cadence®

Pegasus™ Interactive User Guide

Product Version 23.11
September 2023

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Contents

Preface

This document describes how to verify a layout against physical design rules using the Pegasus™ Interactive. Using Pegasus Interactive, you can capture and validate design intent and ensure design convergence by performing in-design interactive DRC checks on the layout immediately after editing the layout. These checks are performed using the Pegasus rule deck, which is a signoff rule deck.

The intent of this manual is to provide:

- In depth knowledge of setting Pegasus Interactive
- Details of its verification capability
- Information on customizing its rule sets

How This Manual is Organized

The chapters are organized as following:

- Getting Started: Specifies prerequisites for using Pegasus Interactive including design rules preparations and error browser setup.
- Pegasus Interactive Toolbar: Describes how to customize toolbar and run option graphical interface that is best suit for your environment.
- All About Design Rules: Snapshot: Provides concepts of snapshots and how to create and manage them that reflects your intents.
- Invoke Design Rule Checking: Describes how to use Pegasus Interactive to perform DRC in Verify-Design mode.
- Error Viewing: Describes how to use different browser utility provided by Virtuoso Layout Suite and Pegasus.
- Batch Commands: Describes batch commands used for snapshot creation
- Environment Variables: Describes Pegasus Interactive environment variables.

Software Requirements

In order to run Pegasus Interactive, both Pegasus and Virtuoso Layout Suite software need to be available. For the latest Software requirements, see the *Compatibility Statement* section in the *Pegasus What's New*.

License Requirements

This section describes license requirements.

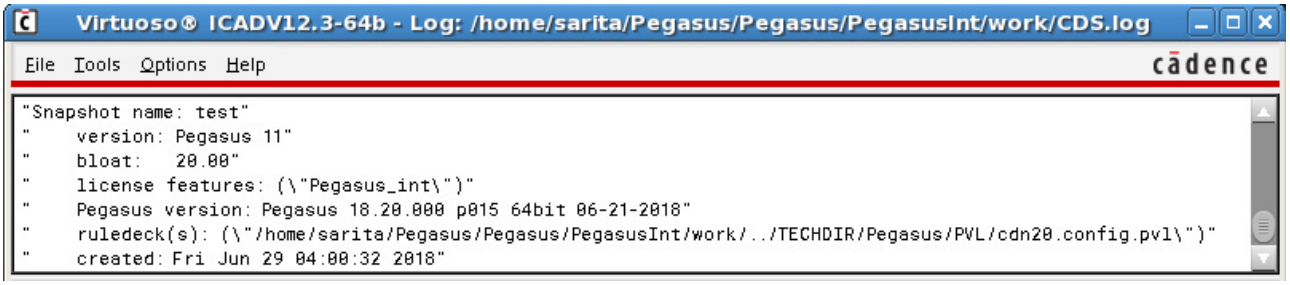
License:

Name	Number	Description
Pegasus_int	PGSS210	Cadence® Pegasus™ Interactive.

To run Pegasus Interactive, you must first have a snapshot. A snapshot is a collection of reference files that Pegasus Interactive refers during DRC. For details of these files, refer to All About Design Rules: Snapshot. For snapshot creation, Pegasus checks for the existence of license Pegasus_int during snapshot file creation, and allows the operation if that licenses exists.

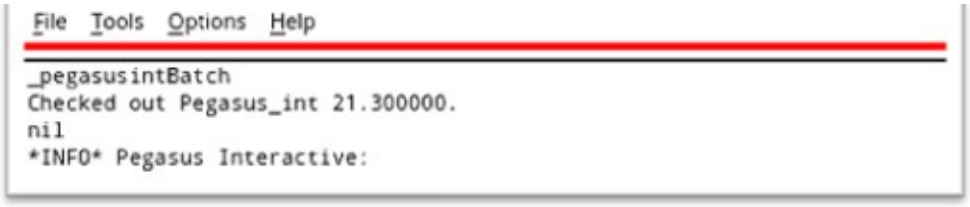
During the snapshot creation, license requirement(s) is documented as a part of the snapshot header information. At this stage, no license is checked out and Pegasus Interactive only determines license requirement based on the content of rule deck.

