You can also adjust parameters for other layers. It is recommended in this example to increase the mesh density in the semiconductor layers so that several mesh layers exist in each region.

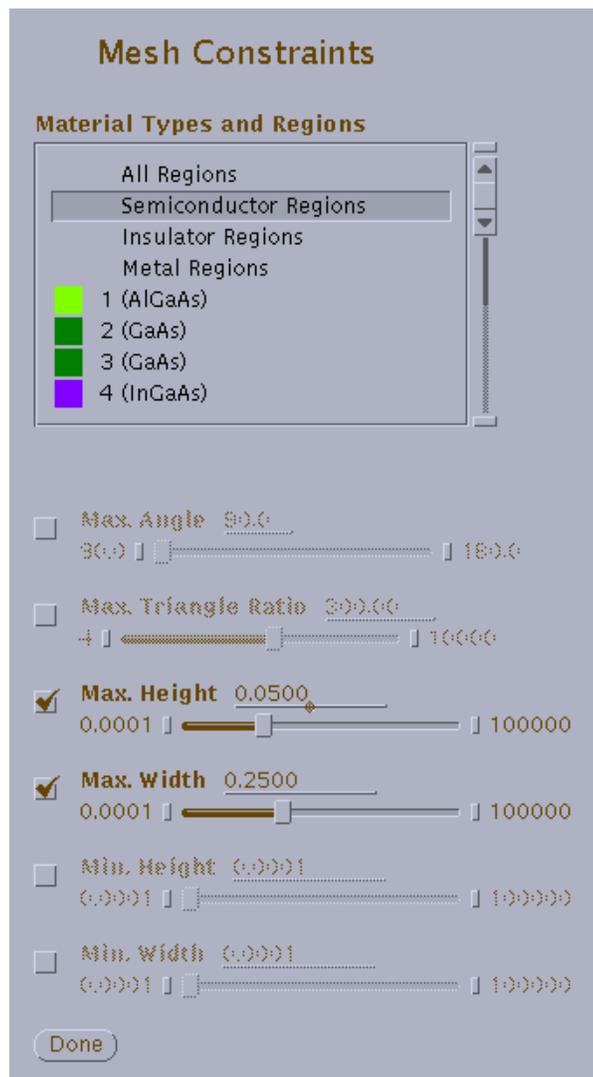


Figure 2-10: Mesh Constraints Menu

Manual Refine Box

The Refine Box (**Mesh**→**Refine Box**) has the **Refine X**, **Refine Y**, **Refine Both**, and **Unrefine** options. This meshing does not work within **MeshBuild**. Instead, it doubles the density of the mesh, in the area specified, in the X, Y, or both directions, or reduces the mesh density by half. But because these mesh changes are not easily reproduced, this action is not recommended.

2.2.5: Saving The File

You can, and should, save the file in two formats: the structure file and the command file. The structure file is a format used by other SILVACO programs, including TONYPLOT and ATLAS. Therefore, it is necessary to save the structure file for continued device simulation. The command file is a list of the instructions used by DEVEDIT to create the structure and the mesh. To make additional changes at a later time, save a command file so that the original set of DEVEDIT instructions can be read. Additionally, running DEVEDIT in batch mode requires the command file.

Structure File

To save a structure file, right-click **File**→**Save as...** and a pop-up menu will appear (Figure 2-11). By convention, SILVACO structure files end with the extension `.str`. Choose a name and use this extension (for instance, `example1.str`). Then, press the **Save Structure** button. You can now use the file in ATLAS and TONYPLOT.

Command File

To save a command file, also right-click **File**→**Save as...** and enter a name ending with the extension `.de` (for DEVEDIT). It is good practice to use the same filename as the structure file, but ending with the conventional `.de` extension. Therefore, in this case it would be `example1.de`.

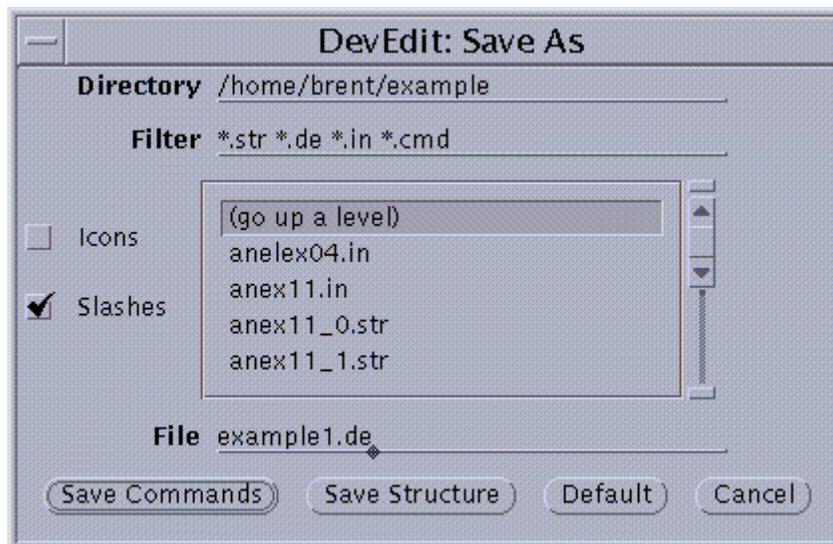


Figure 2-11: File Save Menu

Batch Mode

You can also run DEVEDIT in batch mode, within DECKBUILD, without using the graphics user interface (GUI) mode used above. Although you can directly input the command statements to create a structure and mesh, this method is more difficult than using DEVEDIT in GUI mode. Once the command file has been saved, however, it can be easily loaded, edited, and runned in batch mode within DECKBUILD.

Start DECKBUILD with `deckbuild &`, left click on **File**→**Open**, and select a DEVEDIT command file (example1.de). This command file can be run inside DECKBUILD with minor modifications.

The first line of the DEVEDIT is similar to `DevEdit version=2.4.0.R` in which it is recommended that this line become a comment, beginning with `#`. Then, you should add a new first line that reads `go devedit`, or if you want to retain the version number `"go devedit simflags="-V 2.4.0.R"` (or whatever the version number of DEVEDIT in use).

```

go devedit simflags="-V 2.4.4.A"
#DevEdit version=2.4.4.A

work.area x1=0 y1=-0.05 x2=4.5 y2=0.5
# devedit 2.4.4.A (Tue Dec 30 20:55:13 PST 1997)
# libsfilm 2.0.0.R (Thu May 1 18:03:38 PDT 1997)
# libDW_Misc 1.20.0.R (Mon Apr 28 17:55:25 PDT 1997)
# libCardDeck 1.20.0.R (Tue Apr 29 15:01:54 PDT 1997)
# libGeometry 1.20.0.R (Mon Apr 28 18:17:55 PDT 1997)
# libDW_Set 1.20.0.R (Mon Apr 28 17:57:52 PDT 1997)
# libSVC_Misc 1.20.0.R (Mon Apr 28 18:20:53 PDT 1997)
# libSDB 1.0.6.C (Tue Dec 30 14:47:45 PST 1997)
# libSSS 1.20.0.R (Mon May 5 16:29:45 PDT 1997)
# libMeshBuild 1.20.4.A (Tue Dec 30 20:36:26 PST 1997)
# libDW_Make 1.1.3.R (Tue Dec 2 01:59:58 PST 1997)
region reg=1 mat=AlGaAs color=0x7fff00 pattern=0x9 \
  polygon="0,0,0.084 0,0.05 1.5,0.05 1.7,0.05 2.3,0.05 2.5,0.05 4.5,0.05
4.5,0.084" \
  polygon="4.5,0.098 4.5,0.5 0,0.5 0,0.098"
#
impurity id=1 region.id=1 imp=Acceptors \
  peak.value=1000000000000000 ref.value=1000000000000000 comb.func=Multiply
#
constr.mesh region=1 default

region reg=2 mat=GaAs color=0x7f00 pattern=0x9 \
  polygon="0,0 1,0 1.1,0 1.5,0.05 0,0.05"
#
impurity id=1 region.id=2 imp=Donors \

```

next line stop cont run quit Line: 1
paste init pause clear restart kill Stop: None

```

libDW_Make 1.1.3.R (Tue Dec 2 01:59:58 PST 1997)

MeshBuild Library based on MeshBuild v1.9.0
Copyright (c) 1991 Integrated Systems Laboratory, ETH Zurich, Switzerland

Executing on host: kokanee Tue Aug 18 17:12:14 1998

DevEdit>

```

DEVEDIT started DEVEDIT

Figure 2-12:Running DevEdit in Batch Mode within DeckBuild

We also recommend that you add the line `structure outfile=test.str` at the end of the file to explicitly save the structure in a structure file.

2.3: EXAMPLE 2 - REMESHING AN EXISTING STRUCTURE

2.3.1: Loading The Structure

Typically, a structure is created in ATHENA and saved in a structure file. For this tutorial, an existing SILVACO example will be used.

2.3.2: Obtain Existing Structure

Start DECKBUILD, by entering:

```
deckbuild &
```

in a UNIX terminal window. Right-click on **Main Control** and select **Examples...** The DECKBUILD: EXAMPLES library of examples will appear in a separate pop-up menu. Double-click on MOS1, then double-click again on mos1ex01.in. Then, click on **Load example**. This loads the ATHENA/ATLAS input deck into DECKBUILD, which copies the structure and log files into the directory where you launched DECKBUILD.

In DEVEDIT, left click on **File** and select **Load**. Then, highlight (again with the mouse) mos1ex01_0.str and click on **Load File**. The ATHENA created structure is now in DEVEDIT. DEVEDIT command files (*.de) are loaded in the same manner.

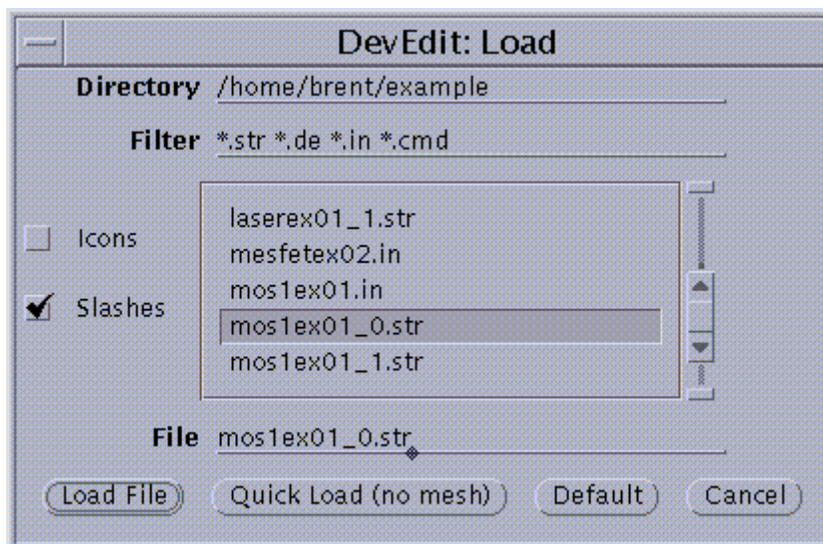


Figure 2-13: DevEdit Load File Menu

2.3.3: Structure Editing

We recommend that if you created the structure in ATHENA, do not make any changes to the structure (i.e., material boundaries, doping distribution). If you created the structure in DEVEDIT, then you should edit the command file (*.de) but not the structure file. Once you finish the edits to a DEVEDIT command file, you should save both a command file and a structure file.

For this example and this purpose, only change the mesh. Minor changes to the material boundaries will be done in boundary conditioning as described in “Boundary Conditioning” on page 2-16.

2.3.4: Display

Zoom

You can depress the left mouse button anywhere on the displayed structure, and drag it to another location, defining a rectangle between the two points. Once you release the mouse button, only the selected area will be displayed (zoom). The bottom and right axes will have sliders for panning. The **Full View** button on the right menu will return the display to the entire work area.

Displaying the Doping

Also on the main menu, there are options to display, or show, attributes of the structure. The first button defaults to **Net Doping**, and the second button defaults to **Off**. Select the second button to **Fine**. This option shows a relatively fine gradation of doping throughout the structure. **Coarse**, **Medium**, and **Very Fine** display options also exist. Be aware that the finer the display, the longer the refresh time takes. **Options** under **Net Doping** include separate options for donors, acceptors, and specific impurities. One can also select the location of the **Contour Legend**, by using the right mouse button adjacent to **Contour Legend**.

Impurity Junctions

This toggle switch will display the p-n junctions in a device.

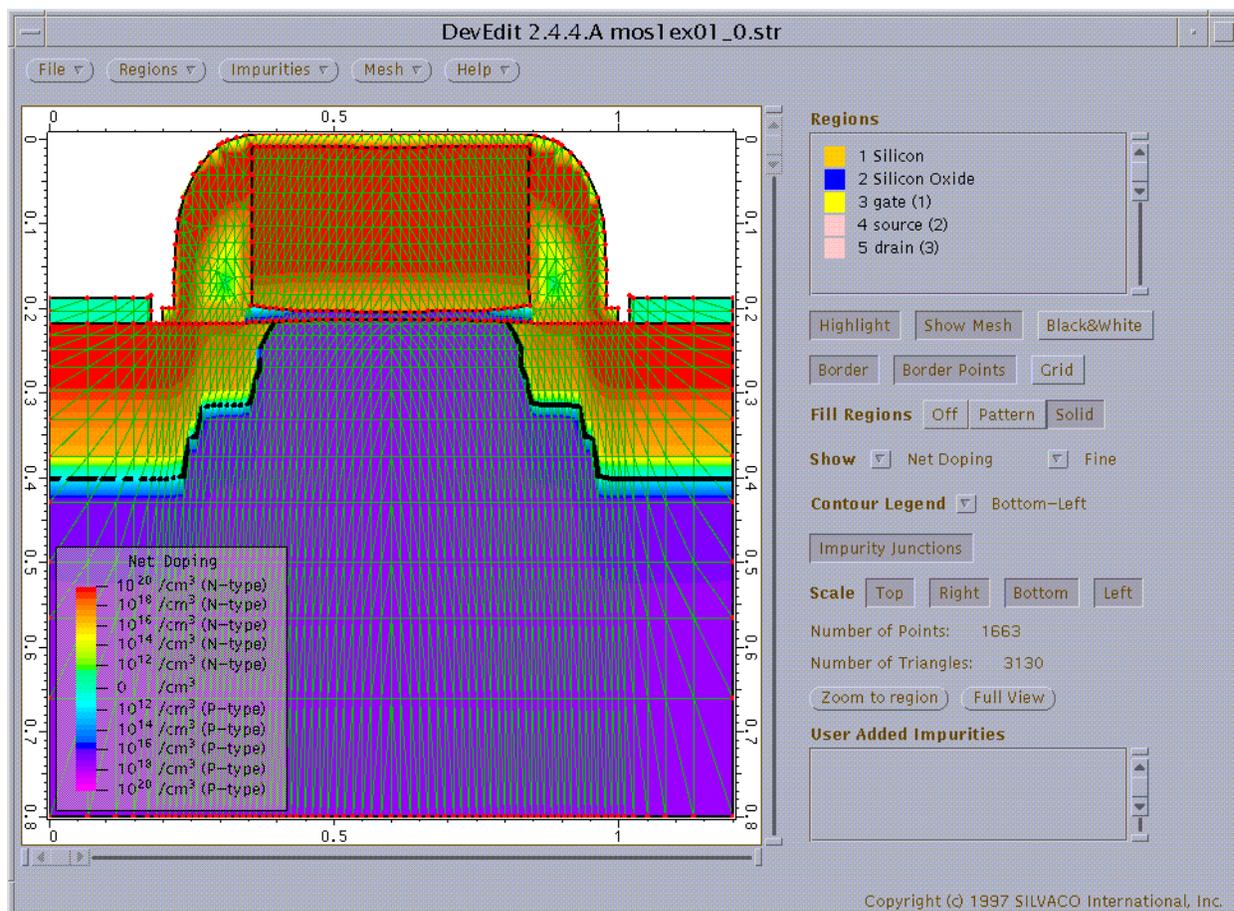


Figure 2-14: Displaying the Net Doping Distribution in DevEdit

2.3.5: Mesh Creation

Boundary Conditioning

There is one important issue that must be considered when creating a new mesh in an existing structure. The first step that you should perform before creating a new mesh is boundary conditioning.

Material boundaries are defined by border points. Necessarily, these border points are also points that define mesh locations. You can make slight modifications to the structure to minimize the number of border points. This process of eliminating points that are not critical is called boundary conditioning. Examples of such unnecessary points include points along a straight line not contributing to the geometry of the structure and colinear points (or nearly colinear, such as within one degree).

Right click on **Mesh** and select **Mesh Parameters**. Both the **Mesh Parameters** and the **Boundary Conditioning** sections are displayed. By default, **Boundary Conditioning** is set to **Automatic**. Note the border points in the structure, denoted as red dots, particularly along the oxide material boundary. If no boundary conditioning were performed, these points would remain and require mesh points at these locations. Click on **Apply** underneath **Boundary Conditioning**. The mesh is removed but many border points are removed also.

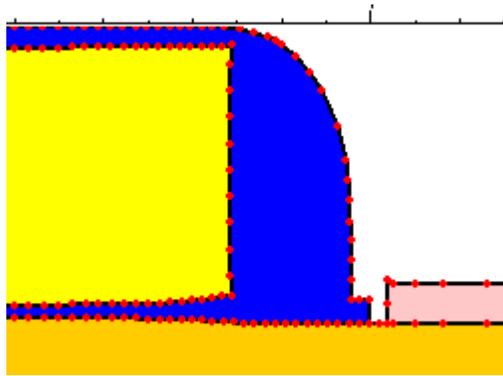


Figure 2-15: Border Points Defining Material Boundaries Before Boundary Conditioning

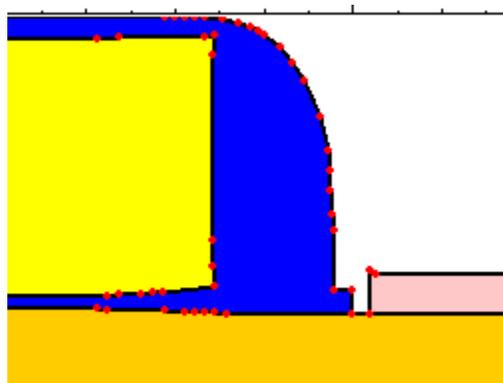


Figure 2-16: Border Points Defining Material Boundaries After Boundary Conditioning

The **Max. Line Slope** is a ratio (greater than one) between the length (X) and height (Y) of mesh triangles along a material boundary. If a mesh triangle has a line slope greater than this value, the triangle is subdivided, so that the original material boundary line is divided into two lines. One line is horizontal (or vertical), and the other line is with an angle set by the maximum line slope.

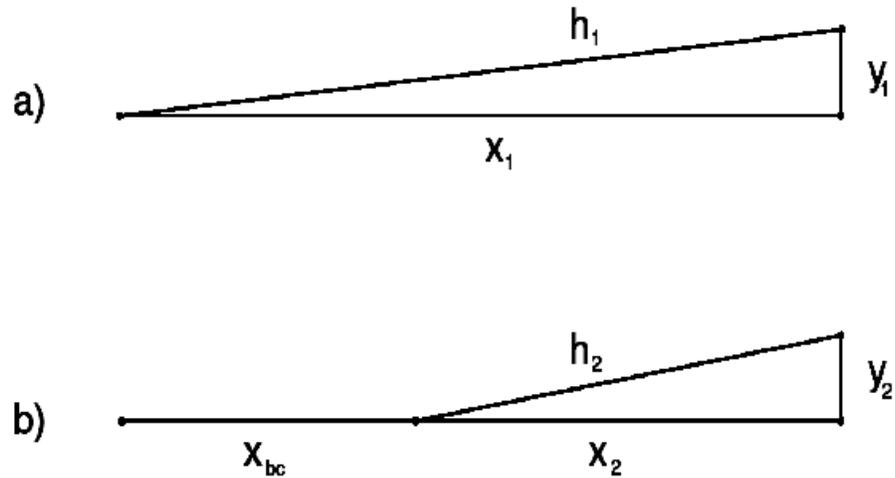


Figure 2-17: Boundary Conditioning with Respect to Maximum Line Slope

Note: The Maximum Triangle Ratio should be larger than the Maximum Line Slope.

The **Rounding Unit** is a distance to which all boundary points are rounded to an even multiple of this value.

The **Line Straightening** value is an angle. If two boundary segments have a joining angle equal to or greater than $(180 - \text{line.straightening})$, the two line segments are combined by removing the joining point.

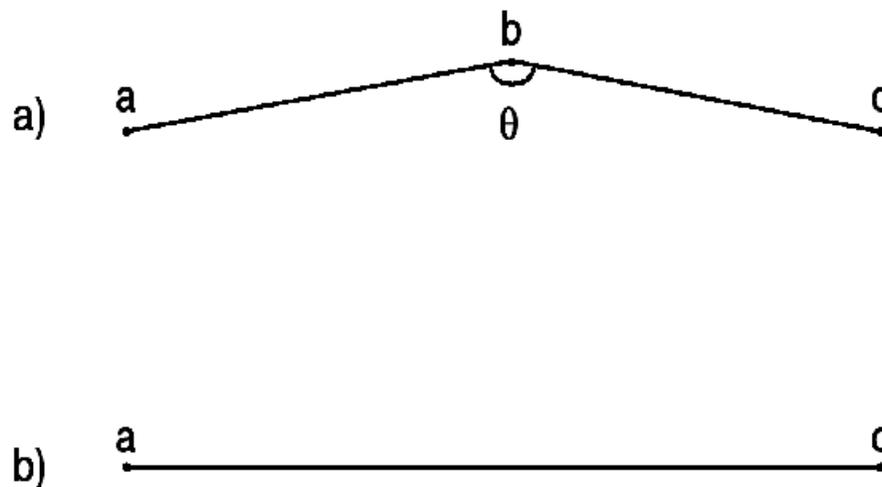


Figure 2-18: Line Straightening

In general, the default settings for boundary conditioning are satisfactory for most structures. After boundary conditioning (i.e., clicking **Apply**), you can proceed to **Refine** on **Quantities and Mesh Constraints**.

Mesh Parameters

Under the **Mesh** pull-down menu, there is an option called **Mesh Parameters** along with **Boundary Conditioning**. **Base Mesh Height** and **Width** can be used to create a fairly uniform underlying mesh. By making these smaller, it may help create more symmetric meshes for some devices. For most devices, setting **Mesh Constraints** provide the same results and also allow for more fine tuning. Therefore, these parameters were not used in this example.

Refine on Quantities

This action allows you to refine the mesh on gradients of various quantities, including donors, acceptors, total and net doping, and molar composition (if the structure was imported from ATLAS, other quantities, including electrical field and potential are available). This is a useful tool for refining the grid on areas that require a finer grid, namely, where gradients exist. This tool is very useful, particularly near p-n junctions.

To refine the mesh, right click on **Mesh**→**Refine on Quantities...** and right click on the **Add**→**Net Doping**. Then, right click on **Mesh**→**MeshBuild** (or left click on **Mesh**, which selects **MeshBuild** by default). **MeshBuild** operates by building a mesh with instructions of mesh generation, which have been modified when the **Refine on Quantities** was changed to include gradients in the net doping. The mesh has changed to include a finer mesh where gradients of the net doping exist.



Figure 2-19: Refine on Quantities Menu for Mesh Refinement on Net Doping Gradient

Click on **Done** at the bottom of the panel. This action returns to the main DEVEDIT panel. If the existing panel is not the main panel, click on **Cancel** or **Done** on the exiting panel. DEVEDIT returns you to the previous panel and ultimately to the main panel. Near the bottom of the main panel, the **Number of Points** and **Number of Triangles** are listed. These numbers are useful to gauge the total number of points with the more subjective interpretation of the quality of the grid.

Return to **Refine on Quantities**. The **Scale** default is **Logarithmic**, which is appropriate for doping concentrations. If the value of the quantity varies more than the sensitivity parameter, the mesh is made more dense locally. Accordingly, decreasing the value of the sensitivity increases the number of mesh points. You can change the sensitivity value slightly, then select **Mesh**→**MeshBuild** and see the effect on the mesh.

The **Transition** value is the minimum value of the quantity that is considered significant. If the **Transition** value is set to 10e10, then gradients of **Net Doping** between 10e9 and 10e8 are not considered when the mesh is created.

Mesh Constraints

The **Mesh Constraints** section is the principal method for controlling mesh construction. In this section, you can control the maximum triangle ratio, the maximum and minimum height and width, either throughout the device, or selectively in given regions, in material types, or underneath regions or materials.

It is important to avoid obtuse triangles in the semiconductor, however, obtuse triangles in the poly gate, oxide and metal regions are acceptable. A logical method would be to allow obtuse triangles in all regions, and then override the semiconductor regions to only include acute and right triangles.

Begin with selecting **All Regions** in the **Material Types and Regions** box. For **Max. Angle**, either enter or move the slide bar to 150. Next, select **Semiconductor Regions** and click on the box next to **Max. Angle** and set the value to 90. After re-building the **Mesh**, you can observe that the number of mesh points have decreased. This has been done without adding obtuse triangles to the semiconductor, yet relaxing the mesh in areas that are not of interest.