Following figure illustrates `test` snapshot header information. It indicates that Pegasus_Int license is required for Pegasus Interactive to run successfully.



License Handling while Running Pegasus Interactive

Pegasus Interactive licenses checkout is one time process when running Pegasus Interactive for the first time. License checkout is printed in CIW:



Unlike its sign-off DRC counterpart, license(s) continues to be checked-out until any of following termination condition is met:

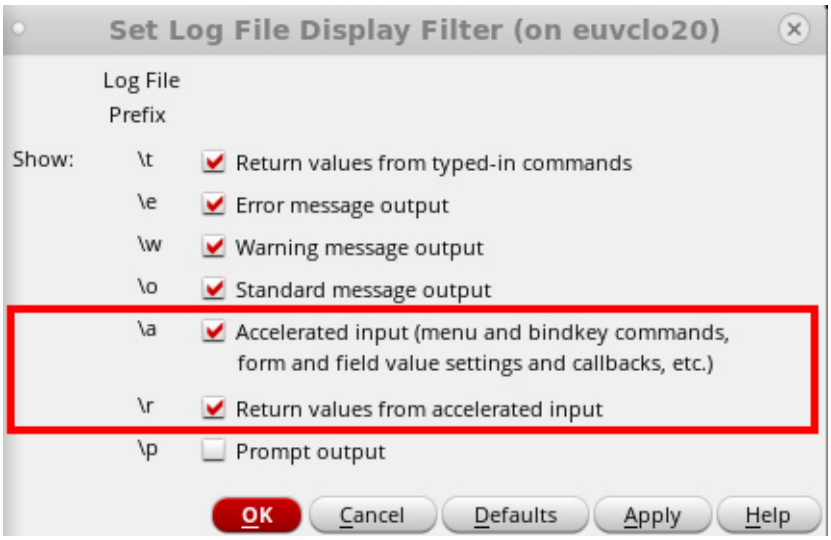
- Close all of layout within a Virtuoso Session
- Terminate Virtuoso session
- Interactive release license through Pegasus Interactive Run Options GUI
- One hour of inactivity

The license release status is printed in CIW. By default, the returned message is *Checking in Pegasus_int*.

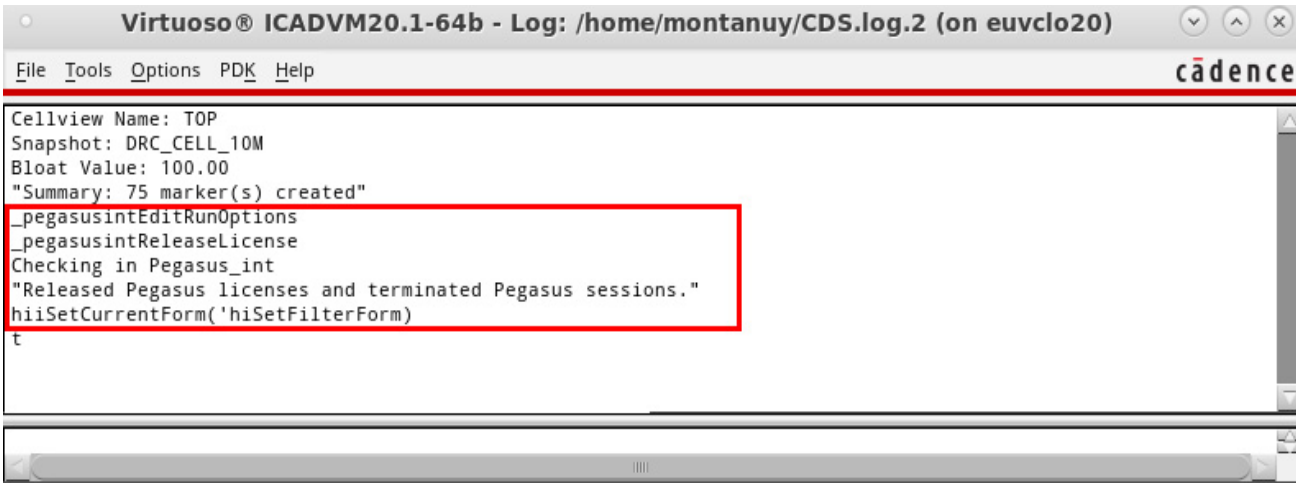
For getting, the complete message:

Released Pegasus licenses and terminated Pegasus sessions

select */a* and */r* check boxes from the *CIW*: *Options -> Log Filter ... -> Set Log file Display filter options* form as shown below:



This will give you the complete message:



Related Documents for Pegasus Interactive

The following documents provide more information about the topics discussed in this guide.

Installation, Environment, and Infrastructure

- For information on installing Cadence products, see the *Cadence Installation Guide*.

- For information on the Virtuoso design environment, see the *Virtuoso Design Environment User Guide*.
- For information on database SKILL functions, including data access functions, see the *Virtuoso Design Environment SKILL Reference*.
- For information on library structure, the library definitions file, and name mapping for data shared by multiple Cadence tools, see the *Cadence Application Infrastructure User Guide*.

Technology Information

- For information on how to create and maintain a technology file and display resource file, see the *Virtuoso Technology Data User Guide* and the *Virtuoso Technology Data ASCII Files Reference*.
- For information on how to access the technology file using SKILL functions, see the *Virtuoso Technology Data SKILL Reference*.

Virtuoso Tools

- For information on design rule driven editing, see the *Virtuoso Design Rule Driven Editing User Guide*.
- For information on how to perform design tasks with the Virtuoso Layout Suite L layout editor, see the *Virtuoso Layout Suite L User Guide*
- For information on how to use the Virtuoso Layout Suite wire editing capability, see *Interactive Wire Editing* in the *Virtuoso Space-based Router User Guide*.
- For information on custom layout SKILL functions, see the *Virtuoso Layout Suite SKILL Reference*.

Additional Learning Resources

Cadence provides Rapid Adoption Kit (RAK) that allows you to quickly adopt Pegasus Interactive in the user design flow. RAK contains design database and exercises that mimics various stages of the design flow. Ask your local Cadence support for how to access Pegasus Interactive RAK.

Typographic and Syntax Conventions

The following typographic and syntax conventions are used in this manual.

`text`	Indicates text you must type exactly as it is presented.
`z_argument`	Indicates text that you must replace with an appropriate argument. The prefix (in this case, *`z_`*) indicates the data type the argument can accept. Do not type the data type or underscore.
`[ ]`	Denotes an optional argument. When used with vertical bars, they enclose a list of choices from which you can choose one.
`{ }`	Used with vertical bars, they denote a list of choices from which you must choose one.
|	Separates a choice of options.
...	Indicates that you can repeat the previous argument.
/	Separates the possible values that can be returned by a Cadence SKILL language function.
text	Indicates names of manuals, menu commands, form buttons, and form fields.

1

Getting Started

This chapter provides basic information on setting up tool and how to run Pegasus Interactive for the first time.

The content of this chapter is organized as follows:

Prerequisites

In order for Pegasus Interactive to run and report errors correctly, following information needs to be pre-established:

- Snapshot: pre-compiled DRC deck that is loadable by Pegasus
- Color scheme setup for Pegasus Results Viewer

When both Pegasus and Virtuoso are available, Pegasus Interactive toolbar is ready to be used on the layout window once it is opened. However, toolbar may not be fully enabled if no snapshot is detected by Pegasus Interactive during layout opening.

You can create your own set of local snapshots or refer to global snapshots provided by CAD.

For details on creating local snapshot, refer to following:

- Quick Start: Create Snapshot
- Manage Snapshots

For details on setting up global snapshot, refer to following:

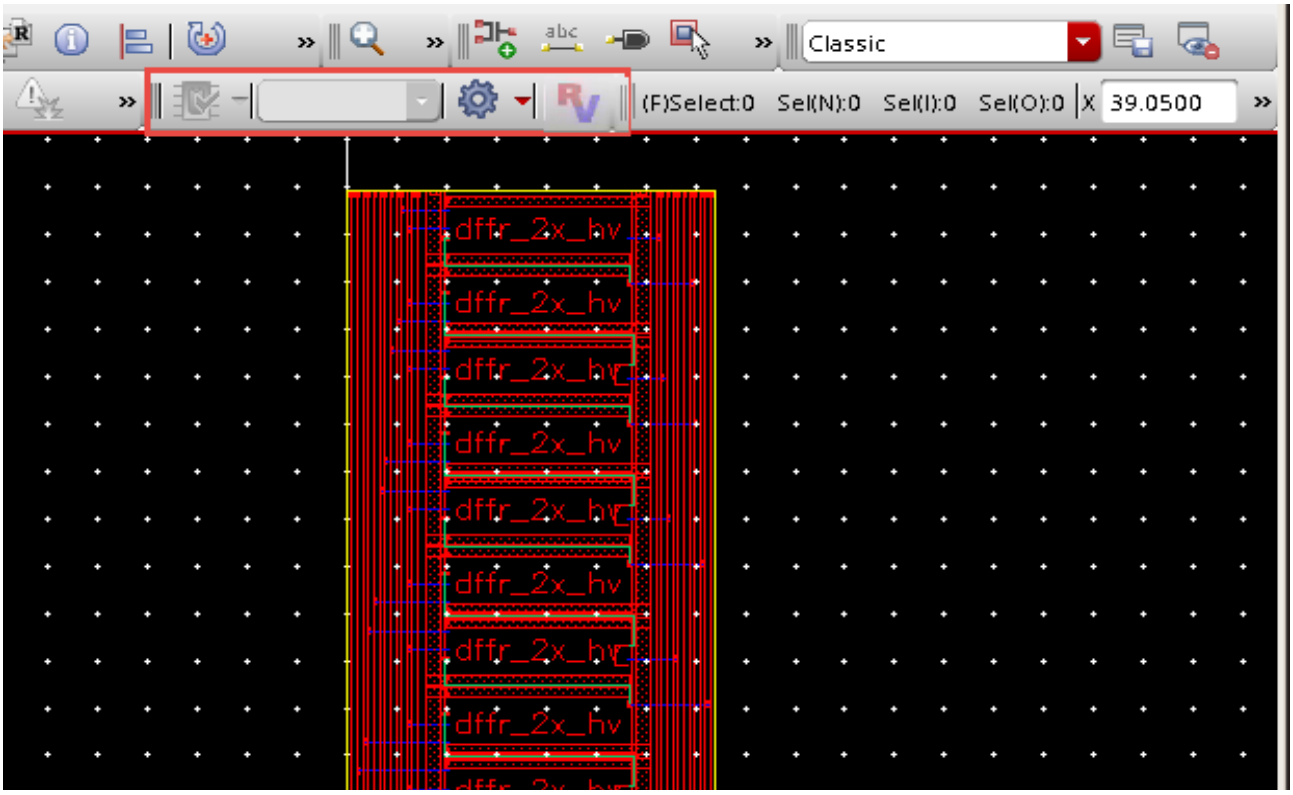
- Snapshot Directory Types and Customizations
- Batch Commands

Pegasus Results Viewer is the default browser for both Pegasus Interactive and Pegasus. It has its own set of Pegasus specific display packages. Thus, it is required for you to integrate this set of display package within Virtuoso for proper error highlight capability.

Using Pegasus Interactive for the First Time

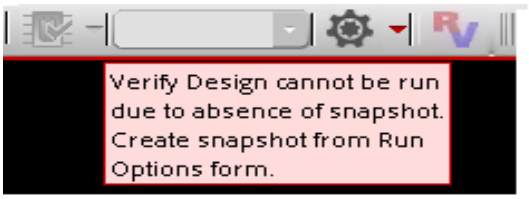
There is no special tool setup for Pegasus Interactive as long as both of Virtuoso and Pegasus installations are present in your Unix PATH.