Further constraints can be imposed in the semiconductor regions. Set the **Max. Height** to 0.1 (all units of distance are microns). Similarly, set the **Max. Width** to 0.1. These values are valid for the mesh creation in all semiconductor regions. As mentioned previously, the mesh in insulators and electrodes are unimportant. Therefore, it is acceptable to leave the mesh arbitrarily large in these regions. By clicking on **Meshbuild** again, you will notice that the mesh in the semiconductor now adheres to these criteria.

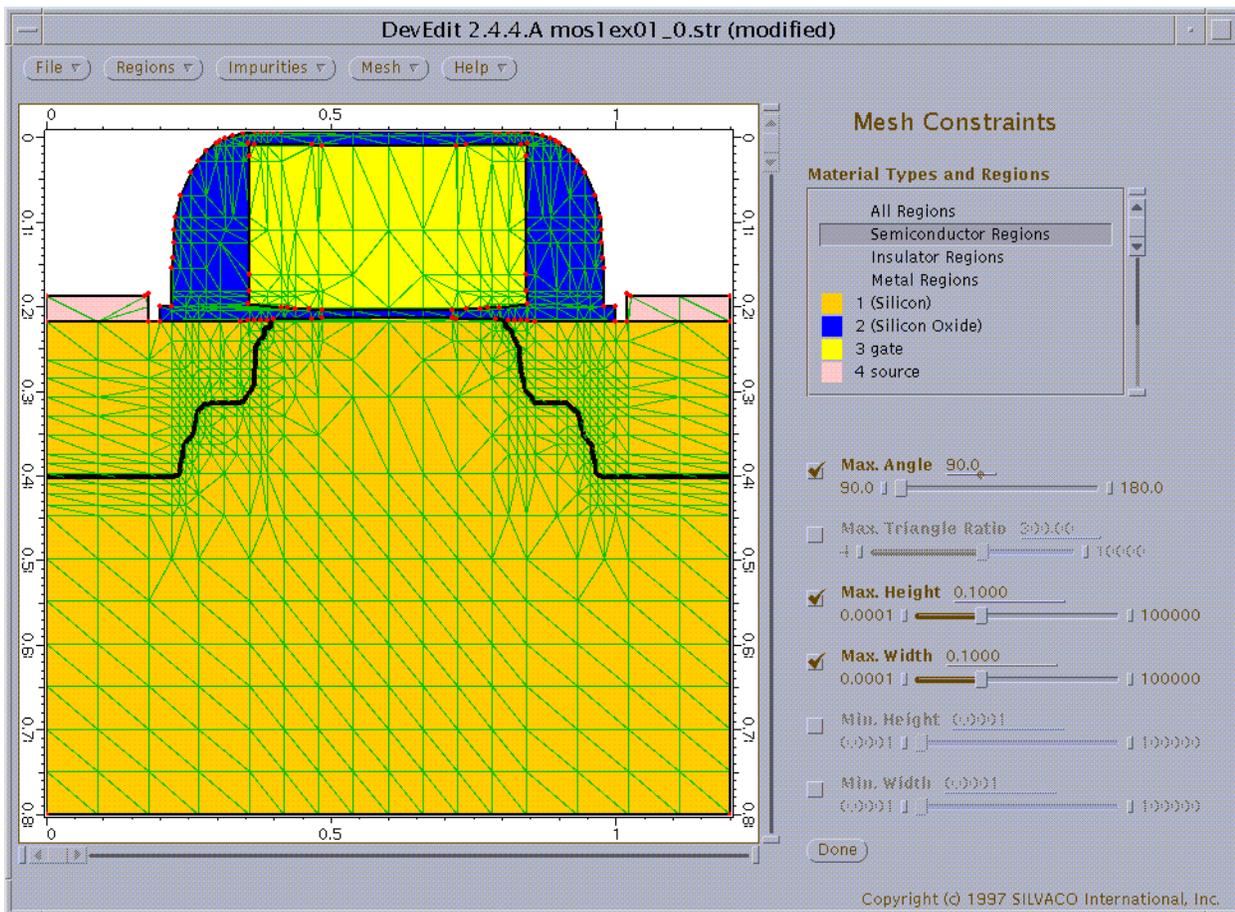


Figure 2-20: Mesh after Refining on Constraints in Semiconductor

You can perform further refinement in more specific areas. Scroll down in the **Material Types and Regions** box. In addition to the general areas (all regions, semiconductors, insulators and metal regions) and the specific regions (in this case, silicon, oxide, gate (poly), source and drain), there are fixed box constraints under **region constraints** and under **material constraints**. These latter options can be very useful for MOS devices, since the channel area is a critical area for a dense mesh.

Select **Add New Under Region Constraints**. Then, click on the **Location** button (if not already depressed), which is adjacent to the **Constraints**. Next to **Under Region**, select (with right mouse button) **gate**. Next to **In Material Type**, select **Semiconductor**. Set **Depth** to 0.1. This procedure has selected the semiconductor underneath the gate, to a depth of 0.1 microns from the silicon surface, as the area for mesh refinement. Next, depress **Constraints**, which displays the same **Mesh Constraints** menu seen previously. Set **Max. Height** to 0.02, and **Max. Width** to 0.04. Click on **Apply**, and click on **MeshBuild**.

The structure now has a mesh that is better suited for device simulation. The original mesh was created primarily for process simulation, which has different requirements, including dense mesh near implant ranges, silicon - oxide interfaces, along diffusion regions, and oxidation areas. For device simulation, the priorities include non-obtuse triangles in the semiconductor, p-n junction boundaries, and areas of high electric field and carrier mobility (such as the channel region in a MOSFET), and do not include oxide shape, electrode shape, and regions far from electrical current.

Note: See S-PISCES chapter of the ATLAS User's Manual for details on typical mesh size required for MOSFETs.

2.3.6: Saving The File

You can save and, should, save the file in two formats: the structure file and the command file. The structure file is a format used by other SILVACO programs, including TONYPLOT and ATLAS. Therefore, you need to save the structure file for continued device simulation. The command file is a list of the instructions used by DEVEDIT to create the mesh (and the structure, if it was made in DEVEDIT, as in Example 1). If you want to make additional changes at a later time, you need to save a command file so that the original set of DEVEDIT instructions can be read. Additionally, running DEVEDIT in batch mode requires the command file.

Structure File

To save a structure file, right-click **File**→**Save as...** and a pop-up menu will appear (Figure 2-11). By convention, SILVACO structure files end with the extension `.str`. Choose a name and use this extension (for instance, `example2.str`). Then, press the **Save Structure** button. You can now use the file in ATLAS and TONYPLOT.

Command File

To save a command file, also right-click **File**→**Save as...** and enter a name ending with the extension `.de` (for DEVEDIT). It is good practice to use the same filename as the structure file, but ending with the conventional `.de` extension. Therefore, in this case it would be `example2.de`.

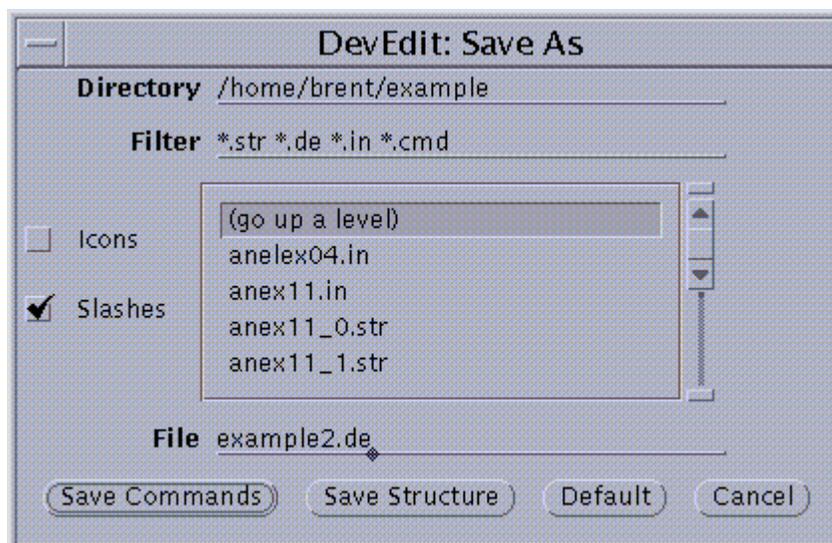


Figure 2-21: DevEdit File Saving Menu

Batch Mode

The goal of DEVEDIT in batch mode for this case is to automate DEVEDIT's mesh creation algorithm, using user defined parameters without the steps of manual regridding. Therefore, you can make minor changes of the process simulation without requiring you to complete the steps of DEVEDIT in GUI mode, since the criteria used in DEVEDIT (such as mesh constraints in material layers, refined grid along p-n junctions, loose grid deep in the substrat) are general and would be valid for similar devices of minor process changes. Therefore, this purpose for batch mode is different than in Example 1.

In this example, the initial structure was created in ATHENA. Device simulation would be performed in ATLAS. This example demonstrates how to incorporate the DEVEDIT re-mesh syntax within the same input deck between the ATHENA and ATLAS sections.

You can start any text editor or use DECKBUILD for editing.

To start DECKBUILD, enter `deckbuild &`, left click on **File**→**Open**, and select a DEVEDIT command file (`example2.de`).

The first line of the DEVEDIT is similar to `DevEdit version=2.4.0.R`, in which it is recommended that this line become a comment, beginning with `#`. Then, add as a new first line to read `go devedit`, or if you want to retain the version number, `go devedit simflags="-V 2.4.0.R"` (or whatever the version number of DEVEDIT in use).

Because the structure was initially created in ATHENA, the DEVEDIT commands to redefine the structure boundaries is redundant and should be deleted. The relevant section begins with **#Set Meshing Parameters**. Therefore, all intermediate lines previous should be removed.

This truncated DEVEDIT batch file should be moved or copied into the ATHENA/ATLAS input file, beginning after the ATHENA syntax and before the ATLAS syntax (See Figure 2-22).

The last line of the ATHENA section is `tonyplot mos1ex01_0.str -set mos1ex01_0.set`. ATLAS begins with `go atlas` at the bottom of the DECKBUILD main window.

```

structure outfile=mos1ex01_0.str

# plot the structure
tonyplot mos1ex01_0.str -set mos1ex01_0.set

go devedit
#DevEdit version=2.4.0.R

# Set Meshing Parameters
#
base.mesh height=10 width=10
#
bound.cond !apply max.slope=30 max.ratio=100 rnd.unit=0.001 line.straightening=1
align.points when=automatic
#
imp.refine imp="Net Doping" scale=log transition=1e+10
imp.refine min.spacing=0.02
#
constr.mesh max.angle=150 max.ratio=300 max.height=10000 \
max.width=10000 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default max.angle=90 max.height=0.1 \
max.width=0.1
#
constr.mesh type=Metal default
#
constr.mesh region=1 default
#
constr.mesh region=2 default
#
constr.mesh region=3 default
#
Mesh Mode=MeshBuild

base.mesh height=10 width=10

bound.cond !apply max.slope=30 max.ratio=100 rnd.unit=0.001 line.straightening=1
align.Points when=automatic

structure outfile=example2.str

##### Vt Test : Returns Vt, Beta and Theta #####
go atlas

# set material models
models cvt srh print

```

next line stop cont run quit Line: 1
paste init pause clear restart kill Stop: None

It is now Tue Oct 6 14:30:02 1998
Athena 4.3.0.R is executing on "kokanee"

Loading model file 'athenamod'... done.
ATHENA>

ATHENA started ATHENA

Figure 2-22:Running DevEdit in Batch Mode within DeckBuild after ATHENA Process Simulation

We also recommend that you add the line `structure outfile=example2.str` at the end of the DEVEDIT section of the input file to explicitly save the structure in a structure file. Any filename can be used with the convention of the `.str` extension used for structure files.

With this procedure, you can run the entire deck and incorporate the mesh re-creation from DEVEDIT. Any minor changes in the ATHENA process simulation file does not require changes to the DEVEDIT command file.

An alternative method of using DEVEDIT in batch mode is to use the command file as a distinct file and not incorporate it within ATHENA-ATLAS input file(s). As already mentioned, you must edit the file to remove the structure definition (up to the # Set Meshing Parameters line). DEVEDIT, however, must initialize from the original structure. This can be accomplished by including an initialize command immediately after commencing DEVEDIT. Therefore:

```
go devedit init infile=moslex01_0.str
```

then continue with the command file. The ATLAS file would then begin with

```
go atlas mesh infile=example2.str
```

or whatever name of the structure file the user-defined at the end of the DEVEDIT command file.

2.4: Advanced Features

2.4.1: 3D Structures

Treatment of three dimensional structures in DEVEDIT3D is much the same as the two dimensional version with the exception of the Z plane. The handling of region addition and modification, electrodes, impurities, mesh parameters and constraints, and boundary conditioning are all done similarly in 3D as 2D. Anyone wishing to use DEVEDIT3D should either read this first or both of these examples.

DEVEDIT3D creates a prismatic based 3D solid consisting of several 2D planes. Each region is initially defined throughout all Z planes, then the region must be defined for beginning and ending Z planes.

One capability of DEVEDIT3D is to extend an existing 2D structure (created in ATHENA) into three dimensions. Such an example is discussed below.

Start DEVEDIT3D by entering `devedit3d &` in a UNIX terminal window. In a similar fashion to Section 2.3: “EXAMPLE 2 - REMESHING AN EXISTING STRUCTURE”, start DECKBUILD and load the example `mos2ex04` into DECKBUILD. In DEVEDIT, load in `mos2ex04_0.str` by right-clicking on **File**→**Load...** and a **DevEdit Load** window will appear. Then, select the file `mos2ex04_0.str` and enter an **End Z** value of 1.1 and click on **Load File**.

You can modify the polysilicon gate in the same manner as the 2D version by selecting the region and by selecting **Regions**→**Modify Region**. The X and Y points are maintained, but the **Start Z** and **End Z** planes should be changed to 0.3 and 0.8 respectively. Click on **Apply**.

To add aluminum electrodes, select **Regions, Add region...**, and specify the X and Y coordinates (either by mouse or by keyboard) of (0.15, -0.15), (0.15, 0.05), (0.6, 0.05), and (0.6, -0.15). The **Start Z** and **End Z** should be set to 0.0 and 0.1 respectively. Toggle the **electrode** button, select **drain**, and click on **Apply**. A similar set of X and Y coordinates should be used to define a source contact, but from Z=1.0 to Z=1.1.

The remaining difference between the 2D and 3D versions of DEVEDIT is the Z-plane mesh. Right click on **Mesh**, then select Z planes. Mesh planes in the Z plane are automatically inserted at material boundaries. These mesh planes are listed in **boldface** and cannot be modified or deleted. Additional mesh planes are included, and are listed in *italics*, which can be edited.

You have three methods of controlling the Z plane mesh:

- **Max. Plane Spacing:** Limits the distance in the Z plane of the mesh.
- **Max. Spacing Ratio:** The ratio of the spacing from one neighboring mesh plane to the next.
- Selectively add, modify, and delete specific mesh planes by entering the location and spacing of the plane.

DEVEDIT3D, however, overrides modifications you have entered to maintain the **Max. Plane Spacing** or **Max. Spacing Ratio** constraints.

There are no 3D graphics in DEVEDIT3D. To view the 3D structure after meshing, save a structure file. This file can be loaded and displayed in TONYPLOT3D.

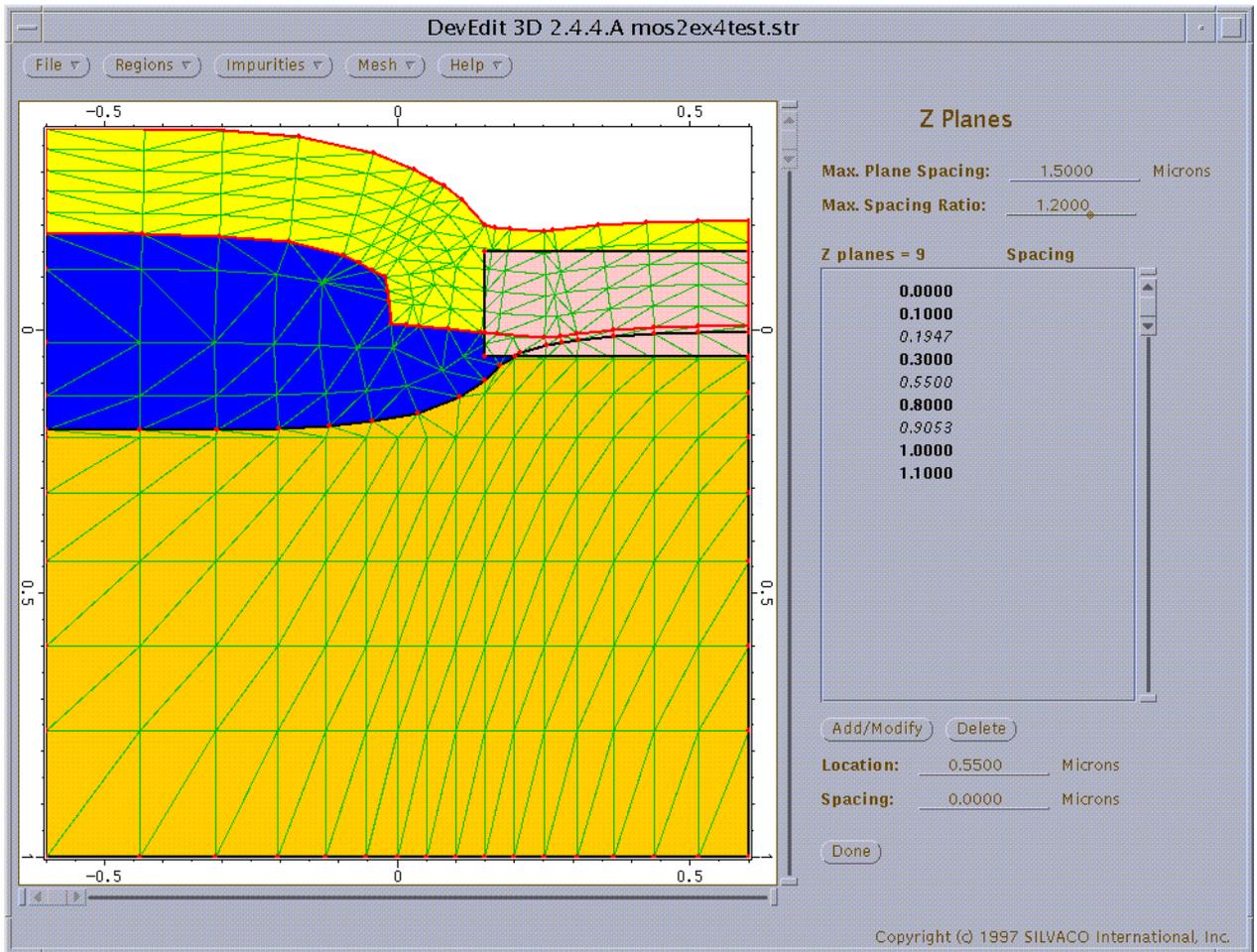


Figure 2-23: Three-dimensional Mesh Creation in Z-plane

2.4.2: Combining Two ATHENA Structures into a Single Device

Since CPU time required to run process simulation is super-linear with the number of grid points, there can be significant CPU time savings by splitting a large simulation into sections. These sections can be joined together at the end of the process simulation using DEVEDIT.

The **JOIN** function combines the device currently loaded in DEVEDIT with another saved structure into one device. There are controls for aligning the edges of the materials or regions.

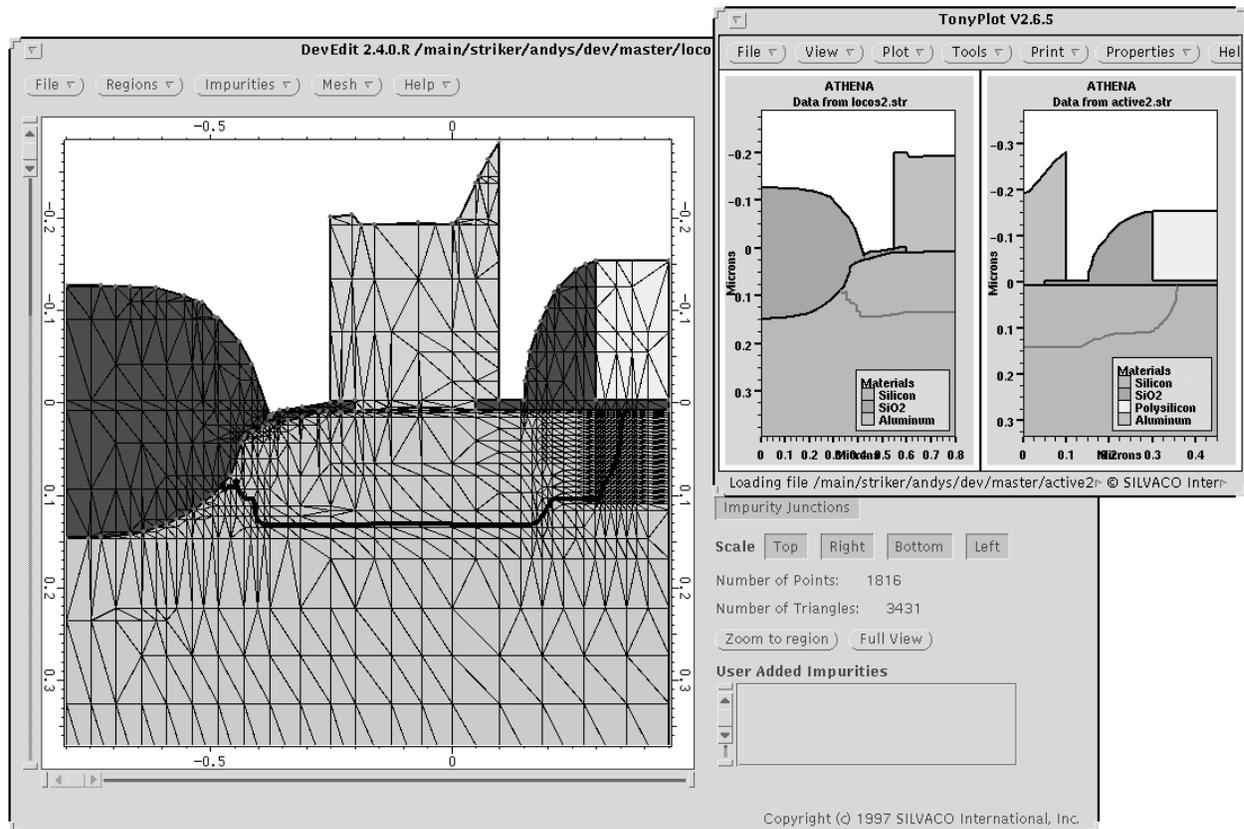


Figure 2-24: Device Structure Joined in DevEdit

2.4.3: Stretch and Cut

DEVEDIT has three commands, **STRETCH**, **SQUEEZE**, and **CUT**, to modify the structure and to increase or decrease height or width. Both commands are available in both the GUI mode and batch mode of DEVEDIT. These commands are under **Regions** in the GUI mode of DEVEDIT.

STRETCH and **SQUEEZE** functions allow a line or region to be expanded or contracted along the vertical or horizontal directions. For rounded shapes, the curvature is altered, but left intact. Stretch does not work as a simple **add** function. Otherwise, it causes a discontinuity in the curvature.

With an existing structure in DEVEDIT, select **Regions**→**Stretch/Squeeze**. The Direction buttons will appear. These buttons allow you to stretch either horizontally (left or right) and vertically (up or down). The mouse or keyboard can be used for **Start** and **End** locations. The **Old Size** indicates the existing size of the line or region selected. The **New Size** allows you to adjust the region size to either smaller or larger than the existing size.

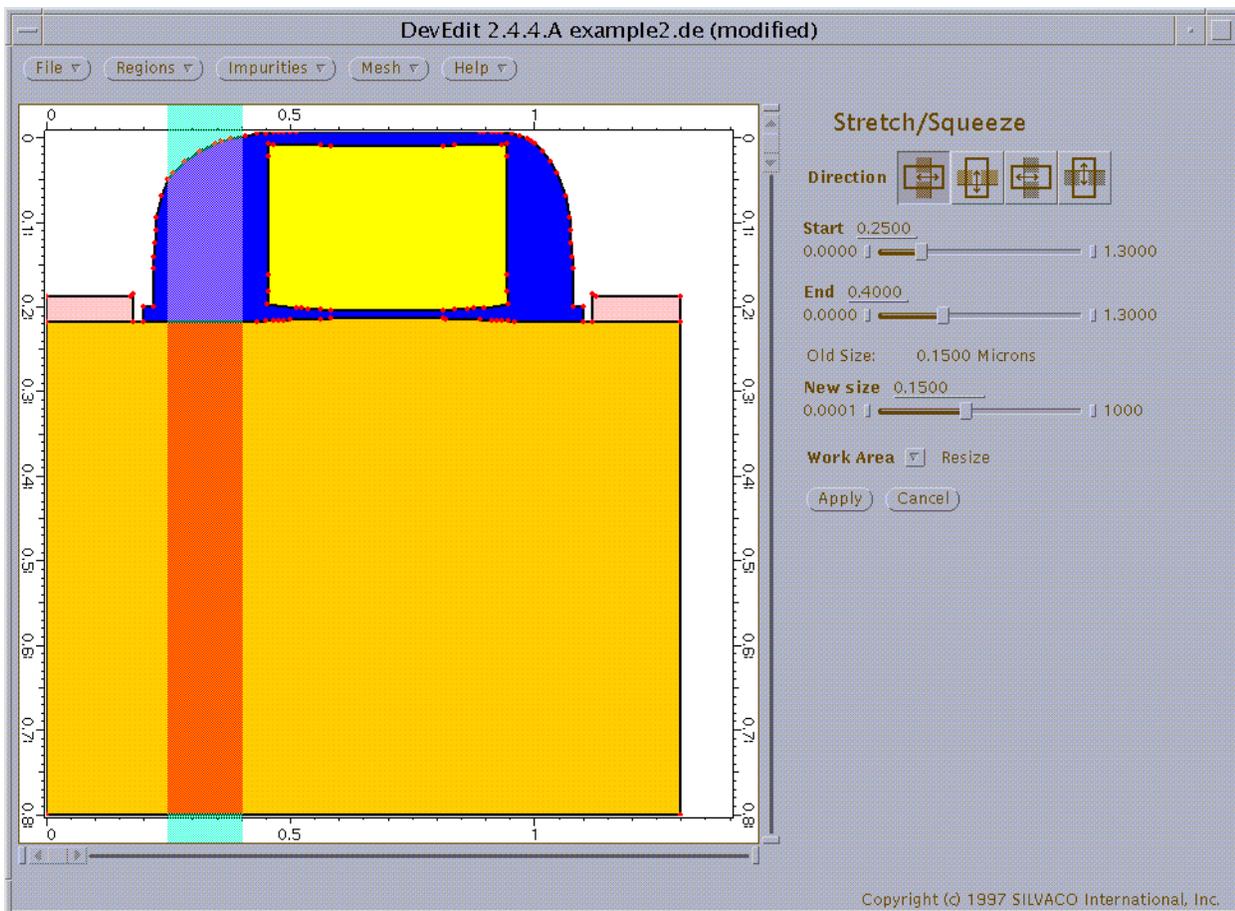


Figure 2-25: Stretching an Existing Structure

The **Cut...** function works in a similar manner. You can select the area using the mouse or keyboard and delete the area.

2.4.4: Circular Devices

DEVEDIT also allows you to create circular objects. You can create both circles and arcs. This feature is unique to DEVEDIT within the SILVACO VWF suite of tools - neither ATHENA nor ATLAS can easily create a circular region. To create a circular object, select **Region**→**Add Region**. Then, right click on **Move/Add Point** and select **New Circle/Arc**. The right panel then displays **Create Circle or Sector**. The first **Mouse mode** setting is **Set Center...** Click on a point in the work area to select the center of the circle. After selecting a center point for the circle, the **Mouse mode** changes to **Set Radius and Start Angle...** You can select a complete, closed circle, or restrict the region to a portion of a circle. Once chosen, the **Mouse mode** changes to **Set End Angle...** The **Points around circle** option allows you to determine the number of points of the circle.

DEVEDIT is still approximating a circular region with straight interconnecting lines. But you can choose between 30° segments (12 points) to 1° segments (360 points). The latter approximating a circle quite well.

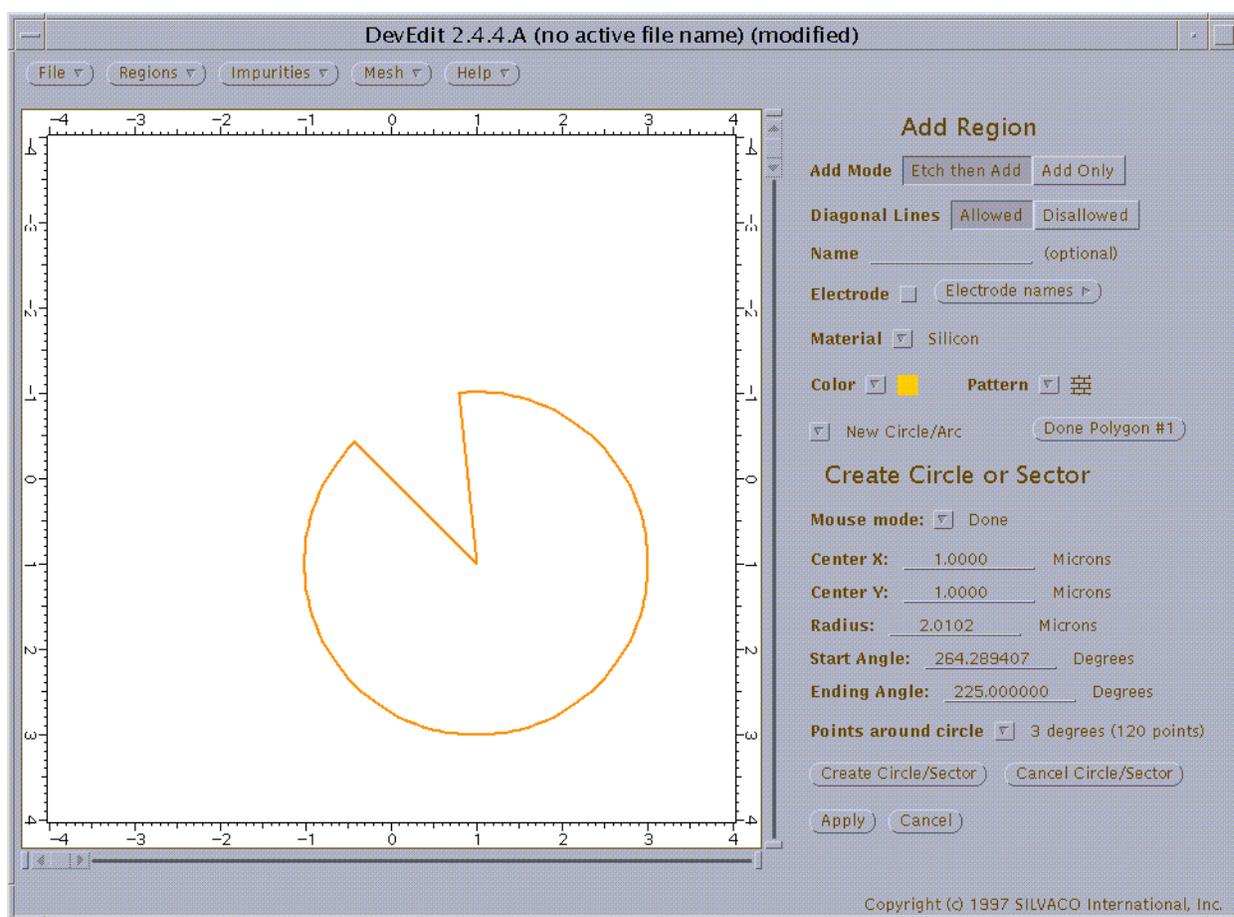


Figure 2-26: Creating a Circular Region

2.4.5: Summary

The goal of creating a mesh in DEVEDIT is to accommodate competing interests of numerical efficiency and accuracy in device simulation. DEVEDIT can be used to create a device structure in a graphics user interface mode (as in Section 2.2: “EXAMPLE 1 - CREATE A NEW STRUCTURE”), or to create a new mesh in an existing structure (as in Section 2.3: “EXAMPLE 2 - REMESHING AN EXISTING STRUCTURE”).

Parameters that you defined control the advanced mesh algorithm within DEVEDIT. Advanced features in DEVEDIT include 3D structure creation and prismatic mesh generation, combining two structures into a single device, stretching and cutting, and circular device creation.

Beginning users can find examples in DECKBUILD EXAMPLES of DEVEDIT in batch mode. If you need help, you can contact SILVACO's Applications Engineers (support@silvaco.com).

3.1: STRUCTURE EDITING

This section describes the general editing functions that are available in GUI mode. DEVEDIT stores a history of all events that have taken place under DEVEDIT control. These events are stored internally and can be written to a **Command** file (see Section 3.6.6: “Saving the SILVACO Standard Structure File”). The GUI mode is not available on Windows.

3.1.1: Zooming

The **Zoom** function is completed by holding the left mouse button down and dragging over the desired zoom area. In the event of needing to zoom in halfway through object definition when the mouse button cannot be held down, holding the Shift key down overrides any editing actions and zooms in. Zooming out is accomplished by double-clicking. This action zooms out by a factor of four times.

3.1.2: Panning

Panning over a structure is accomplished by moving the scroll bars situated initially at the top right and bottom left of the editing screen. Holding the left mouse button down while selecting either end of the scroll button moves the displayed structure to the left and right and up or down. A single click on the scroll bar end stops and pans all the way across the displayed structure. Panning can also be accomplished by holding the middle mouse button down on the inside the main edit window, while dragging the structure vertically or laterally across the screen.

3.1.3: Editing Summary

The capability of DEVEDIT to define structures on the screen is explained in the next three sections (**Material**, **Doping**, and **Meshing**). A region is generally defined as a piece of material structure, for example, a gate, an oxide, an aluminum contact, or a poly layer. Adding a material region to the DEVEDIT structure is accomplished by following five simple steps.

1. Select the required resolution.
2. Select the **Add Region** mode.
3. Select the material, its color and doping concentration
4. Draw the region on the screen.
5. Click on the **Apply** button.

3.1.4: Selecting The Resolution

When drawing the material region, DEVEDIT snaps boundary point locations of the newly defined region to the resolution defined by the drawing grid. A drawing grid displayed as tick marks on the **Main** panel can be overlaid by clicking on the **Grid** button. If a very thin layer is required, the tick marks may not allow this resolution. Increasing the resolution can be accomplished by zooming into the region of interest. It may also be useful to split the screen allowing the use of more than one drawing resolution. Thus, the resolution of the screen should be chosen before adding a material region.

Note: Zooming in can be accomplished midway through a region definition by pressing the Shift key down. The Shift key overrides the current action so the left mouse button will always control a zoom in function while the Shift key is held down. Region definition can continue once the Shift key is released.

3.2: EDITING REGIONS

3.2.1: Adding a Region

To activate the **Add Region** mode, select **Regions**→**Add**. The right side of the DevEdit Base Window then changes to show the **Add Region** control panel. You can select the **Add** mode at the top of the **Add Region** panel. Two options exist and determine the way regions are added to an existing structure. The **Add Only** option adds a region without recessing into an existing region. In this mode, the existing regions take precedence over the new region being added. The **Etch then Add** option recesses a new region into an existing region. The new region being added takes precedence over the existing regions.

Note: You can etch away or delete a section by creating a new region using the **Etch then Add** mode and deleting the whole region after defining it.

3.2.2: Selecting Region Material

A region's material should be selected before, or during, the process of region definition. Although **Silicon** is the default material, you can select other materials from the **Material** menu on the **Add Region** panel. A large number of materials are currently available on the **Material** menu, and more materials can be provided by contacting the local SILVACO representative.

Each material is assigned a different color by default. This can be changed, if two regions have to be distinguished, by selecting a color from the **Color** menu on the **Add Region** panel (Figure 3-1). Colors are selected purely for DEVEDIT visualization. The color information is not saved into the final structure file and so it is not applicable for later applications, i.e., TONYPLOT.

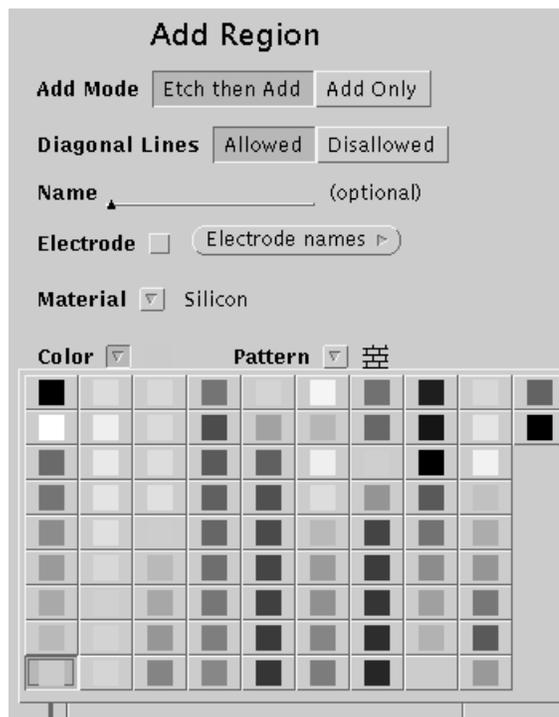


Figure 3-1: Color Menu

3.3: DRAWING REGIONS

A region is normally defined by a polygon, which denotes its outer boundary. This boundary can be drawn by clicking the left mouse button on the first point of the polygon. The point is placed when the left mouse button is released, not when it is pressed. This is to match the feel when points are dragged and is described later. This starts a new polygon. After placing the first point, a line is rubber-banded to the mouse until the second point is placed. After that, two lines are rubber-banded. One from the previous point to the mouse; the other from the mouse to the first point creating a polygon. Points are added until the **Done Polygon** button or the right mouse button is clicked. When the **Done Polygon** button appears, you can click on it with the right mouse button in the drawing window.

For example, to define a rectangular region, click the left mouse button four times in the screen at the vertices required to make up the rectangle. Then, click the right mouse button to finish the polygon. You can make selections from the right side control area any time. An example of this operation is shown in Figure 3-2, where two regions have been defined with a third region partially defined.

After clicking on **Apply**, each region added to the screen is listed as a material and as a number on the scrolling list in the region screen. You can select regions by clicking left on an entry in the scrolling list. A selected region is surrounded by a red outline in the main edit window.

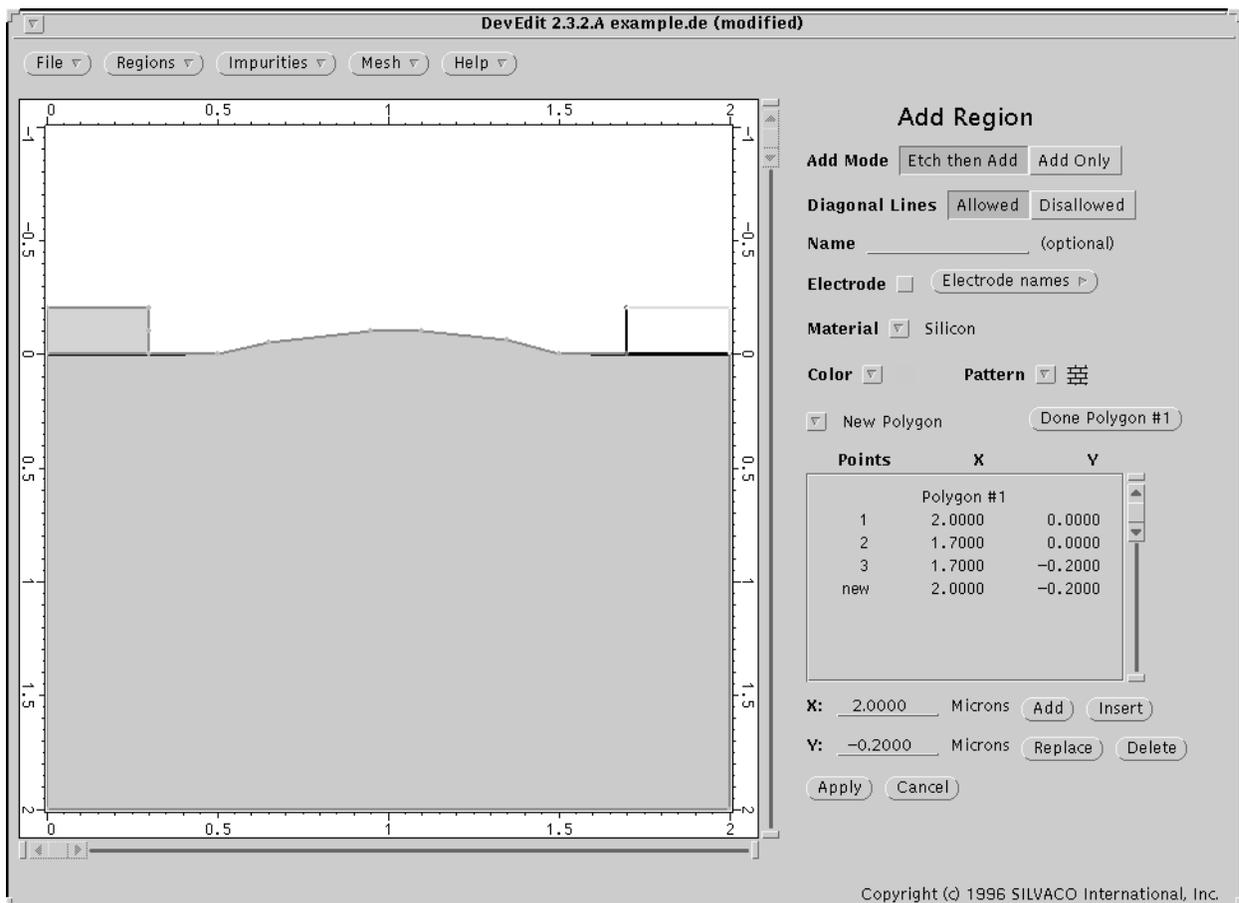


Figure 3-2: Drawing a Region

Regions can also be made from multiple polygons. For example, an oxide region can completely surround a polysilicon region, thereby having two polygons (an outer edge and an inner edge). Normally, this would be done by first adding the oxide as one polygon. Using the **Etch then Add** mode, you can add the polysilicon region to change the oxide region in the process. You can add a second polygon, after completing the first polygon, by selecting **New Polygon** or by clicking on the left button at the starting location for this new polygon. Use the **New Polygon** option, if you wish to start the new polygon near an existing point or line. Otherwise, it is considered a modification operation and does NOT start a new polygon.

Alternatively, a region may contain no area but only describe an interface line. This is sometimes used for electrodes to limit the mesh size or show a contact beginning at a simulation limit. To draw this type of region before adding any points, select **New Line** instead of **New Polygon**.

You can create a polygon by selecting **New Circle/Arc**. This forms a regular polygon, or a piece of a regular polygon, after selecting various parameters. These parameters include the center of the polygon/circle, a radius, a start and ending angle, and the number of faces (angle between points).

3.3.1: Setting Base Impurity

A region can be made of a material with a constant base doping. To define this property, select **Set Base Impurities**. This completes any shape editing currently performed if possible. You cannot change the base impurities while adding or modifying a region until the region has a consistent shape. The impurity types are separated into five categories. Here is a list of the supported impurities:

Silicon Donors

- Antimony
- Arsenic
- Phosphorus

Silicon Acceptors

- Aluminum
- Boron
- Gallium
- Indium

Generic Donors/Acceptors

- Donors
- Acceptors

Composition Fractions

- Comp. Fraction X
- Comp. Fraction Y

Net Doping (directly)

You can also set some of these values using **Resistivity** tables by clicking on the **Set Si Resistivity**.

3.3.2: Deleting Regions

You can delete a region by first selecting the region to be deleted and selecting **Regions**→**Delete**. To select a region, select the region listed in the scrolling list of regions on the main panel. A selected region is surrounded with a red colored highlighted border in the main screen. Adding a region can be used to etch a region out of an existing structure.

3.3.3: Modify Regions

To modify a region, select the region on the main panel in the region list and select **Regions**→**Modify**. The region parameters can now be changed as in the add region.

3.3.4: Deleting Boundary Points

Boundary points on a region's boundary can be deleted at any time. To delete a single point or a number of points, select the **Delete Points** option. The cursor appears as a skull and cross bones. Hold the left mouse down and drag a box over the points to be deleted. Upon releasing the mouse button, the points are deleted and the region modified. This may be important in minimizing the number of unnecessary points leading to large numbers of triangles.

3.4: DOPING DEFINITION

Doping can be defined analytically with DEVEDIT. Doping can be graphically defined by indicating an area or line. This line or box is given attributes that describe in detail the doping to be added to the structure.

3.4.1: Defining an Impurity Source Line

You can define doping about a line source drawn on the device. The line source can be vertical or horizontal to the main axis. To define a doping source line, select **Impurities**→**Add**. The Impurity Panel will appear in the area to the right of the base window (see Figure 3-3). Set the **Draw Mode** switch to **Line**. A line source has a constant doping concentration along that line segment and the concentration can roll off in magnitude in any perpendicular direction from the line segment. To define the line on the screen, click the left mouse button at the start of the line and click again at the end of the line. Once drawn, you can adjust the line by clicking a revised point at either end of the visible line. The end of the line to be modified is determined at about the half way point along the line. Thus, modifying an existing line is simple so that placement can be accurate. The position of a line is fixed only when you press the **Apply** button on the main doping control screen.

Add Impurity

Impurity

Color

Draw Mode

Start X: Microns

Start Y: Microns

End X: Microns

End Y: Microns

Peak Concentration:

Reference Value:

Join Function

Y Rolloff:
Distance:

X Rolloff:
Distance:

Figure 3-3: Impurity Panel

3.4.2: Defining an Impurity Source Box

You can define a doping distribution about a box drawn on the device. The box source can be vertical or horizontal to the main axis. To define a doping source box, select **Impurities**→**Add**. The Impurity Panel will appear in the area to the right of the base window (see Figure 3-3). The control screen to the right changes at this point. Set **Draw Mode** switch to **Box**. A box source implies a constant doping concentration within the box and the concentration can roll off in magnitude in any perpendicular direction to the box. To define the box on the screen by holding down the left mouse button at a box corner and dragging the cursor over the screen. Once drawn, you can adjust the box by moving the top right and the bottom left corners of a box. You can move a corner by holding down the left hand mouse button at the corner to be moved while dragging the mouse. Thus, modifying an existing box is simple so that placement can be accurate. You can place a box by defining the coordinate points of the boxes sides. You can define these fields by filling in the **Start X**, **Start Y**, **End X**, and **End Y** text field entries. This feature is often useful for more exact placement. The position of a box is only fixed when you click the **Apply** button on the main doping control screen.

3.4.3: Doping Source Attributes

You can define doping to roll off in perpendicular directions to the source boundary (either a line or a box). These attributes are simple analytical expressions that have traditionally been used in the description of semiconductor devices. These functions can be listed as follows:

Gaussian

- Error Function
- Linear
- Logarithmic
- Exponential Constant
- Step function

Each of the functions refer to the way the doping distribution decays with distance from the edge of the doping source (box or line).

3.4.4: Deleting Source Objects

You can delete a doping source object by selecting the object from the list on the top level menu and selecting **Impurities**→**Delete**. A selected object is surrounded by a red line on the screen.

3.4.5: 3D Doping

You can define three dimensional doping objects in the case of a 3D structure. To do this, set `DEVEDIT` in 3-D mode by entering:

```
devedit3d &
```

Specify X, Y and Z parameters in the case of a 3D doping objects being added to the structure. As in the 2D case, this action is controlled from the **Add Impurities** control panel as shown in Figure 3-5.

3.5: MESHBUILD MESHING

To mesh a structure, select **Mesh**→**MeshBuild**. The **MeshBuild** code originated from ETH Zurich and has been enhanced for use inside the DEVEDIT environment. **Meshbuild** starts with a mesh adapted to the defined boundary. Efficient meshes require that the boundary is conditioned before meshing. This means the boundaries are altered slightly to accommodate the meshing algorithm. **Meshbuild** allows the selection of arbitrary boxes in space for either manual mesh refinement or manual mesh relaxation. **Meshbuild** can be passed a solution quantity on the mesh (i.e., boron concentration), whereby it automatically refines the mesh based upon the gradient of the quantity. You can mesh a structure by selecting **Mesh**→**Meshbuild**.

Note: Boundary conditioning is strongly advised just before meshing any structure, although it is not essential.

During the mesh routine, an ETH (Zurich) Meshbuild Acknowledgment screen appears. You can cancel meshing at any time by pressing the **Cancel** button.

3.5.1: Boundary Conditioning

To use Boundary conditioning menu option, select **Mesh**→**Mesh Parameters**. The **Boundary Conditioning** control panel will then appear (see Figure 3-4).

Use boundary conditioning before creating any mesh. If you modify a structure, condition the boundaries before remaking the mesh.

The image shows a dialog box titled "Mesh Parameters". It has two main sections. The first section, "Mesh Parameters", contains three input fields: "Base Mesh Height" with a value of 0.2500 and unit "Microns", "Base Mesh Width" with a value of 0.2500 and unit "Microns", and "Max. Triangle Ratio" with a value of 100.0000. The second section, "Boundary Conditioning", contains a radio button group with "Off", "Manual", and "Automatic" options, where "Automatic" is selected. Below this are three more input fields: "Max. Line Slope" with a value of 30, "Rounding Unit" with a value of 0.0010 and unit "Microns", and "Line Straightening" with a value of 1 and unit "Degree(s)". At the bottom of the dialog are four buttons: "Apply", "Cancel", "Default", and "Restore".