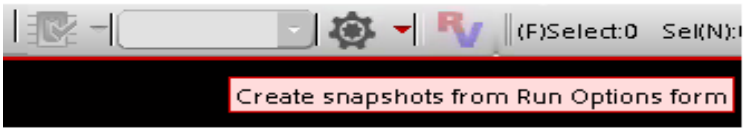
Once you launch Virtuoso and open a layout in any Virtuoso tier (L/ XL/ GXL), Pegasus Interactive toolbar is decked on the layout window as following:



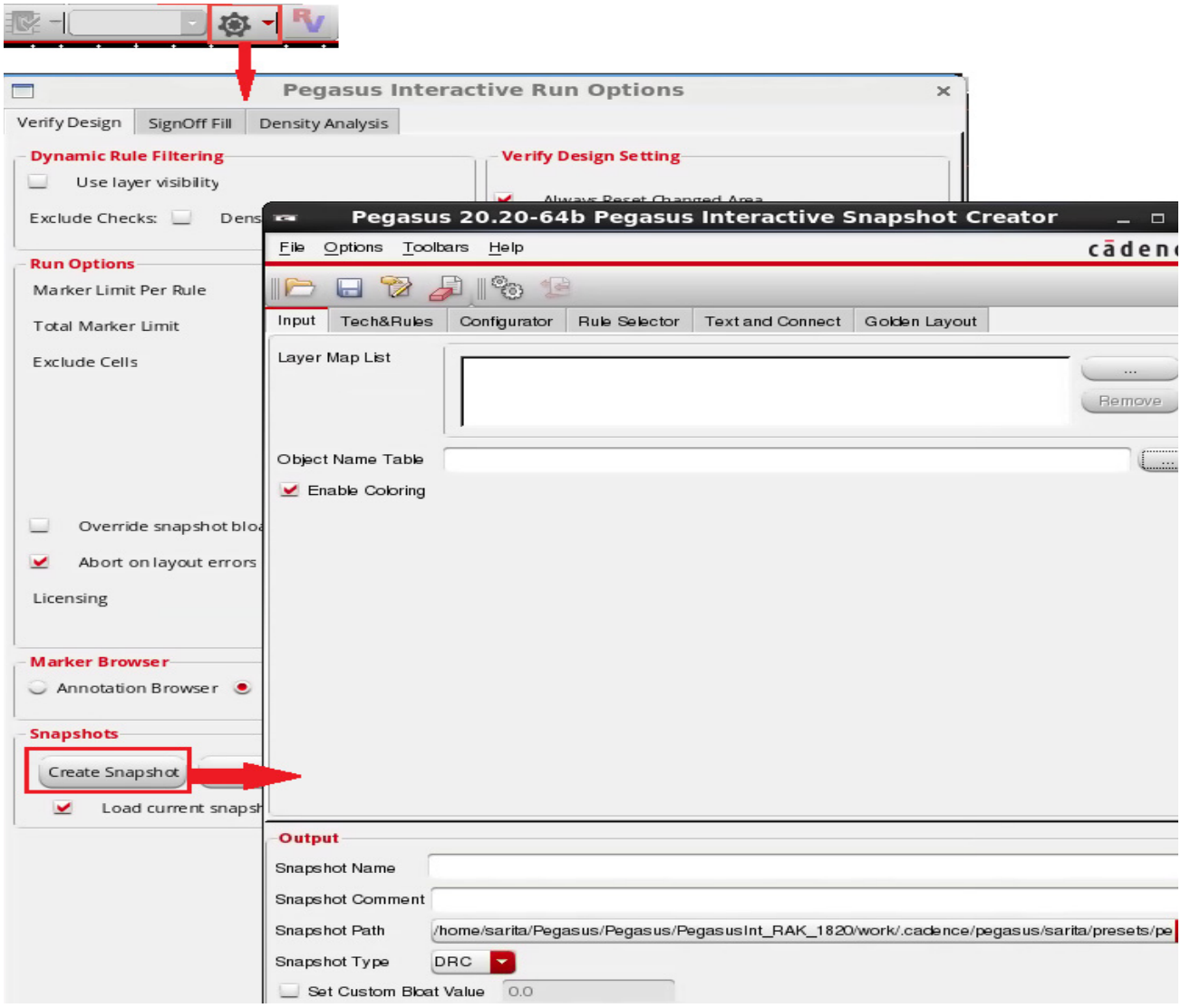
At this point, Pegasus Interactive is not ready to use since all of icons except *Pegasus Interactive Run Option* icon are disabled.