Figure 3-4: Boundary Conditioning Panel

3.5.2: Limitations

A few basic limitations of **Meshbuild** should be realized before starting to use the mesh generator. **Meshbuild** creates a mesh with few obtuse triangles. This improves convergence during subsequent solutions. In order to maintain this criteria, **Meshbuild** adds a large number of triangles in or around border points. For this reason, try to minimize the number of border points. You can make slight modifications to the structure to minimize the number of boundary points. This concept is called Boundary Conditioning (see Section 3.5.1: “Boundary Conditioning”). Selecting **Mesh**→**Refinement Limits** sets up the Boundary Conditioning Control panel (see Figure 3-4). Here, you can make a number of approximations to improve the meshability of the structure. Most structures can be conditioned by clicking on the **Apply** button. This should be the first option. Boundary Conditioning attempts to get rid of points that are not critical.

Boundary Conditioning falls under the following categories:

- Any points on a straight line not contributing to the geometry of the structure are eliminated.
- A geometrical rounding error can be supplied with a default limit of 0.001 micron. Be careful of this critical layer thicknesses when using this option (i.e., gate oxides). This option is controlled with the **rounding unit** text field on the boundary conditioning control screen.
- A line straightening algorithm is available. The option straightens a line if it bends by only a small amount defined as an angle in the **line straightening** text field.
- **Meshbuild** essentially refines a basic, coarse, tensor product mesh. You can control the spacing of this base mesh. To do this, use the two options for the X and the Y directions available on the Boundary Conditioning Control Panel.

3.5.3: Mesh Constraints

Mesh constraints allow you to require specifications be met in the mesh. While mesh parameters are guidelines for the meshing algorithm, mesh constraints are requirements that must be met (see **Maximum Mesh Angle** for exception). There are two major parts in defining mesh constraints. The effected area and the constraints being opposed on that area. The types of effected areas are described in the next few paragraphs and the types of constraints that can be applied.

Constraint areas are either region based or rectangle based. Region based constraint areas are set up in a hierarchy where the most specific constraint area overrides the less specific areas that apply. Rectangle areas, as you may have guessed, are rectangles. They are specified by absolute coordinate or in relationship to regions (as under metal regions). When two rectangles overlap, the constraint applied to the overlapping area is the most restrictive constraint. When region constrains and rectangle constraints overlap, the most restrictive constraint is also used.

In region based constraints, each region has its own set of mesh constraints. If a specific constraint is not set for a region, the value from the mesh constraint specified for that region’s material type is used. If it is not set, the value from the **All Regions** mesh constraint is used. This allows setting constraints for all regions, a specific material type or a specific region.

The scope list allows you to specify the set of mesh constraints. You must turn on the check box to the left of each value if the value is to be changed. For **All Regions**, the check boxes are always set because those values are the default values for all other scopes. For material type scopes (e.g., **Semiconductor Regions**, **Insulator Regions**, and **Metal Regions**), turning on the check box sets the default value for regions made of that material type, overriding the **All Regions** value for those regions. If you set the check box for a particular region, the value to the right of the check box applies only to that region.

Note: When you turn off the check box, the value displayed is the current default value for the current score.

Finally, we get to the constraints that can be imposed.

max.angle - Maximum angle of a triangle in the specified region or scope. In certain cases, limitations in the mesh build algorithm require a few triangles to be obtuse, regardless of the setting of the maximum angle. This keeps mesh build from creating an infinite number of triangles. Work is currently ongoing at SILVACO to resolve this restriction. If the angle is less than 180°, the maximum connectivity of a node is 12.

max.triangle.ratio - Maximum ratio of a triangle's height and width.

max.height - Maximum height of a triangle.

max.width - Maximum width of a triangle.

min.height - Triangles with height less than this value are not refined in the Y direction during impurity refinement. Some triangles may be shorter than this value to allow for geometry

min.width - Triangles with width less than this value are not refined in the X direction during impurity refinement. Some triangles may be narrower than this value to allow for geometry.

3.5.4: Adaptive Meshing

Adaptive meshing is an efficient way of adding grid points in areas of interest semi-automatically without having to add too many points. The purpose is to search for steep gradients over a solution and to add grid points locally to these regions. A mesh can be adapted after creating a basic boundary compliant mesh. A mesh can be adapted to the gradient of any impurity quantity on the mesh. To generate an adapted mesh, select **Mesh**→**Impurities**→**Refine**.

3.5.5: Refinement

Refinement on impurity concentration gradient control shows the control screen for impurity defined mesh refinement. The **Minimum Mesh Spacing** control refers to the minimum size of a mesh element after adapting. A mesh is not be refined down below this entered value even in the case of large concentration gradients. You can select available impurities from the **Add Menu** option. All impurities present in the current structure are available to mesh with (e.g., in the case of a SPISCES2 solution) a large number of impurities, such as Electron Temp, doping conc, and potential. You can select any number of impurities from this menu for addition to the scrolling list of impurities.

Refinement on impurity concentration gradient control show three impurities: boron, arsenic and phosphorus. The concentration gradients of these selected impurities is used to adapt the mesh. Each impurity has its selected weight (e.g., boron = 0.85). Each selected impurity can be weighted by selecting the impurity on the scrolling list and sliding the weighting factor slider to the required weight. The smaller the number, the finer the grid. The weighting factor refers to the natural log increase of a quantity that is accepted without halving the mesh spacing. For example, if the weight for phosphorus was set to 3, the grid would only half its size when the concentration was found to be more than $(\exp 3) = 20.03$ X difference between two adjacent points. By default, the weight is 1, so each time a value is multiplied by 2.72 across two grid points, it is earmarked for adapting.

To delete selected impurities from this list, click on the listed impurity and press the **Delete** button. Impurities have to be present in the structure before refinement can take place. Once you setup the weights, click on the **Apply** button. To start the meshing procedure, select **Mesh**→**Meshbuild**.

Boundary conditioning is advised before any meshing procedure (see Section 3.5.1: "Boundary Conditioning"). You can apply boundary conditioning either before or after setting up the adaptive meshing settings.

3.5.6: Manually Refining The Mesh

You can refine any mesh by selecting the menu options. You can refine a selection box in both directions. Alternatively, you can restrict refinement to either the X or Y direction by selecting either of these options: **Mesh**→**Refine Box**→**Refine X** or **Mesh**→**Refine Box**→**Refine Y**. To specify a selection box, hold down the left mouse button and drag over the area to be refined. Each element in the box is reduced in size by a factor of two per direction. You can relax the mesh inside the selection box in a similar way by selecting the **Unrefine** menu option. **Meshbuild** may also affect the mesh outside of the defined box.

3.5.7: Manually Relaxing A Mesh

You can either relax refinement of a mesh manually or adaptively on an impurity gradient. A mesh that was generated purely from a set of boundary points cannot be relaxed. To relax a mesh, select **Mesh**→**Refine Box**→**Unrefine**. Define a box by dragging the mouse over an area with the left mouse button held down. The mesh is refined in the area of the box. **Meshbuild** may also affect the mesh outside of the defined box. The unrefine attempts double the size of each element inside the defined bounding box.

3.5.8: Tensor Product Mesh

A tensor product mesh is one that runs from top to bottom and from left to right at every point on the mesh. It is highly inefficient but produces straightforward matrix conditioning. After a structure has been defined, it can be meshed using a tensor product algorithm.

DEVEDIT supports a modified tensor product mesh using the following algorithm. All boundary points are used to create a true tensor product mesh. But since there can be new points created during that process, the **MeshBuild** algorithm is used to resolve these new points. This also allows you to refine the resulting mesh using the **Refine Box** commands. To create a modified tensor product mesh, select **Mesh**→**Tensor Product**.

3.5.9: Work Area Resizing

To resize the work area, select **Regions**→**Resize Work Area**. You can then add new coordinates to the **Resize** work area control panel. Click on **Apply** when done.

3.5.10: Defining 3D Structures

You can specify a prismatic based 3D solid in terms of a number of 2D planes. This is the approach DEVEDIT uses. A super set of all regions are defined, covering all potential Z-planes. Then, individual regions are assigned a start and end Z-plane. For example, imagine a MOSFET conducting into the plane of the screen. A poly gate runs from left to right across the screen. Field isolation regions also appear on the left and right hand side of the screen. You need to specify the values of the poly region in the Z direction. These two values, start and stop, define the gate length. Similarly, you can define all contact metal regions with a start and stop value in the Z direction. As a region is defined, the **Start Z** and **Stop Z** fields should be specified. These values indicate the depth, into the screen, that the region boundaries apply. Therefore, you can define a 3D structure from a single elevation.

3.6: CREATING A NEW MESH

3.6.1: Setting The Mesh Controls

To create a mesh, the first step is to set the meshing controls. This is accomplished by setting the Meshing Parameters. To do this, select **Mesh**→**Mesh Parameters**. A new control panel will appear on the right side of the DEVEDIT base window.

3.6.2: Base Mesh Parameters

Base Mesh Parameters are a guideline to create a mesh that fit the structure. Set the width and height to 0.2 Microns. **Boundary Conditioning** controls simplification of the input structure. When the structure first reads in, all the original boundary points appear as red dots. Boundary conditioning moves these points slightly or eliminates them if the points are unnecessary. At this point, click on **Apply**. It may be noticed that some of the red dots have disappeared. Next, set the refinement parameters and select **Mesh**→**Refinement Impurities**. This panel controls areas have a finer mesh than the others based on the value of the impurity.

Note: Refining on impurities only takes place in the Semiconductor region.

3.6.3: Refining On Impurities

To cause refining for all doping, especially at impurity junctions, select **Add**→**Net Doping**. Next, to cause a reasonable number of triangles in the channel under the gate, select **Add**→**Boron**. To make more triangles, set the sensitivity of boron to 0.2 on the lower part of the **Refinement Impurities** panel. The lower the number, the more triangles that are generated.

3.6.4: Mesh Constraints

The final operation is to set the mesh constraints. To do this, select **Mesh**→**Mesh Constraints**. In this example, it is assumed that obtuse triangles are acceptable in all regions except in the semiconductor. First, select the **All Regions** and set the **Max. Angle** to 180 either using the slider or entering in the value. This causes generation of far fewer triangles and points. But since most simulators give poor results, or no results at all, if there are obtuse triangles in the semiconductor, the semiconductor regions need to have a stronger restriction. Select **Semiconductor Regions** and observe that the **Max. Angle** reads 180.0. To override the **All Regions** setting, select the check box to the left of **Max. Angle** and set it to 90.0. You can override this value in a specific semiconductor region by selecting that region and setting its values.

3.6.5: Final Meshing

At this point DEVEDIT is ready to mesh, click on **Done**. To mesh, select **Mesh**→**Meshbuild**. Since **MeshBuild** is the default action on the **Mesh** menu, you can also select it by simply clicking on **Mesh**. DEVEDIT now produces a mesh. A **Cancel** button will appear during meshing if you supply an unreasonable meshing parameter (e.g., too many triangles are being produced).

3.6.6: Saving the SILVACO Standard Structure File

To save this file, select **File**→**Save as....** Then, specify a name to save the file (such as `test.str`) and click on the **SILVACO Standard Structure File** button. Then, click on **Save File** to produce a file readable by all SILVACO 2-D Simulators.

3.7: IMPURITIES

3.7.1: Viewing Impurities

Currently, you can view only net doping. To view net doping, pull down the **Show Net Doping** menu on the base window. This menu controls pixel resolution of the displayed net doping contours. To display the net doping legend, pull down the **Net Doping Legend** menu and select a location to place the legend. Doping at $1e+10$ level is considered to be approximately zero for graphing purposes.

3.7.2: Impurity Definition

Impurities can be read in when a structure file is loaded, or defined using a source line or box and rolloff function or profile. A structure file is a 2D or 3D impurity profile that is used to extrapolate on to the final mesh (these do not show up in the **User Impurities** panel list).

You can easily add a new impurity distribution by performing the following steps:

1. Enter the add impurity mode.
2. Define a impurity source area.
3. Define the rolloff directions.
4. Define the rolloff functions.
5. Define a join function.

3.7.3: Impurities Loaded From A Structure

When a structure file is loaded, the complete mesh is temporarily saved. When the device is saved as a structure file, each new mesh point's value is found by extrapolating the value based on the mesh point's material and the values from the closest triangle or prism in the original mesh. These values are added to any user-created impurities to create the final impurity profile.

3.7.4: Add Impurity Mode

To add an impurity distribution, select **Impurities**→**Add Impurity...** (that is the default action for **Impurities**). The Add Impurity panel will then appear (see Figure 3-5).

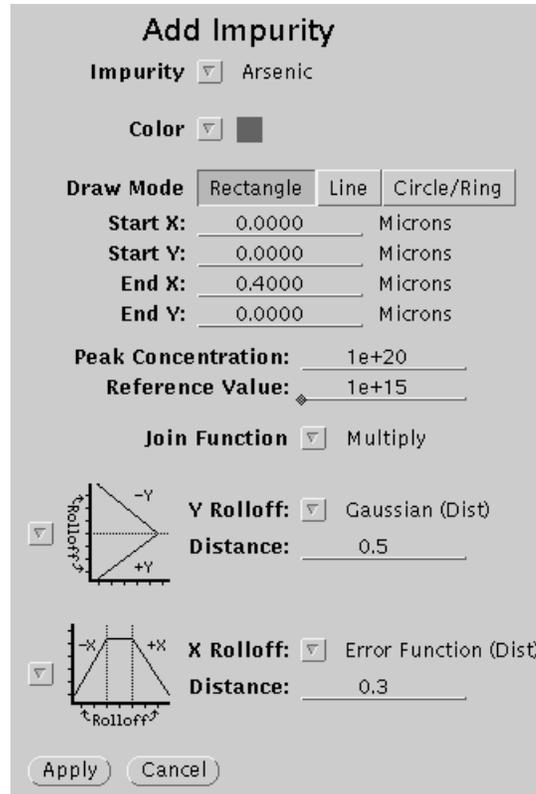


Figure 3-5: Add Impurity Panel

Note: You cannot select Add Impurity until you finish modifying an impurity. If you add an impurity, the Add Impurity panel will be displayed in its current state.

3.7.5: Defining An Impurity Source Area

The impurity source area is the area in which the impurity is set to a **Peak Concentration**. The area can be a box, a rectangle, a vertical line, a horizontal line, or point. For this area, there are 4 possible (or 6 in 3D mode) rolloff directions as follows:

- above the area (-Y)
- below (+Y)
- left (-X)
- right (+X)
- front (-Z) for 3D
- back (+Z) for 3D

These are fully described in Section 3.7.6: “Defining Impurity Roll-off Direction”.

To define the impurity source area, draw a rectangle on the existing device. To place the first corner, press and release the left mouse button over the desired location. Now move the mouse to the opposite corner, press and release the left mouse button again. To change the shape, move the mouse to the corner you wish to move, then press and hold the left mouse button. Move the pointer to the new position and release the button. Observe that the corner is placed. Changing **Draw Mode** to **Line** forces the drawing to be either a horizontal or vertical line.

You can alternatively set the locations manually by entering values into the fields **Start X**, **End X**, **Start Y**, and **End Y**. In 3D mode, you must enter the **Start Z**, and **End Z** fields. This feature is also useful for more exact placement.

3.7.6: Defining Impurity Roll-off Direction

After defining an impurity source area, you must define how the impurity decreases as the distance from the source area increases along each axis. The first step in this procedure is to select the appropriate icon desired. If the impurity source area is a rectangle or a vertical line, the **Y Rolloff** icon choices are as seen in Figure 3-6a. This is the case if the **Start Y** value is less than the **End Y** value. In this case, the detailed description for each icon can be found by locating the icon in column 1 of Figures 3-6a through 3-6b.

If the impurity source area is a horizontal line or a point, the **Y Rolloff** icon choices are as seen in Figure 3-7. This is the case if the **Start Y** value equals the **End Y** value. In this case, the detailed description for each icon can be found by locating the icon in column 2 of Figures 3-6a through 3-6b.

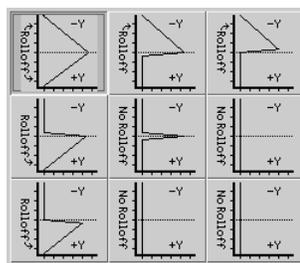


Figure 3-6a: Rolloff Icon
Start Y ≠ End Y

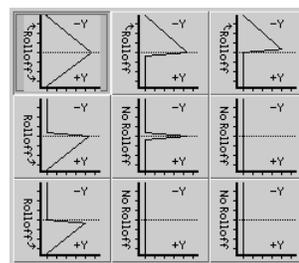


Figure 3-6b: Rolloff Icon
Start Y = End Y

If the impurity source area is a rectangle or a horizontal line, **X Rolloff** icon choices are as seen in Figure 3-7a. This is the case if the **Start X** value is less than the **End X** value. In this case, the detailed description for each icon can be found by locating the icon in column 3 of Figures 3-7a through 3-7b.

If the impurity source area is vertical line or a point, the **X Rolloff** icon choices are shown in Figure 3-7a. This is the case if the **Start X** value equals the **End X** value.

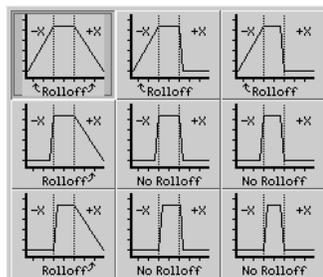


Figure 3-7a: Rolloff Icon
Start X ≠ End X

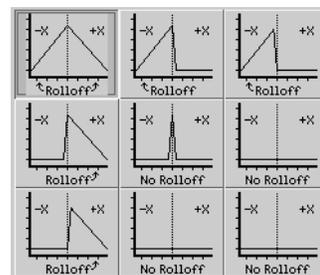


Figure 3-7b: Rolloff Icon
Start X = End X

Alternatively, the source area can radiate out of a circle. This is mainly used in 3D mode.

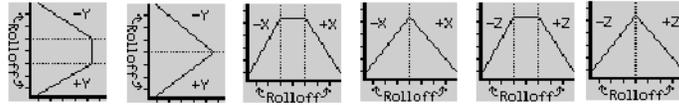


Figure 3-8: Circular

Rolloff=Both

Rolloff=Both (Figure 3-9) causes the same rolloff function to be used in both the negative and positive directions away from the impurity source area. While each axis has its own roll off function, to create different rolloffs in the negative and positive directions on the same axis, see Section 3.8.8: “Combining Impurity Rolloffs”.

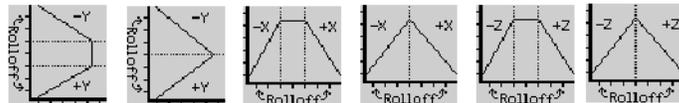


Figure 3-9: Rolloff=Both

Rolloff=High

Rolloff=High (Figure 3-10) causes the rolloff function to be used in the positive direction from the impurity source area. In the negative direction, the impurity value drops to (steps down to) zero if the distance is greater than zero.

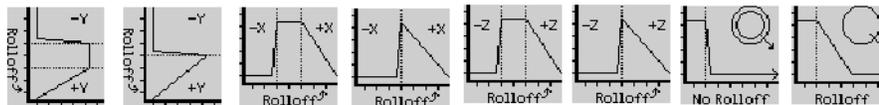


Figure 3-10: Rolloff=High

Rolloff=High.P.Step

Rolloff=High.Premature.Step (or **High.P.Step**) (Figure 3-11) causes the rolloff function to be used in the positive direction from the impurity source area. In the negative direction, the impurity value drops to (steps down to) zero if the distance is greater or equal to zero. This causes the negative edge of the impurity source area to have a zero value. As you can see, if the start value along the axis equals the end value, the entire impurity source area will have a zero value.

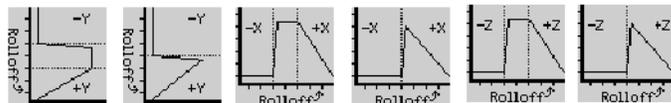


Figure 3-11: Rolloff=High.P.Step

Rolloff=Low

Rolloff=Low (Figure 3-12) causes the rolloff function to be used in the negative direction from the impurity source area. In the positive direction, the impurity value drops to (steps down to) zero if the distance is greater than zero.

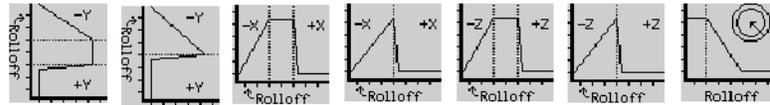


Figure 3-12: Rolloff=Low

Rolloff=Step

Rolloff=Step (Figure 3-13) causes no rolloff function to be used. The impurity source area contains the peak value, which immediately steps down to zero away from the impurity source area in the given direction.

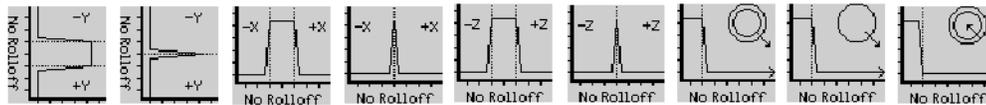


Figure 3-13: Rolloff=Step

Rolloff=Step.P.Low

Rolloff=Step.Premature.Low (or **Step.P.Low**) (Figure 3-14) causes no rolloff function to be used. The impurity source area contains the peak value except along the negative edge, which is zero. All areas outside the impurity source area also get zero values. As you can see in columns 2 and 4 of Figure 3-16, this is completely useless if Start X/Y equals End X/Y.

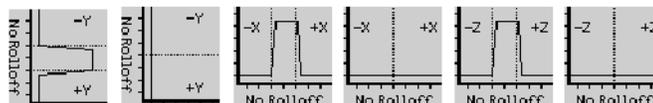


Figure 3-14: Rolloff=Step.P.Low

Rolloff=Low.P.Step

Rolloff=Low.Premature.Step (or **Low.P.Step**) (Figure 3-15) causes the rolloff function to be used in the negative direction from the impurity source area. In the positive direction, the impurity value drops to (steps down to) zero if the distance is greater or equal to zero. This causes the positive edge of the impurity source area to have a zero value.

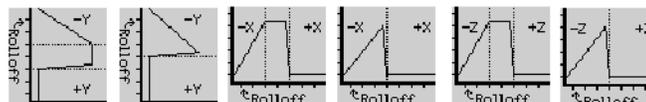


Figure 3-15: Rolloff=Low.P.Step

Rolloff=Step.P.High

Rolloff=Step.Premature.High (or **Step.P.High**) causes no rolloff function to be used. The impurity source area contains the peak value except along the positive edge, which is zero. All areas outside the impurity source area also get zero values. As you can see in columns 2 and 4 of Figure 3-16, this is completely useless if Start X/Y equals End X/Y.

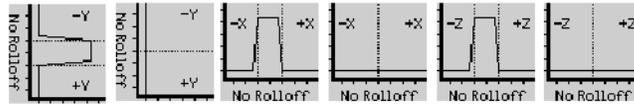


Figure 3-16: Rolloff=Step.P.High

Rolloff=P.Step

Rolloff=Premature.Step (or **P.Step**) causes no rolloff function to be used. The impurity source area contains the peak value except along the edges, which are zero. All areas outside the impurity source area also get zero values. As you can see in columns 2 and 4 of Figure 3-17, this is completely useless if Start X/Y equals End X/Y.

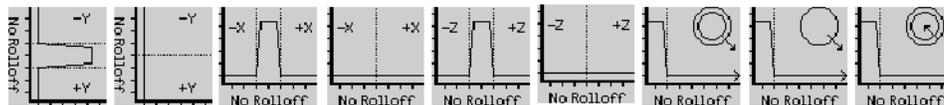


Figure 3-17: Rolloff=P.Step

3.8: ROLL-OFF FUNCTION

The **Roll-off** function calculates the vertical and the horizontal rolloffs separately and then uses a join function to connect the 1D rolloffs. In DEVEDIT3D, there is also a width distance rolloff. There are three types of 1D rolloff functions: analytic, analytic distance, and profiled. There are also three types of join functions: multiply, interpolate, and miter. When selecting a 1D rolloff function, the first column of choices are the analytic functions described below. The next column lists any added doping profiles. These are described just after the analytic functions. Therefore, the overall roll-off function is made by joining either 2 or 3 1D roll-off functions and a join function.

3.8.1: Analytic Functions

Analytic functions are used to describe the rolloff curve from the impurity source area. The function is given a distance (distance) to the points of interest and a user supplied constant. Analytic functions return $p=func(d, K)$ where p is between 1 (Peak value) and 0.

User Supplied Variables

You supply:

- Constant: K

Location Dependent Variables

- d = Distance from Base depending on joint function used.
- p = The result of the function, which is doping as the proportion of the Peak value.

Gaussian $p=func(d,K)$

$$p = \exp\left(\frac{-d^2}{2K^2}\right) \quad 3-1$$

Error Function $p=func(d,K)$

$$p = \exp\left(-1.02\frac{d}{K}-0.79276\frac{d^2}{K^2}-0.019345\frac{d^3}{K^3}\right) \quad 3-2$$

Logarithmic $p=func(d,K)$

$$p = \frac{1.0}{1.0 + \ln(dK + 1.0)} \quad 3-3$$

Exponential $p=func(d,K)$

$$p = \exp(-dk) \quad 3-4$$

Constant

$$p = 1.0 \quad 3-5$$

Step Function

Deprecated Option: Use either constant rolloff or rolloff icons instead of the step function. This option will be removed in future releases.

3.8.2: Analytic Functions (Dist)

Analytic functions are used to describe the rolloff curve from the impurity source area. The function is given a distance (distance) to the points of interest. You also supply a peak concentration, a reference concentration, and the distance between the two. Analytic functions return $p = \text{func}(d, P, R, D)$ where p is between 1 (Peak value) and 0. Therefore, $p \cdot P$ equals the value at distance d .

User Supplied Variables

You supply the following to derive k in the distance functions (Dist).

- Peak Concentration: P
- Reference Value: R
- Distance: D

Location Dependent Variables

- d = Distance from Base depending on joint function used.
- p = The result of the function, which is doping as the proportion of the Peak value.

Gaussian (Dist) $p = \text{func}(d, P, R, D)$

$$k = \frac{D}{\sqrt{-2 \ln\left(\frac{R}{p}\right)}} \quad \text{derived from} \quad \frac{R}{P} = \exp\left(\frac{-D^2}{2k^2}\right) \quad 3-6$$

$$p = \exp\left(\frac{-d^2}{2k^2}\right) \quad 3-7$$

Error Function (Dist) $p = \text{func}(d, P, R, D)$

$$\text{First solves for } k: \frac{R}{P} = \exp\left(-1.02\frac{D}{k} - 0.79276\frac{D^2}{k^2} - 0.019345\frac{D^3}{k^3}\right) \quad 3-8$$

$$p = \exp\left(-1.02\frac{d}{k} - 0.79276\frac{d^2}{k^2} - 0.019345\frac{d^3}{k^3}\right) \quad 3-9$$

Linear (Dist) $p = \text{func}(d, P, R, D)$

$$k = \frac{1.0 - \frac{R}{P}}{D} \quad 3-10$$

$$p = 1.0 - dk \quad 3-11$$

Note: If $p < 0$, then $p = 0$.

Logarithmic (Dist) $p=func(d,P,R,D)$

$$k = \frac{1}{D} \left[\exp\left(\frac{P}{R} - 1.0\right) - 1.0 \right] \quad 3-12$$

$$p = \frac{1.0}{1.0 + \ln(dk + 1.0)} \quad 3-13$$

Exponential (Dist) $p=func(d,P,R,D)$

$$k = \frac{1}{D} \ln\left(\frac{P}{R}\right) \quad 3-14$$

$$p = \exp(-dk) \quad 3-15$$

3.8.3: Doping Profiles

Doping profiles are a list of distances and impurity concentrations at those distances. These can be obtained from SSUPREM3 structure files, SSUPREM4 1-D structure files, and the SPDB doping database. By using these values and one of the four extrapolation functions, you can add an impurity using this profile as a rolloff function instead of one of the analytical profiles contained in DEVEDIT.

Adding a New Doping Profile

To add a new doping profile, select **Impurities**→**Doping Profiles**. The **User Defined Doping Profiles** control panel will appear on the right side of the screen. The top list shows the doping profiles already read into DEVEDIT. Below is the information about the selected doping profile. The name and impurity type can be changed in this area. To read in a SSUPREM3 structure file or a 1-D ATHENA structure file, click on the **Load File...** button. A popup window will appear. Select a file and load it. A popup notice will then appear asking which impurity you wish to load from that file. Once you load the profile and give the name `NewProfile001` or `NewProfile 002`, or the first new name not already in use.

Note: If the impurity selected is **active antimony**, **active arsenic**, **active boron**, or **active phosphorus**, DEVEDIT will store them as **antimony**, **arsenic**, **boron**, or **phosphorus**, respectively.

Back in the **User Defined Doping Profiles** command panel, this profile is added to the **User defined profiles** list and becomes the selected item. Details about this profile are displayed below the list. You can now rename the profile. Replace the name in the **Profile Name** field with the desired name. Be careful not to accidentally use a name that already exists in the list. You can now click on **Done** to remove the **User Defined Doping Profiles** command panel. This doping profile has now been added to the possible rolloff functions in the **Add Impurity** panel and the **Modify Impurity** panel.

If you select a user-defined profile as the rolloff function in the **Add Impurity** panel or the **Modify Impurity** panel, the peak value of that impurity distribution will be set to the value of the doping profile at distance equal zero. The impurity will also be set to match the profile. Both fields are grayed

out as long as a doping profile is being used as a rolloff function. While different doping profiles can be used in the X and Y directions, they must be compatible. In other words, have the same value at distance equals zero and be the same impurity type.

3.8.4: Join Function 2D

There are currently three join functions. First, the multiple join works by computing the Y rolloff and applying the X rolloff. This effectively is a multiply of the two rolloffs. Second, the interpolate join works by considering an arc at equal distances between the Y rolloff and the X rolloff though the locations and interpolating the values along the arc. Third, the miter join takes the lower value after allowing for the X rolloff or the Y rolloff.

User Supplied Variables

- Peak Concentration: P
- Reference Value: R
- Rolloff Functions: $func_x$ and $func_y$.

Either

- Constant: K_x and K_y
- Distance to Reference Value: D_x and D_y .

Multiply Join

- Peak Concentration: P
- d_x = X rolloff distance = lateral distance = Δx
- d_y = Y rolloff distance = depth distance = Δy
- $p_x = func_x(d_x, K)$ or $p_x = func_x(d_x, P, R, D_x)$
- $p_y = func_y(d_y, K)$ or $p_y = func_y(d_y, P, R, D_y)$

$$doping = p_x p_y P \quad 3-16$$

Interpolate Join

$$d = \text{total distance} = \sqrt{\Delta x^2 + \Delta y^2} \quad 3-17$$

- $p_x = func_x(d, K)$ or $p_x = func_x(d, P, R, D_x)$
- $p_y = func_y(d, K)$ or $p_y = func_y(d, P, R, D_y)$

$$doping = \left(p_x \frac{\Delta x^2}{d^2} + p_y \frac{\Delta y^2}{d^2} \right) P \quad 3-18$$

Miter Join

- d_x = X rolloff distance = lateral distance = Δx
- d_y = Y rolloff distance = depth distance = Δy
- $p_x = func_x(d_x, K)$ or $p_x = func_x(d_x, P, R, D_x)$
- $p_y = func_y(d_y, K)$ or $p_y = func_y(d_y, P, R, D_y)$

If $p_y < p_x$ then

$$doping = p_y P \quad 3-19$$

else

$$doping = p_x P \quad 3-20$$

To add these changes to the device, click on **Apply**. Click on **Cancel** if you do not want to add or modify this impurity.

3.8.5: Join Function 3D

The three dimensional join functions are about the same as the two dimensional ones with an extra term. First, the multiple join works by computing the Y rolloff, applying the X rolloff, and finally the Z rolloff. This effectively is a multiply of the two rolloffs. Second, the interpolate join works by considering an arc at equal distances between the rolloffs though the locations and interpolating the values along the arc. Third, the miter join takes the lower value after allowing for each rolloff seperately.

User Supplied Variables

- Peak Concentration: P
- Reference Value: R
- Rolloff Functions: $func_x$, $func_y$, and $func_z$.

Either

- Constant: K_x , K_y , and K_z
- Distance to Reference Value: D_x , D_y , and D_z .

Multiply Join

- Peak Concentration: P
- d_x = X rolloff distance = lateral distance = Δx
- d_y = Y rolloff distance = depth distance = Δy
- d_z = Z rolloff distance = width distance = Δz
- $p_x = func_x(d_x, K_x)$ or $p_x = func_x(d_x, P, R, D_x)$
- $p_y = func_y(d_y, K_y)$ or $p_y = func_y(d_y, P, R, D_y)$
- $p_z = func_z(d_z, K_z)$ or $p_z = func_z(d_z, P, R, D_z)$

$$doping = p_x p_y p_z P \quad 3-21$$

Interpolate Join

$$d = \text{total distance} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad 3-22$$

- $p_x = func_x(d, K_x)$ or $p_x = func_x(d, P, R, D_x)$
- $p_y = func_y(d, K_y)$ or $p_y = func_y(d, P, R, D_y)$
- $p_z = func_z(d, K_z)$ or $p_z = func_z(d, P, R, D_z)$

$$doping = \left(p_x \frac{\Delta x^2}{d^2} + p_y \frac{\Delta y^2}{d^2} + p_z \frac{\Delta z^2}{d^2} \right) P \quad 3-23$$

Miter Join

- Peak Concentration: P
- d_x = X rolloff distance = lateral distance = Δx
- d_y = Y rolloff distance = depth distance = Δy
- d_z = Z rolloff distance = width distance = Δz
- $p_x = func_x(d_x, K_x)$ or $p_x = func_x(d_x, P, R, D_x)$
- $p_y = func_y(d_y, K_y)$ or $p_y = func_y(d_y, P, R, D_y)$
- $p_z = func_z(d_z, K_z)$ or $p_z = func_z(d_z, P, R, D_z)$

If $p_x < p_y$ and $p_x < p_z$ then

$$doping = p_x P \quad 3-24$$

else if $p_y < p_z$ then

$$doping = p_y P \quad 3-25$$

else

$$doping = p_z P \quad 3-26$$

To add these changes to the device, click on **Apply**. Click on **Cancel** if you do not want to add or modify this impurity.

3.8.6: Deleting Impurities

To delete an analytic impurity, select an impurity in the **User Defined Impurity** list on the main panel. Then, select **Impurities**→**Delete**.

3.8.7: Editing Impurities

To edit an analytic impurity, select an impurity in the **User Defined Impurity** list on the main panel. Then, select **Impurities**→**Modify**. The same panel used for adding an impurity will appear with all the filled in values. You can now edit these values.

3.8.8: Combining Impurity Rolloffs

In the special case where the rolloff in one direction along an axis is different from the rolloff in the other direction along the same axis, two impurity specifications must be combined to produce one impurity profile. Special care must be taken when defining these impurity definitions to prevent singularities in the seam between the two impurity definitions. This could cause the impurity to be zero or twice the “Peak concentration” at the seam.

The following is an example of combining two impurity definitions to simulate the surface effect of a doping implant:

1. Enter DEVEDIT and set the work area to 0,-0.25 and 2,2.
2. Create two regions. The first region is silicon with corners at 0,0 2,0, 2,2, and 0,2, and a base of boron of $1e+16 / \text{cm}^3$. See Section 3.2.1: “Adding a Region” for details. The second region is aluminum with corners at 0,0, 1,0, 1,-0.2.

3. Create an impurity distribution to describe the rolloff in the down direction and the lower half in the desired profile.
4. Click on the **Impurities** button at the top of the screen (see Figure 3-18) and set the following parameters:
 - **Impurity:** Arsenic
 - **Start X:** 0
 - **Start Y:** 0.2
 - **End X:** 1
 - **End Y:** 0.2
 - **Peak Concentration:** $1e+20$
 - **Reference Value:** $1e+16$ (base impurity value of silicon)
 - **Rolloff = High Y Rolloff:** Gaussian (Dist)
 - **Distance:** 1 (distance to junction)
 - **Rolloff = Both X Rolloff:** Error Function
 - **Constant:** 0.1
5. Press **Apply**.

Add Impurity

Impurity

Color

Draw Mode Rectangle Line Circle/Ring

Start X: Microns

Start Y: Microns

End X: Microns

End Y: Microns

Peak Concentration:

Reference Value:

Join Function

Y Rolloff: Distance:

X Rolloff: Distance:

Figure 3-18: Downward Impurity Rolloff

6. To add the other half, click on the **Impurities** button (see Figure 3-19) and set the following parameters:
 - **Impurity:** Arsenic
 - **Start X:** 0
 - **Start Y:** 0.2

- **End X:** 1
- **End Y:** 0.2
- **Peak Concentration:** $1e+20$
- **Reference Value:** $1e+16$ (impurity value at surface of silicon)
- **Rolloff = High Y Rolloff:** Gaussian (dist)
- **Distance:** 1 (distance to junction)
- **Rolloff = Both X Rolloff:** Error Function
- **Constant:** 0.1

7. Press **Apply**.

Figure 3-19: Upward Impurity Rolloff

Now, you can create a mesh and save the file as a **SILVACO Standard Structure File** named `example2.str` (perform the steps in Sections 2.2.4: “Mesh Creation” and 2.2.5: “Saving The File”). Figure 3-20 shows the net doping contours on the created device using TONYPLOT. A cut line is made in the vertical direction as shown in Figure 3-21.

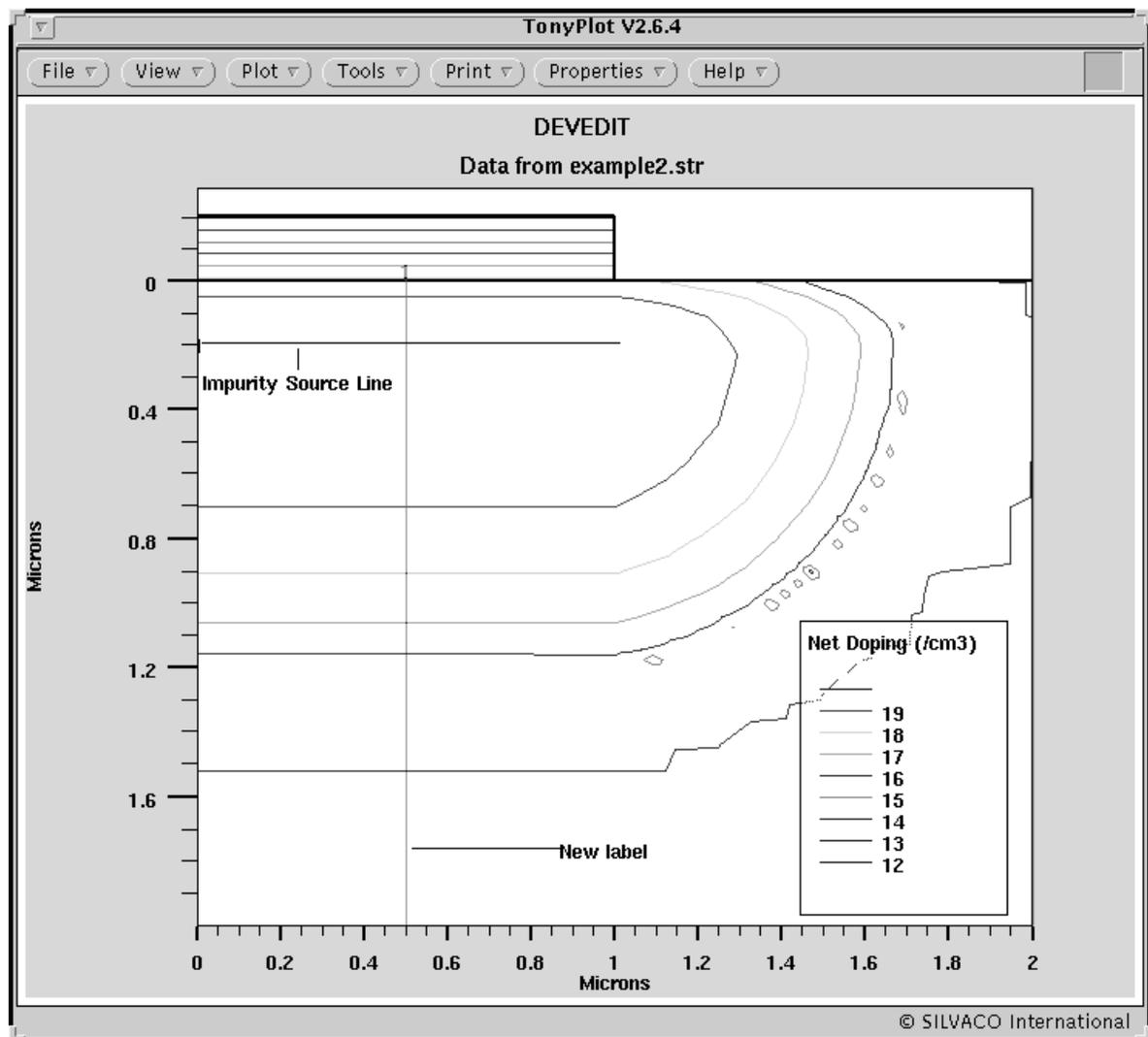


Figure 3-20: Net Doping Contour on Created Device using TonyPlot

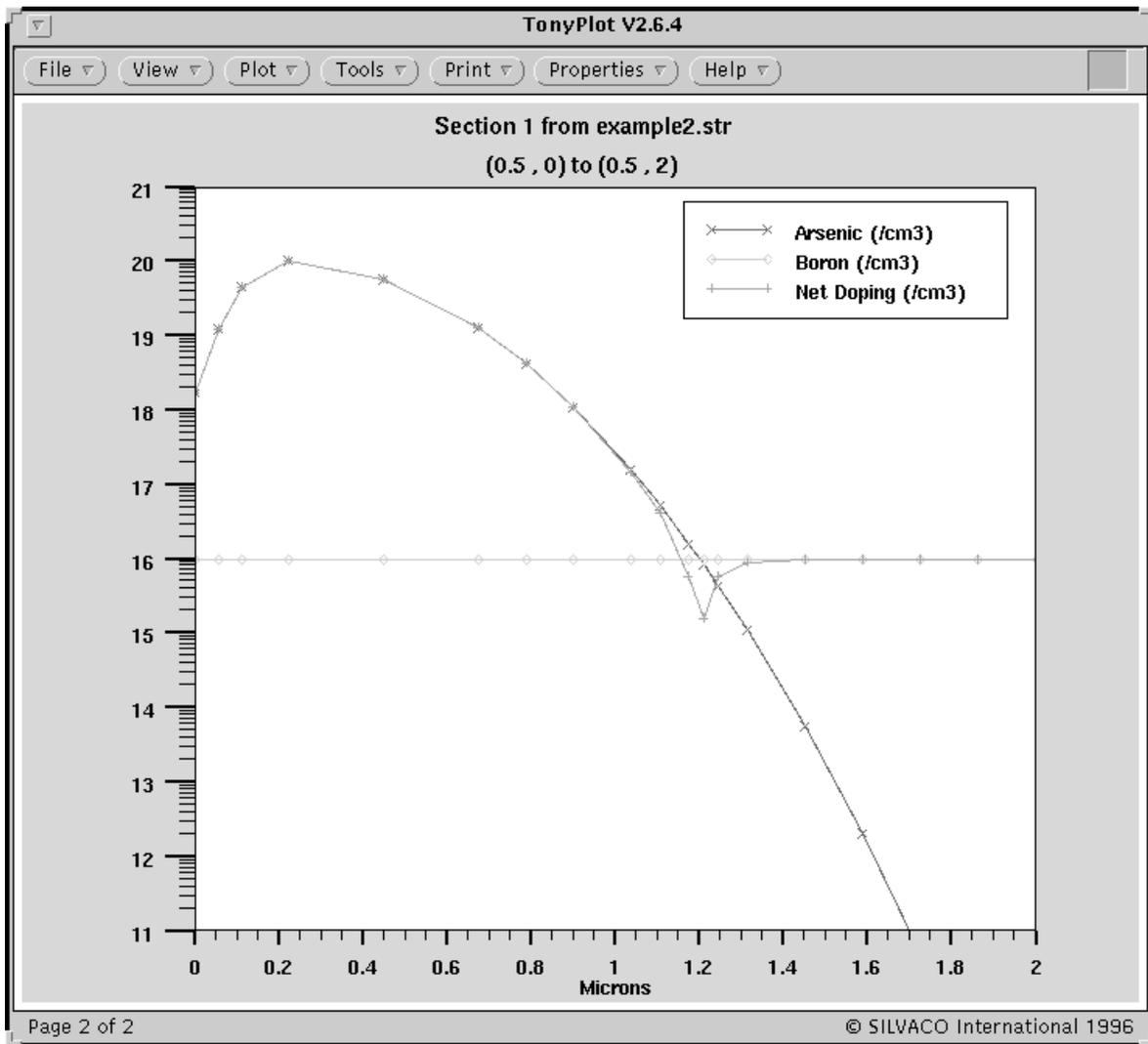


Figure 3-21: Cutline Made in the Vertical Direction

4.1: Overview

This chapter contains a complete description (in alphabetical order) of every statement and parameter used by any of the DEVEDIT products. The following documentation is provided for each statement:

- The statement name
- The syntax of the statement
- A list of all of the parameters of the statement and their type
- A description of each parameter or group of similar parameters
- An example of the correct usage of each statement

4.1.1: Cards And Parameters

Card Syntax

Parameter	Description
PARAM	Required parameter
[PARAM]	Optional parameter
<N>	Integer or floating point number
<C>	User defined string can be quoted by 'or'
<BOOLEAN>	The boolean strings can be true, false, yes, no, on or off. In the case param, param can be used for param=true and !param or ^param can be used for param=false.
<POINT2D>	A 2D point (e.g., 0.5,3.0) note. Points are in Microns.
<POINT2D_LIST>	A list of 2D points (e.g., "0,0 1,0 1,1 0,1 0,0") quotes must be used if more than one point is in list.
{PARAM1 PARAM2}	Parameters are associated with each other.
PARAM1 PARAM2	Supply either param1 or param2 but not both.
PARAM1 PARAM2	Supply param1 or param2 or both.
PARAM...	param repeated one or more times
[PARAM]...	param repeated zero or more times

Examples

In the following example, param1 or param3 must be supplied. If, and only if, param1 is supplied an optional param2 can be supplied. In any case, at least one param4 must be supplied:

```
card {param1=<n> [param2=<c>]} | param3 param4=<n>...
```

Line Continuation

Cards can be continued across multiple lines by ending the line with a backslash (\). In this case, the next line is considered part of the same line.

Note: When loading a command file, lines can also be continued by starting the next line with a plus sign (+). This function has been deprecated and is only supplied for backwards compatibility and can be removed in future versions.

Comments

Comments can be placed at the end of any card or on lines by themselves. Comments start with a number sign (#) and end at the end of the line, regardless whether the line ends with a backslash (\).

Note: Deprecated function: lines starting with \$ are also considered comments. This is for backwards compatibility and can be removed in future releases.

Parameter Section

In the parameter section for each card, each paragraph starts with a parameter's full name and value type. Following this, there can be a list of alternate parameter names. This list is contained in parenthesis. You can use any length abbreviation for a card or parameter as long as it uniquely identifies the card or parameter. A reasonable length, however, should be used to keep names from becoming ambiguous when a newer version is installed.

4.2: BASE.MESH

Starting point when generating a new mesh.

Syntax

```
BASE.MESH [HEIGHT=<N>] [WIDTH=<N>]
```

Description

When a mesh is created with the mesh card and the mode is `mesh.build`, **MeshBuild** creates a base tensor product mesh and refines the geometry, impurities, and mesh constraints. The base mesh may not be regular to allow for the geometry of the device. The original default for width and height is 100000 microns, which means the structure will have 4, 6 or 9 squares in the base mesh, regardless the device size.

Parameters

HEIGHT=<n> (*h,y*): Maximum height (Δy) of each rectangle in the base mesh in microns.

WIDTH=<n> (*w,x*): Maximum width (Δx) of each rectangle in the base mesh in microns.

Replaces Card

```
BaseMesh [Height=<n>] [Width=<n>]
```

See Also

MESH, CONSTRAIN.MESH

4.3: BOUNDARY.CONDITIONING

Reduces the number of boundary points.

Preferred Abbreviation

`bound.cond` or `bnd.cond`

Syntax

```
BOUNDARY.CONDITIONING [ [WHEN=] <C> ] [ [WHEN=] NEVER | ONCE | AUTOMATIC ] \
[ MAXIMUM.SLOPE=<N> ] [ MAXIMUM.RATIO=<N> ] [ ROUNDING.UNIT=<N> ] \
[ LINE.STRAIGHTENING=<N> ] [ ALIGN.POINTS [= <BOOLEAN> ] ]
```

Description

Set or apply boundary conditions to limit complexity of borders between region. This may help limit the number of triangles creating during mesh creation. This may destroy the existing mesh.

Parameters

[WHEN=] never | once | automatic (default=automatic): Specifies when boundary conditions will be performed.

Never: Turns off boundary conditioning.

Once: Performs boundary conditioning now (when card is read).

Automatic: Performs current conditioning and before each mesh command.

MAXIMUM.SLOPE=<n> (max.slope): The maximum ratio of the vertical height to the horizontal width of each boundary segments. If the ratio is greater than this, it is broken into two line segments; one vertical or horizontal and one with this limit.

Note: Vertical and horizontal lines are considered to have a zero ratio, not an infinite one. Therefore, only lines close to being horizontal and vertical lines are affected. This is used to limit the number of triangles `mesh.build` creates. This number must always be less than the maximum triangle ratio (`max.ratio`).

MAXIMUM.RATIO=<n>] (MAX.RATIO): The maximum ratio of a triangles height to its width.

ROUNDING.UNIT=<n> (RND.UNIT, RND): All boundary points are rounded to an even multiple of this unit.

Note: Points created by `mesh.build` are not rounded to this unit but will be strongly affected by the initial boundary point locations.

LINE.STRAIGHTENING=<n> (LINE.STR): If two boundary segments have a joining angle of greater than or equal to 180° `line.straightening`, the two line segments are combined by removing the joining point.