You can query the reason why icons are disabled by hovering your cursor on top of any disabled icons. Tooltips are displayed and informs you what are additional steps required to enable toolbar icons.

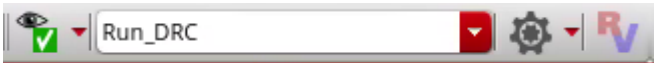




In order to activate the *Verify-Design* icon, you need to first create a snapshot by invoking *Snapshot Creator* within *Pegasus Interactive Run Options* form.



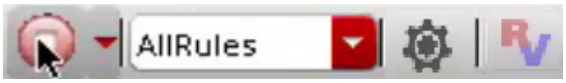
To create a snapshot from the first time, refer to *First Time under Quick Start: Create Snapshot*. Once the snapshot is created, all of icons are activated except Pegasus Results Viewer icon. Pegasus Results Viewer icon is inactivated due to no run results available for Pegasus Results Viewer browser to load.



To run DRC, you can click *Pegasus Interactive Verify-Design* icon using its default setting or other setting described in Verify Design Mode in the *Invoke Design Rule Checking* chapter.

Once Pegasus Interactive is running DRC against the layout, the *Pegasus Interactive Verify-Design* icon changes to a *Stop* symbol. At this moment, you are free to do any of following:

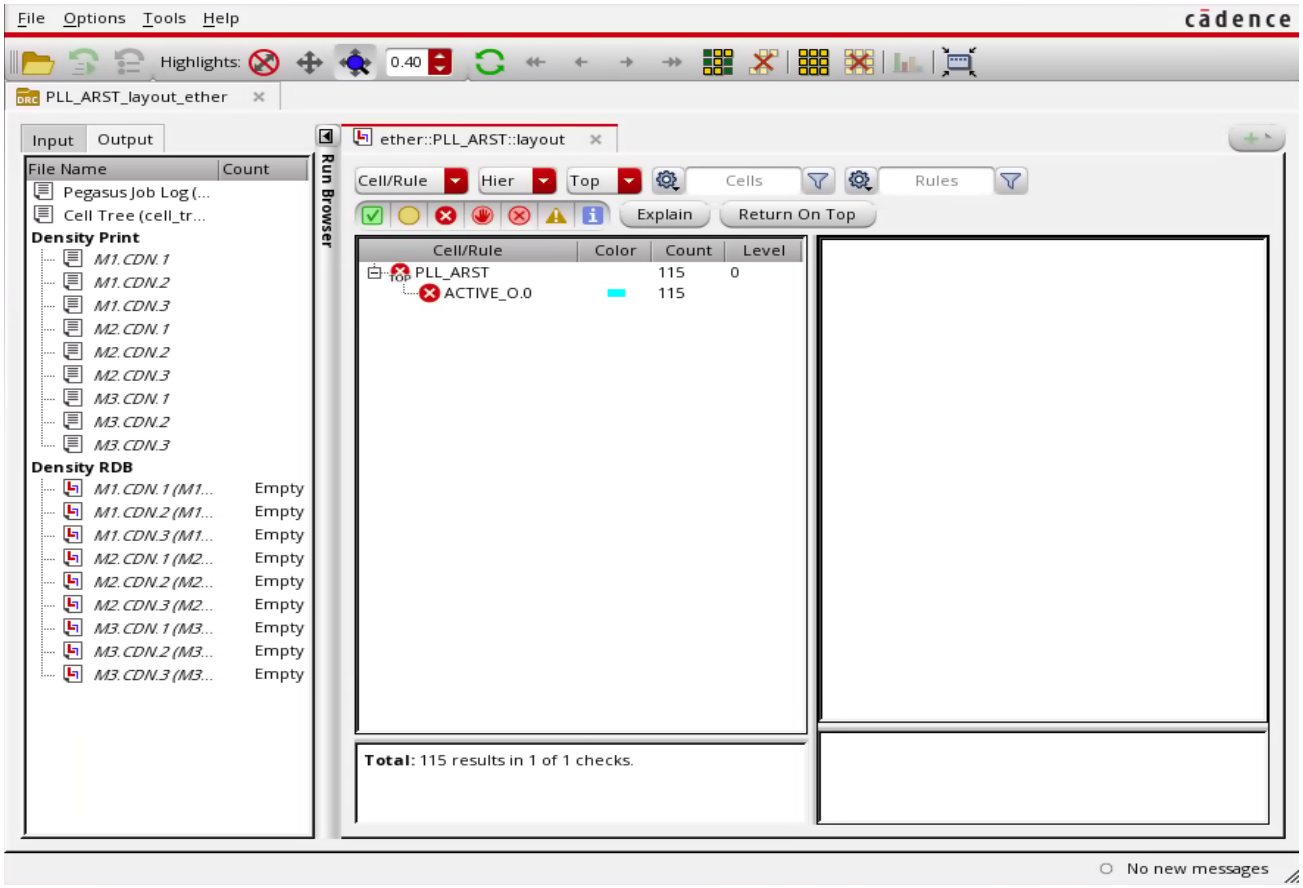
- Terminate the DRC run any time
- Continue editing the same or different layout for the same Virtuoso Session