ALIGN.POINTS[=<BOOLEAN>]: If a boundary point joins two almost horizontal lines, you can move the point slightly in the horizontal direction to align it with other points. This is also true in the vertical direction.

Replaces Card

```
BoundaryConditioning [AutoConditioning=<c>] [MaxSlope=<n>] \  
[MaxRatio=<n>] [RoundingUnit=<n>] [LineStraightening=<n>] \  
[AlignPoints[=<boolean>]] [NoSet] [NoApply]
```

See Also

MESH, CONSTRAIN.MESH

4.4: CONSTRAINT.MESH

Sets limits (constraints) on triangles created during mesh and refine operations.

Preferred Abbreviation

`constr.mesh`

Syntax

```
CONSTRAIN.MESH [GLOBAL] [REGION.ID=<N>] [REGION.NAME=<C>] \
[MATERIAL=<C>] [MATERIAL.TYPE=<C>] [x1=<N> y1=<N> x2=<N> y2=<N>] \
[UNDER.REGION=<C> | UNDER.MATERIAL=<C> | UNDER.GATE DEPTH=<N>] \
[DEFAULT] [MAXIMUM.ANGLE[=<N>]] \
[MAXIMUM.RATIO[=<N>]] [MAXIMUM.ADJACENT[=<N>]] \
[MAXIMUM.HEIGHT[=<N>]] [MAXIMUM.WIDTH[=<N>]] \
[MINIMUM.HEIGHT[=<N>]] [MINIMUM.WIDTH[=<N>]]
```

Description

Mesh constraints are used to determine the size and shapes of triangles during the meshing phase. Weaker constraints create a less dense mesh and may be used to improve execution time in subsequent process/device simulations. The constraints are arranged in a hierarchy: global constraints, material type constraints, and region specific constraints. To determine the active constraint in a region, DEVEDIT starts with the global value. You can override this value by a material type constraint, which you can then override by a region specific constraint.

When DEVEDIT is started, all material type constraints default to the global constraints. The global constraints have predetermined default values. As new regions are added, the regions initially default to the material type constraint associated with the regions material. Once you set a value, you can restore it to default to the more general level by using a parameter with no value. For example, `MAX.ANGLE=100` can later be cleared by using `MAX.ANGLE`. All values for the current constraint(s) can be cleared by using the `DEFAULT` parameter (see Figure 4-1).

In addition to this constraint hierarchy, there may be rectangular based constraints. These areas can be specified with use absolute coordinates using `x1`, `y1`, `x2`, and `y2`. The rectangular area can also be implied in a semiconductor area by using `under.region`, `under.material`, or `under.gate`, and setting `depth`.

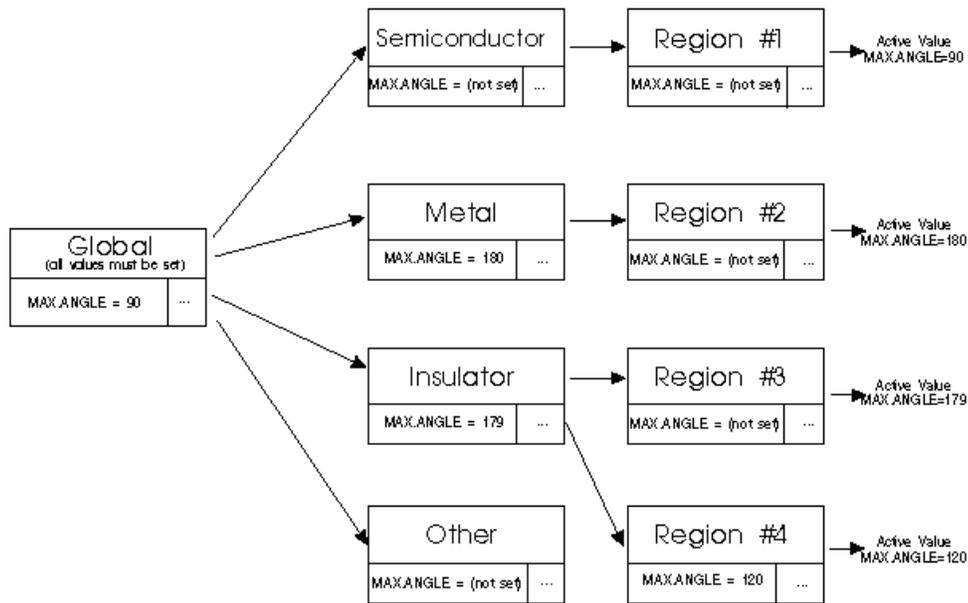


Figure 4-1:Constraint Hierarchy

Parameters

GLOBAL sets global constraints. The global constraints is also if region and material type are NOT used.

REGION.ID=<n> (REG) specifies which region (by region id number) these constraints apply to. Several region parameters can be set at the same time.

REGION.NAME=<c> (REG) specifies which region (by name) these constraints apply to. Several region parameters can be set at the same time.

MATERIAL.TYPE=<c> (MAT.TYPE, TYPE) specifies what material types these constraints apply to. These values can be: **Semiconductor**, **Metal**, **Insulator**, and **Other**.

x1=<N> y1=<N> x2=<N> y2=<N> are the four corners of a rectangular area to which the following constrains will apply.

UNDER.REGION=<C>>: In the semiconductor region under this region, a rectangle is defined from the surface to the depth specified using the `depth` parameter. If the region is disjointed, multiple rectangles can be formed.

UNDER.MATERIAL=<C>: In the semiconductor region under regions of this material, a rectangle is defined from the surface to the depth specified using the `depth` parameter. If multiple regions are made of the same material, multiple rectangles can be formed. These values can be **Poly**, **Aluminum**, and **SiO2**. See Section 4.29: “GENERIC PARAMETER - MATERIAL” for a more complete list.

UNDER.GATE: This is currently the same as “`under.mat=poly`”. It may become more intelligent about finding the gate in future releases.

DEPTH=<N>> specifies how deep (in Microns) the rectangle should be from the surface of the semiconductor region. This parameter must be used in conjunction with `under.region`, `under.material`, or `under.gate`.

DEFAULT: Before setting specified values, reset all values in the specified region, material type, or global constraints to their default values.

MAXIMUM.ANGLE[=<n>] (MAX.ANGLE): Maximum angle a triangle can contain. MAX.ANGLE has a range between 90 and 180°. Normally points are limited by mesh.build to 12 connections. If the maximum angle is set to 180, the connections limit is removed. In no case will an angle actually be 180°.

MAXIMUM.RATIO[=<n>]] (MAX.RATIO): Maximum ratio of a triangles height to its width.

MAXIMUM.HEIGHT[=<n>] (MAX.H): Maximum height of a triangle in microns.

MAXIMUM.WIDTH[=<n>] (MAX.W): Maximum width of a triangle in microns.

MINIMUM.HEIGHT[=<n>] (MIN.H): Triangles shorter than this are not shortened during impurity refinement.

MINIMUM.WIDTH[=<n>] (MIN.W): Triangles narrower than this are not narrowed during impurity refinement.

EXAMPLES

```
# allow all regions to have slightly obtuse triangle
# (sets global constraints) constr.mesh max.angle=100
# allow only non-obtuse triangles in semiconductor regions
constr.mesh mat.type=semiconductor max.angle=90

# allow region #1 to have slightly obtuse triangle even
# if region is a semiconductor.
constr.mesh reg=1 max.angle=95

# now let region #1 default to it material.type or the
# global constraints
const.mesh reg=1 max.angle

# clear all current setting in region #3 and the metal material
# type constraints and set possible triangles to very obtuse
const.mesh reg=3 mat.type=metal default max.angle=180

# Make sure the triangle in the channel of a mos device
# has small enough triangle (assuming the channel is directly
# below a region named "gate").
const.mesh under.region=gate depth=0.5 max.height=0.1 max.width=0.25

# Make sure all contacts have enough connecting point for simulation.
const.mesh under.mat=aluminum depth=0.0001 max.width=0.25
```

Replaces Card

```
ConstrainMesh [Region=<n> | Material=<c> | Type=<c>] \  
  [[!^]MaxAngle[=<n>] [[!^]MaxRatio[=<n>]] \  
  [[!^]MaxAdjacent[=<n>]] [[!^]MaxHeight[=<n>]] \  
  [[!^]MaxWidth[=<n>]] [[!^]MinHeight[=<n>]] \  
  [[!^]MinWidth[=<n>]]
```

See Also

IMPURITY.REFINE, MESH

4.5: CUT

Cuts out a strip from the device and join the two pieces together.

Syntax

```
CUT {{X1=<N> X2=<N>} || {Y1=<N> Y2=<N>}} [AUTOMATIC.JOIN[=<BOOLEAN>]]
```

Description

Cuts out a vertical strip between $x=x1$ and $x=x2$, or cuts out a horizontal strip between $y=y1$ and $y=y2$, or both. Points at $x1/y1$ will not be moved. Points at $x2/y2$ will be moved to $x1/y1$ dragging point beyond $x2/y2$ with them.

Parameters

X1=<n>: Start of x direction cut.

X2=<n>: End of x direction cut.

Y1=<n>: Start of y direction cut.

Y2=<n>: End of y direction cut.

AUTOMATIC.JOIN[=<boolean>] (AUTO.JOIN): If two regions made of the same material presently touch, the regions are joined into one region. The attributes from the region with the lowest ID will be used for the combined region. The default value is true.

4.6: DEPOSIT

Deposits a layer of material.

Syntax

```
DEPOSIT MATERIAL=<C> THICKNESS=<N> [ROUNDING.ANGLE=<N>] \
[[SIDE=]TOP|LEFT|RIGHT|BOTTOM] [START=<N>] [END=<N>] \
[REGION.ID=<N>] [REGION.NAME [COLOR=<N>] [PATTERN=<N>] \
AUTOMATIC.JOIN[=<BOOLEAN>]]
```

Description

Deposits a uniform thickness of a material on the specified side.

Parameters

[SIDE=]TOP|LEFT|RIGHT|BOTTOM: The side on which to deposit the new region. The default is `side=top`.

THICKNESS=<n>: Thickness (in microns) of the deposit.

START=<n>: The start of the deposit. If `side=top` or `side=bottom`, start is an *x* coordinate, otherwise it is a *y* coordinate. Default value is left side of structure or top of structure respectively.

END=<n>: The end of the deposit. If `side=top` or `side=bottom`, start is an *x* coordinate, otherwise it is a *y* coordinate. Default value is right side or bottom of structure respectively.

ROUNDING.ANGLE (RND.ANGLE): When making a corner during a deposit, this angle is used to determine how many points are used. The angle should be between 5 and 45° and be an even divisor of 90. The default angle is 30°.

REGION.ID=<n> (REG.ID): The region ID number of the deposited region if the region `reg` is not joined with an existing region. See `auto.join`.

REGION.NAME=<c> (REG.NAME): The region name of the deposited region if the region `reg` is not joined with an existing region. See `auto.join`.

MATERIAL=<c> specifies which material will be deposited (e.g., Silicon, Aluminum, AlGaAr). See Section 4.29: “GENERIC PARAMETER - MATERIAL” for a more complete description.

AUTOMATIC.JOIN[=<boolean>] (AUTO.JOIN): If two regions made up of the same `auto.join` material presently touch, the regions are joined into one region. Attributes from region with the lowest id will be used for the combined region. The default value is true.

COLOR=<n> (COLOUR): The color used to display region during DEVEDIT in X windows mode. This is a RGB bitmap with eight bits color. DEVEDIT has only a limited subset of these colors. Therefore, the closest match is used. Some basic colors can be specified by name, such as red, green, blue, yellow, cyan, magenta, black, and white. See Section 4.27: “GENERIC PARAMETER - COLOR” for a more complete description.

PATTERN=<n>: The Fill pattern used to display region in DEVEDIT X windows mode. See Section 4.30: “GENERIC PARAMETER - PATTERN” for a more complete description.

4.7: FLIP

Flips (make a mirror image of) the device.

Syntax

```
FLIP X=<N> || Y=<N>
```

Description

Flips (make a mirror image of) the device around a vertical or horizontal line or both.

Parameters

X=<n> flips device around the vertical line $x=<n>$.

Y=<n> flips device around the horizontal line $y=<n>$.

Examples

```
flip x=1
# point 0,0 becomes 2,0
# point 2,2 becomes 0,2
# point 4,4 becomes -2,4
flip y=0

# point 0,0 becomes 0,0
# point 2,2 becomes 2,-2
# point 4,4 becomes 4,-4
flip x=3 y=2

# point 0,0 becomes 6,4
# point 2,2 becomes 4,2
# point 4,4 becomes 2,0
```

4.8: IMPURITY

Adds, replaces or deletes an impurity profile (analytic implant).

Syntax

```

IMPURITY {ID=<N> DELETE [REGION.ID=<N>]} \
  | { [ID=<N>] [REGION.ID=<N>] [IMPURITY=<C>] \
    { [PEAK.VALUE=<N>] | [RESISTIVITY=<N>] } \
    [REFERENCE.VALUE=<N>] [COLOR=<N>] [COMBINATION.FUNCTION=<C>] \
    [ { { [Y1=<N> Y2=<N> ROLLOFF.Y=<C> \
          [ CONCENTRATION.FUNCTION.Y=<C> \
            [ COEFFICIENT.Y=<N> | CONCENTRATION.PARAM.Y=<N> ] \
            [CONCENTRATION.SCALE.FACTOR.Y=<N>] ] } \
        || { X1=<N> X2=<N> ROLLOFF.X=<C> \
          [ CONCENTRATION.FUNCTION.X=<C> \
            [ COEFFICIENT.X=<N> | CONCENTRATION.PARAM.X=<N> ] \
            [CONCENTRATION.SCALE.FACTOR.X=<N>] ] } } \
      | { X=<N> Y=<N> R1=<N> ROLLOFF.R1=<C> \
        [ CONCENTRATION.FUNCTION.R1=<C> \
          [ COEFFICIENT.R1=<N> | CONCENTRATION.PARAM.R1=<N> ] \
          [CONCENTRATION.SCALE.FACTOR.R1=<N>] ] \
        [ R0=<N> ROLLOFF.R0=<C> \
          [ CONCENTRATION.FUNCTION.R0=<C> \
            [ COEFFICIENT.R0=<N> | CONCENTRATION.PARAM.R0=<N> ] \
            [CONCENTRATION.SCALE.FACTOR.R0=<N>] ] ] } \
      | { BASE1=<N>, <N> BASE2=<N>, <N> \
        ROLLOFF.Y=<C> \
          [ CONCENTRATION.FUNCTION.Y=<C> \
            [ COEFFICIENT.Y=<N> | CONCENTRATION.PARAM.Y=<N> ] \
            [CONCENTRATION.SCALE.FACTOR.Y=<N>] ] \
        ROLLOFF.X=<C> \
          [ CONCENTRATION.FUNCTION.X=<C> \
            [ COEFFICIENT.X=<N> | CONCENTRATION.PARAM.X=<N> ] \
            [CONCENTRATION.SCALE.FACTOR.X=<N>] ] } } ] \
    [ { Z1=<N> Z2=<N> } | Z=<N>, <N> \
      ROLLOFF.Z=<C> \
        [ CONCENTRATION.FUNCTION.Z=<C> \
          [ COEFFICIENT.Z=<N> | CONCENTRATION.PARAM.Z=<N> ] \
          [CONCENTRATION.SCALE.FACTOR.Z=<N>] ] ] }

```

Description

Implants (adds) an impurity or quantity to a device by using the impurity card. The model used assumes that a rectangle (or a box in 3-D mode) has a peak value and then rolls off from this peak using a vertical formula (y), a horizontal formula (x), and possibly a width formula (z). These formulae can be abruptly stopped in any of the six directions (up, down, left, right, forward, and backward) by using the rolloff mode. The rolloff mode specifies whether a roll-off function is used or where exactly the impurity is dropped to zero. See Section 3.7: "IMPURITIES" for a more detailed description.

Parameters (Z Parameters are only valid in 3D Mode)

ID=<n> is the identifier of which impurity "implant" should be added, replaced or deleted. If no ID is given, the first unused ID will be added.

Note: There is a list of ID's for all regions and a separate list for each region. In other words, there can be an impurity #1 for all regions, an impurity #1 for region #5 and an impurity #1 for region #7, which makes up three different impurities.

DELETE deletes the impurity "implant" identified by ID.

REGION.ID=<n> is the region number which uniquely identifies the region `reg.id` to which region this impurity applies. If no `region.id` parameter is given (the normal case), it applies to all regions.

REGION.ID=<c> (REG.ID) identifies the region with name `<c>` to which this impurity applies. If two regions have name `<c>`, the region with the lowest region number will be replaced or deleted.

REGION.NAME=<c> (REG.NAME) identifies the region with name `<c>` to which this impurity applies. If multiple regions have name `<c>`, all matched regions are changed.

IMPURITY=<c> specifies what impurity this profile is describing (e.g., Boron, Arsenic, Potential). See Section 4.28: "GENERIC PARAMETER - IMPURITY" for a more complete description.

PEAK.VALUE=<n> is the value of impurity in the base box. When using a 1D profile, the peak value in the profile will be linearly scaled to match this peak value.

REFERENCE.VALUE=<n> (REF.VALUE) is the value of impurity at the given distance. See `concentration.function` and `concentration.coefficient` for more information.

X1=<n>: Left side of base box (In microns).

X2=<n>: Right side of base box (In microns).

Y1=<n>: Top of base box (In microns).

Y2=<n>: Bottom of base box (In microns).

Z1=<n>: Front of base box (In microns).

Z2=<n>: Back of base box (In microns).

BASE.1=<n>,<n> is the "left, top" corner of the peak impurity rectangle. This parameter is depreciated. Use `x1` and `y1` instead.

BASE.2=<n>,<n> is the "right, bottom" corner of the peak impurity rectangle. This parameter is depreciated. Use `x2` and `y2` instead.

COMBINATION.FUNCTION=<c> describes how the `x`, `y` and `z` `comb.funcrolloffs` intersect. The possible values are: multiply, interpolate, or miter.

ROLLOFF.Y=<c>, **ROLLOFF.X=<c>**, and **ROLLOFF.Z=<c>**: The possible values for roll-off functions are:

- both
- high
- high.premature.step (high.p.step)
- low
- step
- step.premature.high (step.p.high)
- low.premature.step (low.p.step)
- step.premature.low (step.p.low)
- premature.step (p.step)

CONCENTRATION.FUNCTION.{Y|X|Z}=<c> (**CONC.FUNC.{Y|X|Z}**): Possible values are shown in the table below.

Warning: These values must match exactly. Do **NOT** abbreviate.

Full Name	Short Name
"Gaussian"	gauss
"Gaussian (Dist)"	gauss.dist
"Error Function"	erfc
"Error Function (Dist)"	erfc.dist
"Linear (Dist)"	dist
"Logarithmic"	log
"Logarithmic (Dist)"	log.dist
"Exponential"	exp
"Exponential (Dist)"	exp.dist
"Step Function"	obsolete use rolloff=step
<1d_profile_name>	

COEFFICIENT.{Y|X|Z}=<n> (**CONC.COEFF.{Y|X|Z}**): Concentration coefficient is valid only if concentration function is NOT a 1D profile. If the concentration function is a distance function, this is the distance (in microns) between the peak.value and the reference.value. Otherwise, it is a function specific coefficient.

CONCENTRATION.PARAMETER.{Y|X|Z}=<c> (**CONC.PARAM.{Y|X|Z}**): If the concentration function is a 1D profile, then it must be one of the following approximation functions. These functions specify the points between data points and at the end of the specified data.

Full Name	Short Name
"Log Extrapolate"	log.ex
"Log Interpolate"	log.in
"Linear Extrapolate"	lin.ex
"Linear Interpolate"	lin.in

If the concentration function is not a 1D profile, then `CONCENTRATION.PARAMETER.{Y|X|Z}` is an alias for `COEFFICIENT.{Y|X|Z}`.

CONCENTRATION.SCALE.FACTOR.{Y|X|Z}=<n> (**CONC.SCALE.{Y|X|Z}**): Concentration scale factor is valid only if concentration function is a 1D profile. This allows the rolloff to be shortened (value <1.0) or lengthened (value >1.0) using this linear scalar factor. The default value is 1.0.

COLOR=<n>: The color used to display region during DEVEDIT in X color windows mode. This is an RGB bitmap with eight bits color per color. DEVEDIT has only a limited subset of these colors so the closest match is used. Some basic colors can be specified by name, such as red, green, blue, yellow, cyan, magenta, black, and white. See Section 4.27: "GENERIC PARAMETER - COLOR" for a more complete description.

Replaces Card

```
AddImpurity [ID=<n>] [Region=<n>] [Color=<n>] Base1=<point2d> Base2=<point2d>
[PeakValue=<n>] \ [ReferenceValue=<n> | ContourValue=<n>] \
[{Z1=<n> Z2=<n>} | Z=<n>,<n>] \ CombinationFunction=<c> \
Rolloff1=<c> ConcentrationFunction1=<c> \ Coefficient1=<n> | CParam1=<n>
[CRatio1=<n>] \ Rolloff2=<c> ConcentrationFunction2=<c> \
Coefficient2=<n> | CParam2=<n> [CRatio2=<n>] \
Rolloff3=<c> ConcentrationFunction3=<c> \ Coefficient3=<n> | CParam3=<n>
[CRatio3=<n>]
```

See Also

PROFILE

4.9: IMPURITY REFINE

Sets the limit on the impurity differential across triangles.

Syntax

```
IMPURITY.REFINE IMPURITY=<C> [ID=<N>] SENSITIVITY=<N> \
[scale=<C>] [transition=<N>]
```

or

```
IMPURITY.REFINE MINIMUM.SPACING=<N> | Z=<N>
```

or

```
IMPURITY.REFINE ID=<N> DELETE
```

Preferred Abbreviation

`imp.ref`

Description

These values are used by the meshing routine to determine if triangles are small enough. When the mesh card is run, the current sensitivity of each impurity is tested. If the impurity difference across the triangle is greater than sensitivity, the triangle is broken into smaller triangles.

Parameters

ID=<n< deletes or modifies an exist refinement. If no ID is supplied, the first unused ID (starting from 1) is used.

DELETE deletes the impurity refinement identified by parameter ID.

IMPURITY=<c> specifies what quantity/impurity is being refined (e.g., Boron, Arsenic, and Potential). See Section 4.28: “GENERIC PARAMETER - IMPURITY” for a more complete description.

SENSITIVITY=<n>: If an impurity’s value changes more than sensitivity, smaller triangles are created. If an impurity’s scale is logarithmic, sensitivity is in powers of ten. Impurities are really extrapolated using arc hyperbolic sine (not using logarithms) and then normalized. There should be no noticeable difference on values greater than 10 times the transition value (one level of sensitivity).

SCALE=<C> specifies which scale the sensitivity should use. Different impurities (quantities) have different default scales. See Section 4.28: “GENERIC PARAMETER - IMPURITY” for default values.

- **Linear** uses linear scale.
- **Logarithmic (log)** is an alias for `arc.hyperbolic.sine`.
- **arc.hyperbolic.sine (arc.h.sine)** uses arc hyperbolic sine scale, which is similar to a logarithmic scale.

TRANSITION=<N>: This value is used to modify the `arc.hyperbolic.sine (log)` scale. Values below this value are considered insignificant. Different impurities (quantities) have different default transition values. See Section 4.28: “GENERIC PARAMETER - IMPURITY” for default values.

MINIMUM.SPACING=<n> (MIN.SPAC): If a triangle is narrower than this, it is not narrowed further. If a triangle is shorter than this, it is not shortened. This parameter applies to all impurities currently being refined, not just the impurity specified by this card.

3-D Parameters

Z=<n> refines at the specified z plane

Replaces Card

```
ImpRefine[Type=<c> Value=<n> | Sensitivity=<n>] MinSpacing=<n> Z=<n>
```

See Also

MESH

4.10: INITIALIZE

Clears existing device and load file.

Syntax

```
INIT INFILE=<C> [Z=<POINT2D> | {Z1=<N> Z2=<N>}] MESH[=<BOOLEAN>]
LOAD FILE.NAME=<C> [[TYPE=] <C>] [Z=<POINT2D> | {Z1=<N> Z2=<N>}] MESH[=<BOOLEAN>]
```

Parameters

FILE.NAME=<c> (**file**, **infile**, **inf**): The file name of a SILVACO standard structure file or a DEVEDIT command file.

TYPE=<c> overrides the automatic file type recognition and loads file as the specified type.

SILVACO standard(mas): SILVACO standard structure file.

structure(str): SILVACO standard structure file.

card.deck(deck): DEVEDIT command file.

MESH[=<boolean>]: If mesh is set to false, any mesh commands are ignored and structure files are loaded without their mesh. This can greatly speed up load time. The old mesh is not needed if you are trying to remesh a device. The default is mesh=true (accept mesh commands and load structure file meshes).

3-D Parameters

Z1=<n>: If a 2D region is loaded, convert to a 3D region, using z1 as the starting z plane.

Z2=<n>: If a 2D region is loaded, convert to a 3D region, using z2 as the ending z plane.

Z=<n>,<n> z1,z2 as one parameter.