Once the run is complete, Pegasus Results Viewer is automatically triggered and has error violations populated in the browser. Meanwhile, these errors are highlighted in the layout as well.

By default, Pegasus Interactive generates and displays markers by cell master. Pegasus Results Viewer populates markers by cell name. Markers could be displayed on all cells found in the layout when `drdPegasusIntErrorDisplayMaster` is set to `nil`.

Note: The `drdPegasusIntErrorDisplayMaster` environment variable is available only in ICADV123/IC617 ISR21 or newer. If you are using ICADV123/IC617 ISR20 or older then `drdPVSErrorDisplayMaster` environment variable should be used.



At this point, the Pegasus Results Viewer icon becomes active. You can invoke Pegasus Results Viewer to view the last run results as long as layout is opened regardless its editing status.

For a given snapshot, you may want to run on certain rules based on the layer visibility displayed in the layer Palette. To do so, you can enable Dynamic Rule Filtering (DRF). Once this setting is enabled, Pegasus Interactive only selects check based on current visible layers in the Palette to run DRC.

2

Pegasus Interactive Toolbar

Pegasus Interactive gives you the ability to customize default graphical interface, such as toolbar itself as well as its default run behavior. This chapter goes over each topic in great details.

The content of this chapter is organized as following:

Overview

Pegasus Interactive is an interactive DRC utility available in Virtuoso. It can:

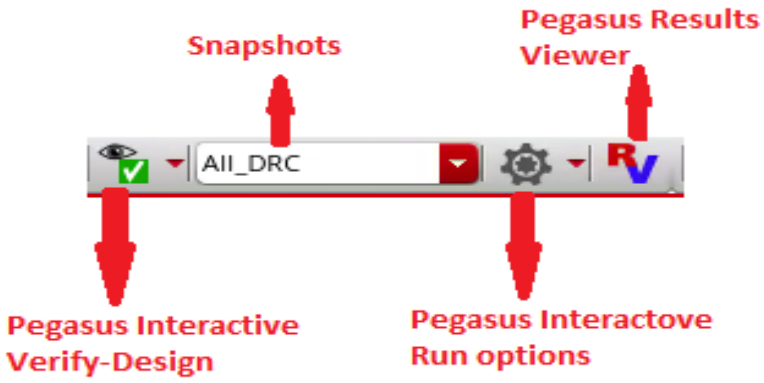
- Directly access OA database and run DRC without data translation
- Refer to Pegasus compiled design rules database, snapshot, without compiling rule deck prior to DRC

Thus, for a quick DRC during layout editing, Pegasus Interactive is more effective than the traditional batch DRC for in design verification solution.