Replaces Card

```
LoadFile FileName=<c> [Type=<c>] [Z=<point2d> | {Z1=<n> Z2=<n>}]
```

See Also

STRUCTURE

4.11: JOIN

Joins two devices together.

Syntax

```
JOIN [ [SIDE=]RIGHT|LEFT|TOP|BOTTOM] FILE.NAME=<C> \
      [ADJUST=<N> | SURFACE.ALIGN[=<BOOLEAN>] ] [SPACER.THICKNESS=<N> \
      [SPACER.MATERIAL=<C>] ] [MIRROR[=<BOOLEAN>] [AUTOMATIC.JOIN[=<BOOLEAN>] ] ]
```

Description

Combines the device currently loaded in DEVEDIT and a device stored in a file into one device.

Parameters

[SIDE=]RIGHT|LEFT|TOP|BOTTOM: The side on which to join the new structure onto. The default is `side=right`.

FILE.NAME=<c>: The filename containing the device to be joined in the DEVEDIT command format or the SILVACO standard structure file format.

ADJUST=<n>: If `side=right` or `side=left`, adjust the device down (or up if negative) a specified amount before performing join. If `side=top` or `side=bottom`, adjust the device to the right (or to the left if negative) the specified amount before performing join.

SURFACE.ALIGN[=<boolean>]: If true perform the necessary adjust `surf.align` to have the semiconductor regions in both devices aligned at the top or left side of the joined device. The default is `surface.align=false`.

SPACER.MATERIAL=<c> (sp.mat) specifies which material joins the two devices (e.g., Silicon, Aluminum, AlGaAr). See Section 4.29: "GENERIC PARAMETER - MATERIAL" for a more complete list.

SPACER.THICKNESS=<n>: The thickness of the spacer inserted between the two devices.

MIRROR[=<boolean>] (FLIP) takes a mirror image of the new device before merging it into the existing device. The default is not to mirror the new device.

AUTOMATIC.JOIN[=<boolean>](AUTO.JOIN): If two regions made up of the same material presently touch, the regions will be joined into one region. The attributes from the region with the lowest ID is used for the combined region. The default value is true.

Note: Currently, the spacer parameters were not ready for release and cannot be included in Version 2.0.0.

4.12: MESH

Creates new mesh using previous set parameters.

Syntax

```
MESH [ [MODE=] MESH.BUILD | TENSOR.PRODUCT | DELETE ]
```

Description

When the mesh card is used, the following steps are performed.

1. Deletes any existing mesh.
2. Performs automatic boundary conditioning (if set). See Section 4.3: “BOUNDARY.CONDITIONING” for more details.
3. Creates a base mesh. In `mesh.build` mode, creates a tensor product mesh using `base.mesh` parameters. See Section 4.2: “BASE.MESH” for more details. In `tensor.product` mode, it creates a tensor product mesh using all currently existing boundary points, those remaining after boundary conditioning.
4. Refines on geometry. Any points not part of the base mesh are now handled.
5. Refines on impurities. See Section 4.9: “IMPURITY REFINE” for more details.
6. Refine on mesh constrains. See Section 4.4: “CONSTRAINT.MESH” for more details.

Parameters

```
[mode=]mesh.build|tensor.product
```

Examples

```
# build mesh using mesh build algorithm.
mesh
mesh mode=mesh.build
mesh mesh.build

# build a first level tensor product mesh.
mesh tensor.prod

# delete existing mesh
mesh del
```

See Also

```
BASE.MESH, BOUNDARY.CONDITIONING,
IMPURITY.REFINE, CONSTRAIN.MESH
```

4.13: MIRROR

Mirrors the device.

Syntax

```
MIRROR [ [SIDE=]RIGHT|LEFT|TOP|BOTTOM] [AUTOMATIC.SPLIT[=<BOOLEAN>]]
```

Description

Mirrors the device by creating a mirror image of the device in the given direction and joining the image to that side.

Parameters

[SIDE=]RIGHT|LEFT|TOP|BOTTOM: The direction to mirror the device. The default direction is right.

AUTOMATIC.SPLIT[=<boolean>] (AUTO.SPLIT): If the region does not touch the mirrored edge, the region will be split into two discrete regions. If `auto.split=false`, the region is disjointed but remains one region. The default is `auto.split=true`.

4.14: MOVE

Moves the device around in the work area.

Syntax

```
MOVE{ [DIRECTION=] {RIGHT|LEFT|UP|DOWN} DISTANCE=<N> } \
|| { [X.ADJUST=<N>] [Y.ADJUST=<N>] }
```

Description

Moves the device around in the work area. Does not change the device at all, only its relative position in space.

Parameters

[DIRECTION=]{RIGHT|LEFT|UP|DOWN} (DIR): The direction to move the device. If this parameter is used, a distance parameter must also be supplied. The default direction is right.

DISTANCE=<n> (DIST): The distance (in microns) to move the device in the specified direction.

X.ADJUST=<n> (X.ADJ) moves the x coordinates of all point by adding x.adjust. x.adjust may be negative.

Y.ADJUST=<n> (Y.ADJ) moves the y coordinates of all point by adding y.adjust. y.adjust may be negative.

Examples

```
move dir=right dist=3
# point 0,0 becomes 3,0

move left dist=3
# point 0,0 becomes -3,0

move y.adj=3
# point 0,0 becomes 0,3

move x.adj=-3 y.adj=3
# point 0,0 becomes -3,3
```

4.15: PROFILE

Adds, replaces, or deletes a 1D profile.

Syntax

```
ROFILE {NAME=<C> DELETE} \
| {[NAME=<C>] IMPURITY=<C> FILE.NAME=<C> | INFILE=<C> | DATA.POINT=<POINT_2D>}
```

Description

Sets up a 1D doping profile that can later be used to distribute an impurity using the impurity card.

Parameters

NAME=<c> is used to identify a 1D profile. This field is required in delete mode. If no name is given and delete is not specified, a new name in the form "NewProfile000" is used. There are a limited set of names that should not be used. See Section 4.8: "IMPURITY" for more details.

DELETE deletes the profile identified by the name parameter.

IMPURITY=<c> specifies which impurity this profile is describing. `imp` (e.g., Boron, Arsenic, Potential). See Section 4.28: "GENERIC PARAMETER - IMPURITY" for a more complete description.

FILE.NAME=<c>: The file name of a SUPREM3 or a 1D SUPREM4 SILVACO Standard Structure file. The selected impurity is extracted in file and the distance from the surface of the semiconductor `inf` and the value at that location are stored. Only the data points in the semiconductor are stored.

DATA. POINT=<POINT_2D>: Each data point consists of a distance from `d.p` the peak value (surface) and the value (concentration) at that location.

Examples

```
# Load an arsenic profile from a suprem3 file
profile file=suprem3.str name="Arsenic Profile" imp=arsenic

# Make a phosphorous profile
profile name=PhosProfile imp=phosphorous d.p=0,7e19 d.p=0.2,1e20 \
d.p=1,1e19 d.p=2,1e18 d.p=3,1e17

# delete the arsenic
profile delete name="Arsenic Profile"
```

Replaces Card

```
Profile [Name=<c>] Impurity=<impurity> \ FileName=<c> |
DataPoint=<point_2d>...
```

See Also

IMPURITY

4.16: QUIT

Exits DEVEDIT or ends reading file.

Syntax

BYE, END, EXIT, QUIT

Description

Any one of these four cards cause DEVEDIT to exit or, if reading a file, the remainder of the file is ignored.

4.17: REFINE

Manually refines existing mesh.

Syntax

```
REFINE [DIRECTION=]X|Y|BOTH|UNREFINE \
{X1=<N> Y1=<N> X2=<N> Y2=<N>} \
| {POINT.1=<N>,<N> POINT.2=<N>,<N>} \
| {LEFT=<N> TOP=<N> RIGHT=<N> BOTTOM=<N>}
```

Description

More (less if unrefine) triangles are created in the x, y, or both directions in the specified rectangle. A mesh must currently exist that was not loaded from a structure file.

Parameters

[DIRECTION=]x|y|both|unrefine (DIR) refines direction.

x: More triangles horizontally. (~twice as many)

y: More triangles vertically. (~twice as many)

both: More triangles. (~four times as many)

unrefine: Less triangles.

X1=<n> (LEFT): Left side of box to (un)refine in microns.

Y1=<n> (TOP): Top of box to (un)refine in microns.

X2=<n> (RIGHT): Right side of box to (un)refine in microns.

Y2=<n> (BOTTOM): Bottom side of box to (un)refine in microns.

POINT.1=<n>,<n>X1,Y1 (P1) as one parameter.

POINT.2=<n>,<n>X2,Y2 (P2) as one parameter.

Replaces Existing Card

```
Refine Mode=<C> P1=<POINT2D> P2=<POINT2D>
```

See Also

MESH

4.18: REGION

Adds, replaces or deletes a region.

Syntax

```
REGION {DELETE {ID=<n> | ID=<c> | NAME=STRING} \
| {{ID=<n>} [NAME=<c>]} | ID=<c>} [MATERIAL=<c>] \
[COLOR=<n>] [PATTERN=<n>] POINTS=<point2d_list> \
[WORK.FUNCTION=<n>] [ELECTRODE.ID[=<n>]] [Z1=<n>] [Z2=<n>]}
```

Preferred Abbreviation

REG

Description

A DEVEDIT device is made up of regions. Each region has a unique region ID number. Regions also can have a name. Several regions can have the same name. Region names should not start with a number. Some places accept a region ID number or a region name and will assume a number is a region ID number.

Regions are made of a material and contain a list of non-intersecting polygons. A region can consist of several discrete polygons. If one polygon is contained in another polygon, it is considered a hole in the containing polygon. If a polygon is inside a “hole” polygon, it describes a region with material and a hole in the hole.

For display purposes (inside DEVEDIT only), a region can contain a color and a fill pattern. This is only used when DEVEDIT opens an X display window.

A region can also be an electrode. In that case, you can either supply the electrode ID number or choose one from DEVEDIT. For future enhancement to simulators, you must also supply a work function.

Parameters

DELETE deletes the region identified by `id=` or `name=`. You should only use one parameter (`id=` or `name=`). If `id` is a number, delete the region with that number. If `id` is a string, delete the region name with the lowest region number. If `name=` is used, delete all region names.

ID=<n> is a region number that uniquely identifies the region to be inserted, replaced or deleted.

ID=<c> identifies the region with name `<c>` to be replaced or deleted. If no region has name `<c>`, insert a new region with name `<c>` and the lowest unused region number. If two regions have name `<c>`, the region with the lowest region number will be replaced or deleted.

NAME=<c> deletes all regions with name `<c>` when in delete mode. Otherwise, the region being replaced or inserted will be given the name `<c>`.

MATERIAL=<c> specifies what material the region is made of (e.g., Silicon, Aluminum, or AlGaAr). See Section 4.29: “GENERIC PARAMETER - MATERIAL” for a more complete description.

COLOR=<n> (**COLOUR**) displays region during DEVEDIT in X windows mode. This is an RGB bitmap with eight bits per color. DEVEDIT has only a limited subset of these colors. Therefore, the closest match is used. Some basic colors can be specified by name, such as, red, green, blue, yellow, cyan, magenta, black, and white. See Section 4.27: “GENERIC PARAMETER - COLOR” for a more complete description.

PATTERN=<n> is used as a fill pattern for display region in DEVEDIT in X windows mode. See Section 4.30: “GENERIC PARAMETER - PATTERN” for a more complete description.

POINTS=<point_2d_list> is used for location of points making polygons that describe the region.

ELECTRODE.ID[=<n>] (ELEC.ID) describes the region as an electrode, setting the electrode number to <n>. If <n> is not supplied, the lowest unused electrode ID number will be used.

WORK.FUNCTION=<n> used only if `electrode.id` is set to define work function for materials. This is not currently being used by any simulators but may be used in future releases.

3D Parameters

Z1=<n> is the Z plane where region starts (in microns) for the 3D Mode.

Z2=<n> is the z plane where region ends (in microns) for the 3D Mode.

Examples

```
# ADD A 10 BY 10 MICRON BLOCK OF SILICON AS REGION #1
# WITH THE NAME "WAFFER" DISPLAYED IN RED
REGION ID=1 NAME=WAFFER MATERIAL=SILICON COLOR=0X30 \
POINTS="0,0 10,0 10,10 0,10 0,0"

# ADD AN ALUMINUM CONTACT AS ELECTRODE #1 USING THE NEXT
# AVAILABLE REGION NUMBER WITH THE NAME "SOURCE"
# USING DEFAULT COLOR.
REGION NAME=SOURCE MATERIAL=AL ELEC.ID=1 WORK.FUNC=4.28 \
POINTS="0,-1 1,-1 1,0 0,0 0,-1"

# SET REGION #3 TO ELECTRODE #2
REGION NAME=SOURCE MATERIAL=AL ELEC WORK.FUNC=4.28 \
POINTS="9,-1 10,-1 10,0 9,0 9,-1"

# DELETE REGION #2
REGION DELETE ID=2

# DELETE THE FIRST REGION NAMED SOURCE
REGION DELETE ID=SOURCE

# DELETE ALL REGIONS NAMED SOURCE
REGION DELETE NAME=SOURCE
```

Replaces Cards

```
AddRegion [Region=<n>] [Name=<c>] [Electrode=<n>] \
[Material=<c>] [Color=<n>] [Pattern=<n>] \
Points=<point2d_list> [WorkFunction=<n>] [Z1=<n>] [Z2=<n>]
DeleteRegion Region=<n>
```

4.19: RENUMBER.REGIONS

Renumbers all regions to fill in gaps or specifically rennumbers one region.

Syntax

```
RENUMBER.REGIONS [FROM=<n> TO=<n>]
```

Parameters

FROM=<n> region ID of region to be renumbered.

TO=<n> region's new Region ID.

4.20: SOURCE

Runs commands stored in file (on existing device).

Syntax

```
SOURCE FILE.NAME=<C> [Z1=<N> Z2=<N>] MESH[=<BOOLEAN>]
```

Parameters

FILE.NAME (FILE) runs DEVEDIT cards contained in the specified file.

MESH=<boolean>: If mesh is set to false, any mesh commands are ignored. The default is mesh=true (accept mesh commands).

3-D Parameters

Z1=<N>: If a 2D region card is read, convert to a 3D region, using z1 as the starting z plane.

Z2=<N>: If a 2D region card is read, convert to a 3D region, using z2 as the ending z plane.

Replaces Card

```
IncludeFile FileName=<c> [Type=<c>] [Z=<point2d>|{Z1=<n> Z2=<n>}]
```

4.21: STRETCH

Stretches the device.

Syntax

```
STRETCH { X.VAL=<N> | Y.VAL=<N> STRETCH.VALUE=<N> }
| { [REGION.ID=<C> | MATERIAL=<C>] LENGTH=<N> \
| STRETCH.VALUE=<N> Y.LENGTH=<N> | Y.STRETCH.VALUE=<N> \
CENTER [=<BOOLEAN>] } \
| { X1=<N> X2=<N> STRETCH.VALUE=<N> | LENGTH=<N> } \
| { Y1=<N> Y2=<N> STRETCH.VALUE=<N> | LENGTH=<N> }
```

Description

STRETCH provides many ways to make the device longer or taller or even narrower or shorter. There are many parameters that can be used in many combinations, however, most of these parameters are only for the most advanced user. Most users should be able to look at the first five examples below and learn all they need about STRETCH.

STRETCH allows a vertical or horizontal line to be stretched out a specified amount or allows a range to be evenly stretched.

The line to be stretched can be a specific location ($x=1$). Alternatively, the line can be the center of a region, specified by the region's name, ID or material. If a line is to be stretched into an area, the width of the new area must be supplied. A `stretch.value` can be supplied to specify a new length. Alternatively, you can determine a new length for the specified region. That is, the amount stretched equals the desired length minus the original length of the region.

To stretch an area to cover an even larger area, an area can be specified by region name, region ID, material name, $x1/x2$ pair, or an $y1/y2$ pair. If a region is specified, the center option must be set to false. This area can be given a new length (using `length`) or an extra amount (using `stretch.value`) to be added to the length.

Length and `stretch.value` are normally assumed to apply to the x direction, unless `y.val` or $y1/y2$ are used. `y.length` or `y.stretch.value` can be used to force the stretch to be in the y direction. Mixed parameters, such as `x.val` and `y.stretch.value`, should not be used together.

In general, it is only useful to do line type stretches in areas where the impurities are fairly constant, like the center of the gate on a MOSFET transistor. This allows a gate's length to be quickly changed for multiple simulations. This is described in the first five examples below.

Parameters

LENGTH=<n> (LEN, X.LENGTH, X.LEN): This is the new x length of the specified region, material, $x1/x2$ area, or $y1/y2$ area. If none of these are specified, the first region made of poly silicon is used. If no such region exists, it is an error. Make sure not to shrink a region. A warning will appear if the region shrinks. If `y.val` or $y1/y2$ are used, this is the same as `y.length`.

Y.LENGTH=<n> (Y.LEN): This is the new y length of the specified region, material, or $y1/y2$ area. If none of these are specified, the first region made of poly silicon is used. If no such region exists, it is an error. Make sure not to shrink a region. A warning will appear if the region shrinks.

REGION.ID=<c>: This is the name or ID number of the region used to identify the stretch line at the center of this region. If `center=false`, then the whole region is stretched.

MATERIAL=<c>: The first region found made of this material is used to identify the stretch line at the center of that region. If `center=false`, then the whole region is stretched. See Section 4.29: "GENERIC PARAMETER - MATERIAL" for a list of materials.

X.VAL=<n> (X) stretches from the vertical line `x=x.val`. `stretch.value` must be supplied with this parameter.

Y.VAL=<n> (Y) stretches from the vertical line `x=x.val`. `stretch.value` must be supplied with this parameter.

STRETCH.VALUE=<n> (STR.VAL, X.STR.VAL) stretches the device much longer. If you use the `x.val` or `y.val` parameters, the stretch expands that line to the new width. If `x1` and `x2` or `y1` and `y2` parameters are used, the stretch is distributed throughout the range given.

Y.STRETCH.VALUE=<n> (Y.STR.VAL) is the same as `stretch.value` except stretch in the y direction. If you use `y.val` or `y1/y2`, `stretch.value` will also be in the y direction.

CENTER[=<boolean>] causes any stretch to happen at the center of the specified region. `center=false` causes the stretch to be distributed over the region. `center` is defaulted to true unless `x1/x2` or `y1/y2` parameters are used, in which case, it defaults to false.

X1=<n>: Start of x direction stretch. Use `stretch.value` or `length` with this parameter.

X2=<n>: End of x direction stretch. Use `stretch.value` or `length` with this parameter.

Y1=<n>: Start of y direction stretch. Use `stretch.value` or `y.length` with this parameter.

Y2=<n>: End of y direction stretch. Use `stretch.value` or `y.length` with this parameter.

Examples

```
# cause the device to be stretched at the center of first
# polysilicon region so that the new length of that region
# will be 1.5 microns in the x direction.
stretch length=1.5
```

```
# cause the device to be stretched at the center of first
# polysilicon region so that the new length of that region
# will be an extra 1.5 microns in the x direction.
stretch stretch.value=1.5
```

```
# cause the device to be stretched at the center of first
# polysilicon region so that the new length of that region
# will be 1.5 microns in the x direction.
stretch material=polysilicon length=1.5
```

```
# cause the device to be stretched at the center of region #3
# so that the new length of region #3 will be 1.5 microns in the # x direction.
stretch reg=3 length=1.5
```

```
# cause the device to be stretched at the center of the region
```

```
# named "gate" so that the new length of that region will be
# 1.5 microns in the x direction.
stretch reg=gate length=1.5

# cause the device to be stretch in the x direction so that new
# length of the region named "gate" will be 1.5 microns. The
# extra area will be evenly distribute across the whole gate.*
stretch reg=gate ^center length=1.5

# cause the region named "gate" to be stretch to 1.5 microns
# in the y direction. The device is expanded at the center of the gate.
stretch reg=gate y.length=1.5

# add an extra.5 microns in depth at y=2.0
stretch y.val=2.0 stretch.value=.5

# stretch out the doping profile between 2 microns in depth
# and 3 microns in depth to take 3 microns instead of 1 micron.
stretch y1=2.0 y2=3.0 length=3.0
```

Note: ^center is the same as center=false as described in Section 4.26: "GENERIC PARAMETER - BOOLEAN TYPE".

4.22: STRUCTURE

Saves current structure to a file.

Syntax

```
STRUCTURE OUTFILE=<C> # TYPE=MASTER (DEFAULT)
SAVE FILE.NAME=<C> [TYPE=<C>] \ # TYPE=CARD.DECK (DEFAULT)
[ [TYPE=]MASTER|STRUCTURE|CARD.DECK]
```

Parameters

FILE.NAME=<c> (FILE, OUTFILE, OUTF): The file name used to store a SILVACO Standard Structure file or a DEVEDIT command file. The structure card only stores SILVACO Standard Structure files.

TYPE=<c>: The type of file to store. The possible values are:

- **MASTER(MAS):** SILVACO standard structure file.
- **STRUCTURE(STR):** SILVACO standard structure file.
- **CARD.DECK(DECK):** DEVEDIT command file.

3-D Parameters

TYPE=<c>: In 3D mode, structure files are normally saved with prismatic elements. mode=tetrahedrons (or mode=tet) can be used to output tetrahedral elements into a structure file.

Replaces Card

```
SaveFile FileName=<c> [Type=<c>]
```

4.23: SUBSTRATE

Special substrate electrode.

Syntax

```
SUBSTRATE DELETE \
| { [NAME=<C>] [ELECTRODE.ID=<N>] [WORK.FUNCTION=<N>] [APPLY [=<BOOLEAN>]] }
```

Preferred Abbreviation

SUBSTR

Description

The substrate is a special region with no thickness that is placed at the bottom of the device. The substrate is always considered to be an electrode.

Parameters

DELETE deletes existing substrate.

NAME=<c>: The name of the substrate electrode (default=substrate).

ELECTRODE.ID=<n> (**elec.id**) describes region as an electrode setting the electrode `elec` number to `<n>`. If `<n>` is not supplied, the lowest used electrode ID number will be used.

WORK.FUNCTION=<n> (**work.func**): Used only if `electrode.id` is used to set work function for materials. This is not currently used by any simulators, however, may be used in future releases.

APPLY[=<boolean>] (for internal use only: default=true): If `apply=false`, only set parameter in the substrate panel.

Examples

```
# Make the substrate electrode #3 and named collector.
substr name=collector elec.id=3 work.func=4.28
```

Replaces Card

```
Substrate Delete \
| { [Name=<c>] [Electrode=<n>] \
[WorkFunction=<n>] [NoApply[=<boolean>]] }
```

4.24: WORK.AREA

Sets viewing area of canvas.

Syntax

```
WORK.AREA { [X1=<N>] [X2=<N>] [Y1=<N>] [Y2=<N>] } \
  { [LEFT=<N>] [RIGHT=<N>] [TOP=<N>] [BOTTOM=<N>] } \
  | { [POINT.1=<POINT2D>] [POINT.2=<POINT2D>] }
```

Description

Sets the viewing area seen in the main window of DEVEDIT (X windows mode only).

Parameters

X1=<n> (LEFT): Minimum *x* value of draw area.

X2=<n> (RIGHT): Maximum *x* value of draw area.

Y1=<n> (TOP): Minimum *y* value of draw area.

Y2=<n> (BOTTOM): Maximum *y* value of draw area.

POINT.1=<n>,<n> (P1): *x1*, *y1* as a single parameter

POINT.2=<n>,<n> (P2): *x2*, *y2* as a single parameter

Replaces Card

```
WorkArea P1=<point2d> P2=<point2d>
```

4.25: Z.PLANES

Defines z planes (3D mode only).

Syntax

```
Z . PLANE [ { Z=<N> DELETE } | Z=<N> [ SPACING=<N> ] ] [ MAXIMUM.SPACING=<N> ] \
[ MAXIMUM.RATIO=<N> ]
```

Description

Z planes are created at the front and back of all regions and at the specified z locations. Additional z planes are added to meet the requirements of SPACING, MAXIMUM.SPACING, and MAXIMUM.RATIO.

Parameters

Z=<n>: A Z plane that must exist or is to be deleted.

DELETE deletes the specified z plane and any associated spacing.

Note: If a z plane at the start or end of a region is deleted, only the spacing associated with that z plane is deleted. The Z plane will still be made part of the structure.

SPACING (SPAC): The spacing around the specified z plane will be restricted to the specified spacing. spacing=0 means there are no restrictions specific to this z plane. spacing=0 is the default case.

MAXIMUM.SPACING=<n> (MAX.SPAC): This is the maximum gap (in microns) between two adjacent z planes.

MAXIMUM.RATIO=<n> (MAX.RATIO): This is the maximum ratio of the gap between two adjacent z planes and the gap to the next z plane.

Replaces Card

```
ZPlane Z=<n> [MaxSpacing=<n>] [MaxRatio=<n>]
```

4.26: GENERIC PARAMETER - BOOLEAN TYPE

Parameters described like `param[=<boolean>]` are boolean parameters, meaning they can be either true or false.

There are several ways to set the parameter to true. They are:

- `param=true`
- `param #just listing the param sets it to true`
- `param=on`
- `param=1`
- `param=yes`

There are several ways to set the parameter to false. They are:

- `param=false`
- `! param`
- `^ param`
- `param=off`
- `param=0`
- `param=no`

The default value for a boolean parameter type is dependent on the specific parameter.

4.27: GENERIC PARAMETER - COLOR

In X window mode, regions and objects can have a color assigned to them. Regions have a default color base on their material. The card, however, with color parameters can override these colors.

The color can be set using a standard 24 bit number with eight bits for each color component, red, green, and blue. The easiest way to use this is use hexadecimal values by starting the number with "0x". Each component than has a range from 00 (no color) to FF (full color). See the example below.

```
color=0xFF0000# Full red
color=0x00FF00# Full green
color=0x0000FF# Full blue
```

Not all color combinations are supported by DEVEDIT. The closest color support by DEVEDIT is used. The eight primary colors can be set by name (e.g., color=red).

Color Name	Color Value
black	0x000000
red	0xFF0000
green	0x00FF00
blue	0x0000FF
cyan	0x00FFFF
yellow	0xFFFF00
magenta	0xFF00FF
white	0xFFFFFFFF

In hexadecimal, digits are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Note: 0x00 < 0x09 < 0x0A < 0x0F < 0x10 < 0xA0 < 0xF0 < 0xFF.

Each component should be considered separately.

4.28: GENERIC PARAMETER - IMPURITY

Any of the following names can be used for impurity parameters. Names can be abbreviated as long as the individual words remain unique in the list. A dot(.) can be used as a word separator, otherwise multiple word impurities must be quoted. The short name is the preferred abbreviation. Special effort will be made to keep these names unique when adding new impurities.

Note: The number symbol(#), the equal sign(=), the single quote (') and the space symbol () must be quoted.

4.28.1: Possible Values

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Vacancies"	vac	log	1.0
"Interstitials"		log	1.0
"Arsenic"		log	1.0e+10
"Phosphorus"	phos	log	1.0e+10
"Antimony"		log	1.0e+10
"Boron"		log	1.0e+10
"Donors"		log	1.0e+10
"Acceptors"		log	1.0e+10
"Composition Fraction X"*	comp.fract.x	linear	
"Composition Fraction Y"*	comp.fract.y	linear	
"Electron Conc Process Simulation"*		log	1.0
"Hole Conc Process Simulation"*		log	1.0
"X Velocity Process Simulation"*	x.vel.p.s	linear	
"Y Velocity Process Simulation"*	y.vel.p.s	linear	
"Dry O2"*		log	1.0
"Wet O2"*		log	1.0
"Interstitial Traps"*		log	1.0
"Gold Conc"*	gold	log	1.0
"Cesium"		log	1.0
"Delta Area"*		linear	
"Stress XX"*		linear	

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Stress XY"*		linear	
"Stress YY"*		linear	
"Fixed Oxide Charge"*		linear	
"Potential"	pot	linear	
"Device Potential"*	dev.pot	linear	
"N Mobility"	n.mob	log	1.0
"P Mobility"	p.mob	log	1.0
"Total Field"*		log	1.0
"Fixed Charge"*		log	1.0
"Impact Ionization Rate"*		linear	
"N Carrier Conc"*	n.car	log	1.0
"P Carrier Conc"*	p.car	log	1.0
"Conduction Current"*	cond.cur	linear	
"Displacement Current"*	disp.cur	linear	
"Total Current"*	t.cur	linear	
"Electron QFL"*		linear	
"Hole QFL"*		linear	
"Valency Band Potential"*	val.band.pot	linear	
"Conduction Band Potential"*	cond.band.pot	linear	
"NetDoping" "Net Doping"*	net.dop	log	1.0e+10
"AbsNetDoping" "Abs Net Doping"*	abs.net.dop	log	1.0e+10
"Charge Concentration"*		log	1.0
"Carrier Concentration"*	car	log	1.0
"Recombination Rate"*		log	1.0
"Impact Gen'd Carriers"*	impact.g.car	log	1.0
"X Dir Electric Field"*		linear	
"Y Dir Electric Field"*		linear	
"Z Dir Electric Field"*		linear	

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Hole Temperature"*		linear	
"Elec Temperature"*		linear	
"Semiconductor temp"*		linear	
"Hole temp gradient"*		linear	
"Electron temp grad"*		linear	
"Electron velocity"*	elec.vel	linear	
"Hole velocity"*	hole.vel	linear	
"Intrinsic Conc (nio)"**		log	1.0
"Aluminium conc"*		log	1.0e+10
"Indium conc"*		log	1.0e+10
"Gallium conc"*		log	1.0e+10
"Carbon conc"*		log	1.0
"QFL Gradient X-comp"*		linear	
"QFL Gradient Y-comp"*		linear	
"Total Field 2"*		linear	
"PAC"		linear	
"Intensity"		linear	
"Norm Intensity"*		linear	
"Norm grad Int"*		linear	
"Total Doping"*		log	1.0e+10
"Net Active Doping"*		log	1.0e+10
"Active Boron"*	act.boron	log	1.0e+10
"Active Phosph"*	act.phos	log	1.0e+10
"Active Arsenic"*	act.arsen	log	1.0e+10
"Active Antimony"*	act.antim	log	1.0e+10
"Insulator Charge"*	ins.ch	log	1.0
"Interface Charge"*	int.ch	log	1.0
"Semi Fixed Charge"*		log	1.0
"Slow State Density"*		log	1.0
"Fast State Density"*		log	1.0

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Occupancy Trap 1"*	occ.trap.1	linear	
"Occupancy Trap 2"*	occ.trap.2	linear	
"Occupancy Trap 3"*	occ.trap.3	linear	
"Occupancy Trap 4"*	occ.trap.4	linear	
"Occupancy Trap 5"*	occ.trap.5	linear	
"Occupancy Trap 6"*	occ.trap.6	linear	
"Occupancy Trap 7"*	occ.trap.7	linear	
"Occupancy Trap 8"*	occ.trap.8	linear	
"Occupancy Trap 9"*	occ.trap.9	linear	
"Occupancy Trap 10"*	occ.trap.10	linear	
"Electron Current X-comp."*	elec.cur.x	log	1.0
"Electron Current Y-comp."*	elec.cur.y	log	1.0
"Electron Current Z-comp."*	elec.cur.z	log	1.0
"Hole Current X-comp."*	hole.cur.x	log	1.0
"Hole Current y-comp."*	hole.cur.y	log	1.0
"Hole Current Z-comp."*	hole.cur.z	log	1.0
"Current X-component"*	cur.x	log	1.0
"Current Y-component"*	cur.y	log	1.0
"Current Z-component"*	cur.z	log	1.0
"Displ Current X-component"*	disp.cur.x	log	1.0
"Displ Current Y-component"*	disp.cur.y	log	1.0
"Photo Generation rate"*		log	1.0
"Hole Current magnitude"*	hole.cur.mag	log	1.0
"Electron Current magnitude"*	elec.cur.mag	linear	
"Current Magnitude"*	cur.mag	linear	
"Current Flow"*	cur.flow	linear	
"Total Current (area)"*	t.cur.area	linear	
"Elec Cur Dens X"*	elec.cur.dens.x	linear	
"Hole Cur Dens X"*	hole.cur.dens.x	linear	
"Current Dens X"*	cur.dens.x	linear	

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Elec Cur Dens Y"*	elec.cur.dens.y	linear	
"Hole Cur Dens Y"*	hole.cur.dens.y	linear	
"N int.x"		linear	
"N int.y"		linear	
"P int.x"		linear	
"P int.y"		linear	
"Extended def size"*		linear	
"Extended def dens"*		linear	
"Current Dens Y"*	cur.dens.y	linear	
"Hole Mobility Lateral"*	hole.mob.lat	linear	
"Elec Mobility Lateral"*	elec.mob.lat	linear	
"Hole Mobility Transverse"*	hole.mob.trans	linear	
"Elec Mobility Transverse"*	elec.mob.trans	linear	
"Electron SRH rec. tno"*		linear	
"Hole SRH rec. tno"*		linear	
"Cooling package Temp"*		linear	
"Equilibrium potential"*		linear	
"Applied potential"*		linear	
"Joule Heat Power"*		linear	
"Total Heat Power"*		linear	
"Rec."*		linear	
"Heat conductivity"*		linear	
"Heat capacity"*		linear	
"Displacement J"*		linear	
"Displacement J X-comp"*		linear	
"Displacement J Y-comp"*		linear	
"Hole Diff coeff."*		linear	
"Electron Diff coeff."*		linear	
"Einstein Rel. Corr. Elect"*		linear	
"Einstein Rel. Corr. Holes"*		linear	

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Ionization Effect. field"*		linear	
"Ionization coeff h+"*		linear	
"Ionization coeff e-"*		log	1.0
"Total recombination rate"*		linear	
"SRH recombination rate"*		linear	
"Auger recombination rate"*		linear	
"User recombination rate"*		linear	
"Surface recombination rate"*		linear	
"Effect. min.Carr. lifetime"*	eff.min.car.lif e	linear	
"SRH rec. Effect. lifetime"*		linear	
"Auger Recom. Eff. lifetime"*		linear	
"User Recomb. Eff. lifetime"*		linear	
"Surf. Recom. Eff. lifetime"*		linear	
"Effective BGN"*		linear	
"Effective BGN from C-band"*		linear	
"Effective GN from V-band"*		linear	
"Effective nie"*		linear	
"Structure temp"*		linear	
"Heat capacitance"*		linear	
"Heat conductance"*		linear	
"Material density"*		linear	
"Package layer index"*		linear	
"Package material index"*		linear	
"Material Type #"*		linear	
"Sem./Ins Material #"*		linear	
"Region #"*		linear	
"Mat.type # w/o electrodes"*		linear	
"Sem./Ins # w/o electrodes"*		linear	
"Region # w/o electrodes"*		linear	

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Electrode #"*		linear	
"SRH par. n1/nie"*		linear	
"Relative Permittivity"*		linear	
"Celsius Temp"*		linear	
"D Vector Magnitude"*		linear	
"D Vector X-comp"*		linear	
"D Vector Y-comp"*		linear	
"D Vector Z-comp"*		linear	
"Cond. Current X-comp"*	cond.cur.x	linear	
"Cond. Current Y-comp"*	cond.cur.y	linear	
"Electron Velocity X-comp"*	elec.vel.x	linear	
"Electron Velocity Y-comp"*	elec.vel.y	linear	
"Hole Velocity X-comp"*	hole.vel.x	linear	
"Hole Velocity Y-comp"*	hole.vel.y	linear	
"Elec. Ionix. Eff. Field."*		linear	
"Hole. Ionix. Eff. Field."*		linear	
"Ratio nie/maj.carr.conc."*		linear	
"Ratio nie/ni"*		linear	
"Eff. BGN w/o T-depend."*		linear	
"Thomson Heat Power"*		linear	
"Net Doping (linear)"*		linear	
"Total Doping (linear)"*		linear	
"Charge Concentration (linear)"*		linear	
"Heat Flow Density"*		linear	
"X Dir Heat Flow Density"*		linear	
"Y Dir Heat Flow Density"*		linear	
"Z Dir Heat Flow Density"*		linear	
"dT/dx"		linear	
"dT/dy"		linear	
"dT/dz"		linear	

Impurity Full Name	Preferred Abbreviations	Default imp.refine	
		Scale	Transition Value
"Gradient of Time"*		linear	
"Z Plane Index"*		linear	

Note: The number symbol(#), the equal sign(=), the single quote (') and the space symbol () must be quoted.

4.29: GENERIC PARAMETER - MATERIAL

Any of the following names can be used for material parameters. Names can be abbreviated as long as the individual words remain unique in the list. A dot (.) can be used as a word separator, otherwise multiple word materials must be quoted. The short name is the preferred abbreviation. A special effort will be made to keep these names unique when adding new materials.

Note: The number symbol(#), the equal sign(=), the single quote (') and the space symbol () must be quoted.

4.29.1: Possible Values

Material Number (Structure File ID)	Material Full Name (TonyPlot Material Name)	Aliases (Compatibility Names)
0	"Gas"	
1	"SiO2"	"Silicon Oxide" "SiO_2" "SiOxide" "Oxide"
2	"Si3N4"	"Silicon Nitride" "SiNO3" "SiNO_3" "SiNitride" "Nitride"
3	"Silicon"	"Si"
4	"Polysilicon"	"Poly"
5	"OxyNitride"	"OxyNit"
6	"Aluminum"	"Al"
7	"Photoresist"	"PhotoRes"
8	"GaAs"	
9	"Sapphire"	
10	"Gold"	"Au"
11	"Silver"	"Ag"
12	"AlSi"	
13	"Tungsten"	"W"
14	"Titanium"	"Ti"
15	"Platinum"	"Pt"
16	"Palladium"	"Pd"
17	"Cobalt"	"Co"
18	"Molibdinum"	"Mo" "Md"
19	"Lead"	"Pb"
20	"Iron"	"Fe"

Material Number (Structure File ID)	Material Full Name (TonyPlot Material Name)	Aliases (Computability Names)
21	"Tantalum"	"Ta"
22	"AlSiTi"	
23	"AlSiCu"	
24	"AlGaAs"	
25	"InGaAs"	
26	"AlInAs"	
27	"InP"	
28	"Vacuum"	
29	"Fictive GaAs"	"GaAs-Hetro" "FGA"
30	"Mask Opaque"	
31	"Mask Clear"	
32	"Germanium"	"Ge"
33	"AlAs"	
34	"TEOS"	
35	"BSG"	
36	"BPSG"	
40	"Alpha Si 1"	"alpha-Si#1" "~a-Si__1"
41	"Alpha Si 2"	"alpha-Si#2" "~a-Si__2"
42	"Alpha Si 3"	"alpha-Si#3" "~a-Si__3"
43	"Alpha Si 4"	"alpha-Si#4" "~a-Si__4"
50 ... 59	"User #1" ... "User #10"	"UD1(<user-defined-name>)" ... "UD10(<user-defined-name>)"
60	"AlxGa1_xAs_x_0.25"	"AlxGa1-xAs, x=0.25" "ALG1" "AlGaAs1 (ALG1)"
61	"AlxGa1_xAs_x_0.5"	"AlxGa1-xAs, x=0.5" "ALG2" "AlGaAs2 (ALG2)"
62	"AlxGa1_xAs_x_0.75"	"AlxGa1-xAs, x=0.75" "ALG3" "AlGaAs3 (ALG3)"
63	"InxGa1_xAs_x_0.50 Unstr"	"InxGa1-xAs, x=0.50, unstrained" "ING0" "InGaAs (ING0)"
64	"InxGa1_xAs_x_0.33 Str GaAs"	"InxGa1-xAs, x=0.33, strained matched to GaAs" "ING1" "InGaAs (ING1)"

Material Number (Structure File ID)	Material Full Name (TonyPlot Material Name)	Aliases (Compatibility Names)
65	"InxGa1-xAs_x_0.75 Str InP"	"InxGa1-xAs, x=0.75, strained matched to InP" "ING2" "InGaAs (ING2)"
66	"AlxIn1-xAs_x_0.50"	"AlxIn1-xAs, x=0.50" "ALIN" "AlInAs (ALIN)"
69	"Barrier"	
70	"TiW"	
71	"PMMA"	
72	"SOG"	
73	"Polyimide"	
74	"Cooling package material"	
75	"Copper"	"Cu"
76	"Tin"	"Sn"
77	"Nickel"	"Ni"
78	"Ambient"	
79	"Air"	
80	"WSix"	"Tungsten Silicide" "WSi2"
81	"TiSix"	"Titanium Silicide" "TiSi2"
82	"NiSix"	"Nickel Silicide" "NiSi2"
83	"CoSix"	"Cobalt Silicide" "CoSi2"
84	"TaSix"	"Tantulum Silicide" "TaSi2"
85	"PdSix"	"Paladium Silicide" "PdSi2"
86	"PtSix"	"Platinum Silicide" "PtSi2" "PtSi2"
87	"MoSix"	"Molybdenum Silicide" "Mdsi2" "MoSi2" "Moly Silicide"
88	"ZrSix"	"Zirconium Silicide" "ZrSi2"
89	"AlSix"	"Aluminum Silicide" "AlSi2"
90	"Insulator"	
91	"Conductor"	
92	"Contact"	
99	"3C-SiC"	

Material Number (Structure File ID)	Material Full Name (TonyPlot Material Name)	Aliases (Compatibility Names)
100	"Diamond"	
101	"SiGe"	
102	"6H-SiC"	"a-SiC"
103	"4H-SiC"	"b-SiC"
104	"AlP"	
105	"AlSb"	
106	"GaSb"	
107	"GaP"	
108	"InSb"	
109	"InAs"	
110	"ZnS"	
111	"ZnSe"	
112	"ZnTe"	
113	"CdS"	
114	"CdSe"	
115	"CdTe"	
116	"HgS"	
117	"HgSe"	
118	"HgTe"	
119	"PbS"	
120	"PbSe"	
121	"PbTe"	
122	"SnTe"	
123	"ScN"	
124	"GaN"	
125	"AlN"	
126	"InN"	
127	"BeTe"	
128	"InGaP"	
129	"GaSbP"	
130	"GaSbAs"	

Material Number (Structure File ID)	Material Full Name (TonyPlot Material Name)	Aliases (Compatibility Names)
131	"InAlAs"	
132	"InAsP"	
133	"GaAsP"	
134	"HgCdTe"	
135	"InGaAsP"	
136	"AlGaAsP"	
137	"AlGaAsSb"	
140	"SiN"	
141	"Computational Window"	
142	"Phase shift"	
143	"Si"	"Green's Silicon (for atlas)"
144	"Polymer"	
145	"CuInGaSe"	
146	"InGaN"	
147	"AlGaN"	
148	"InAlGaN"	
149	"InGaNAs"	
150	"User #11"	"UD11 (<user-defined-name>)"
...
189	"User #50"	"UD50 (<user-defined-name>)"
190	"InGaNp"	
191	"AlGaNAs"	
192	"AlGaNp"	
193	"AlInNAs"	
194	"AlInNp"	
195	"InAlGaAs"	
196	"InAlGaP"	
197	"InAlAsP"	
198	"ITO"	
199	"Pentacene"	
200	"Alq3"	

Material Number (Structure File ID)	Material Full Name (TonyPlot Material Name)	Aliases (Compatability Names)
201	"TPD"	
202	"PPV"	
203	"Organic"	
204	"Ba2YCu3O7"	
205	"Ba2NdCu3O7"	

Note: The number symbol(#), the equal sign(=), the single quote (') and the space symbol () must be quoted.

4.30: GENERIC PARAMETER - PATTERN

On black and white displays, pattern fills are used instead of colors. There are 18 patterns numbered 0 to 17. Pattern 0 (solid) and pattern 17 (empty) are normally not used. Each material has a default pattern, which will be used by the region if no pattern is set.

A

Advanced Features (Meshing Example 2)	
3D Structures	2-25-26
Circular Devices	2-29
Combining Two ATHENA Structures into a Single Device	2-27
Stretch and Cut	2-28
Analytic Functions	
Location Dependent Variables	3-19
User Supplied Variables	3-19-21
<i>See also</i> Roll-Off Functions	
Constant	3-19
Error Function	3-19
Error Function (Dist)	3-20
Exponential	3-19
Exponential (Dist)	3-21
Gaussian	3-19
Gaussian (Dist)	3-20
Linear (Dist)	3-20-21
Location Dependent Variables	3-19, 3-20
Logarithmic	3-19
Logarithmic (Dist)	3-21
Step Function	3-20
User Supplied Variables	3-19, 3-20
<i>See also</i> Roll-Off Functions	
ATLAS	1-6

B

Base Window	
Control Panel	1-4
Control Windows	1-4
Layout and Functionality	1-3
Main Panel Controls	1-4
Boundary Conditioning	3-8, 3-9
<i>See also</i> MeshBuild	

C

Cards And Parameters	4-1-2
<i>See also</i> Statements	
Command File	
Default Files	1-6
<i>See also</i> File Control	

D

DevEdit3D	2-25
Doping	
3D Doping	3-7
Deleting Source Objects	3-7
Impurity Source Box	3-7
Impurity Source Line	3-6
Source Attributes	3-7

Doping Profiles	
Adding	3-21-22
Drawing Regions	
Base Impurity (Doping)	3-4

E

Editing Regions	
Adding	3-2
Selecting Materials	3-2
Error Function	3-19
Error Function (Dist)	3-20

F

File Control	
Command Files	1-6
SILVACO Standard Structure Files	1-5
SILVACO Standard vs. Devedit	1-6

I

Impurities	
Add Impurity Mode	3-14
Defining	3-13
Loading	3-13
Roll-off Direction	3-15-18
Source Area	3-14-15
Viewing	3-13
Impurities (Meshing Example 1)	
Adding	2-8
Displaying the Doping	2-9
Modifying	2-9

J

Join Function	
User Supplied Variables	3-22, 3-23

M

Mesh Creation	
Base Mesh Parameters	3-12
Final Meshing	3-12
Impurity Refinement	3-12
Mesh Constraints	3-12
MESH CONTROLS	3-12
Saving	3-12
Mesh Creation (Meshing Example 1)	
Manual Refine Box	2-12
Mesh Constraints	2-11
Mesh Parameters	2-9
MeshBuild	2-9
Refine on Quantities	2-9-10

Mesh Creation (Meshing Example 2)	
Boundary Conditioning	2-16-17
Mesh Constraints	2-19-21
Mesh Parameters	2-18
Refine on Quantities	2-18-19
MeshBuild	
3D Structures	3-11
Adaptive Meshing	3-10
Boundary Conditioning	3-8
Limitations	3-9
Mesh Constraints	3-9-10
Refinement	3-10
Refining	3-11
Relaxing	3-11
Tensor	3-11
Work Area Resizing	3-11
Meshing	2-1
Creating	2-2-13, 3-12
<i>See also</i> Mesh Creation	
MeshBuild	3-8-11
Meshing Example 1	2-2-13
Meshing Example 2	2-14-24
Remeshing	2-14
Meshing Display (Meshing Example 2)	
Doping	2-15
Impurity Junctions	2-15
Zoom	2-15
Meshing Example 1	
Defining Regions	2-3-7
Impurities	2-8-9
Mesh Creation	2-9-12
Saving Mesh Files	2-12-13
Work Area	2-2-3
Meshing Example 2	
Advanced Features	2-25-30
Display	2-15
Loading the Structure	2-14
Mesh Creation	2-16-21
Obtain existing structure	2-14
Saving Mesh Files	2-21-24
Structure editing	2-14
Summary	2-30
P	
Parameter Type	
BOOLEAN	4-38
COLOR	4-39
IMPURITY	4-40-47
MATERIAL	4-48-53
PATTERN	4-54
Problems	1-1
R	
Regions	
Deleting	3-4
Deleting Boundary Points	3-5
Drawing	3-3-4
Editing. <i>See</i> Editing Regions	
Modifying	3-5
Regions (Meshing Example 1)	
Adding	2-3-4
Electrodes	2-7
Etch then Adding	2-6
Material Selection and Uniform Doping	2-4-5
Modifying	2-6
Setting Mole (Composition) Fraction	2-5
Roll-Off Directions	
Rolloff=Both	3-16
Rolloff=High	3-16
Rolloff=High.P.Step	3-16
Rolloff=Low	3-17
Rolloff=Low.P.Step	3-17
Rolloff=P.Step	3-18
Rolloff=Step	3-17
Rolloff=Step.P.High	3-18
Rolloff=Step.P.Low	3-17
Roll-Off Functions	
Analytic Functions	3-19-21
Analytic Functions (Dist)	3-20-21
Combining Impurity Rolloffs	3-24-28
Deleting Impurities	3-24
Doping Profiles	3-21-22
Editing Impurities	3-24
S	
Save Generic Mesh Commands	1-5
Saving Mesh Files (Meshing Example 1)	
Batch Mode	2-12-13
Command File	2-12
Structure File	2-12
Saving Mesh Files (Meshing Example 2)	
Batch Mode	2-22-24
Command File	2-21
Structure File	2-21
SILVACO Standard Structure File (.str)	
Loading	1-5
Writing	1-5
Solution	1-1
Source Attributes	
Gaussian	3-7
SSUPREM3	3-21
SSUPREM4	3-21
Startup	1-2
Statements	
BASE.MESH	4-3
BOUNDARY.CONDITIONING	4-4-5
CONSTRAINT.MESH	4-6-9
CUT	4-10
DEPOSIT	4-11
FLIP	4-12
IMPURITY	4-13-16
IMPURITY REFINEMENT	4-17-18
INITIALIZE	4-19
JOIN	4-20
MESH	4-21

MIRROR	4-22
MOVE	4-23
PROFILE	4-24
QUIT	4-25
REFINE	4-26
REGION	4-27-28
RENUMBER.REGIONS	4-29
SOURCE	4-30
STRETCH	4-31-33
STRUCTURE	4-34
SUBSTRATE	4-35
WORK.AREA	4-36
Z.PLANES	4-37
Structure Editing	
Editing Summary	3-1
Panning	3-1
Resolution	3-1
Zooming	3-1
W	
When Not to Use	1-1
When to use	1-1

This page is intentionally left blank.