Pegasus Interactive allows you to:

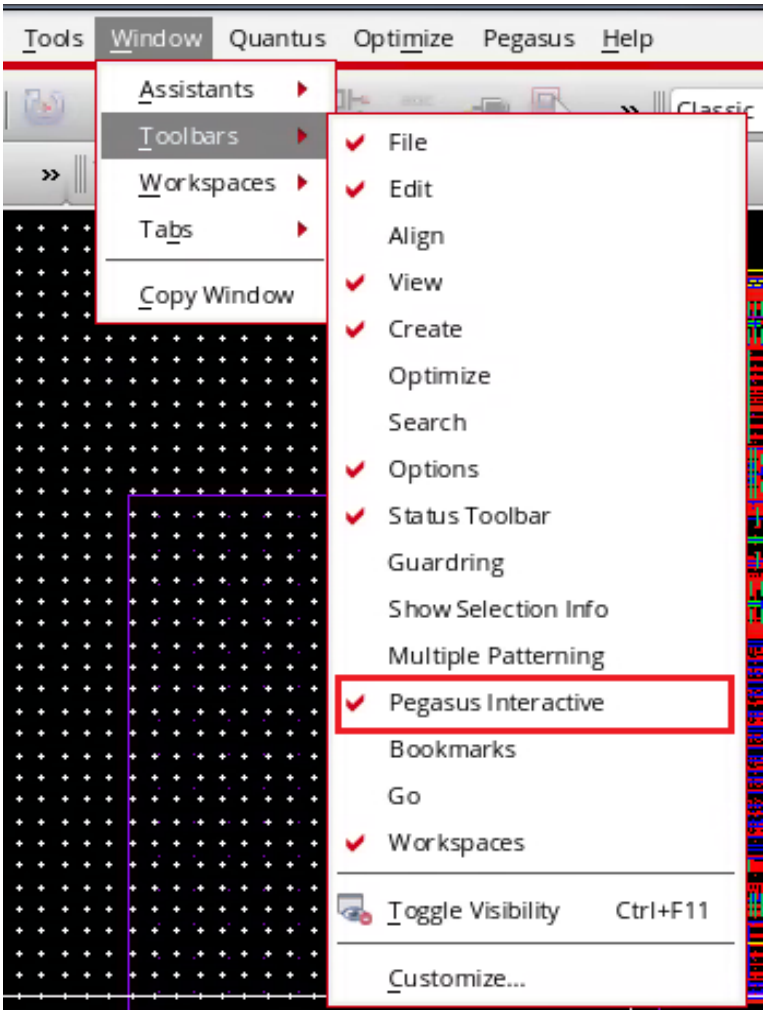
- Edit layout within Virtuoso when Pegasus Interactive is conducting DRC
- Run DRC on either *editable* or the *read-only* cellview

Following figure shows the Pegasus Interactive toolbar:



- *Pegasus Interactive Verify-Design*: DRC operate mode. For detailed description on their usages, see Invoke Design Rule Checking.
- *Snapshots*: For more details on the usage of this toolbar refer to Manage Snapshots.
- *Pegasus Interactive Run Options*: For more details on the usage of this toolbar refer to Run Settings for Verify Design
- *Pegasus Results Viewer*: For more details on the usage of this toolbar refer to Error Viewing.

You can set Pegasus Interactive toolbar availability through Window > Toolbars > Pegasus Interactive. If Pegasus Interactive is not part of Toolbars selection, this is the indication that you did not point to the proper Pegasus installation.



Toolbar Icons

You can customize the toolbar. Following figure shows different ways of customizing the default Pegasus Interactive toolbar.

Related Topics

- Verify-Design on page 24
- Default Active Snapshot on page 25
- Run Option on page 25
- Error Browsing Utility on page 26

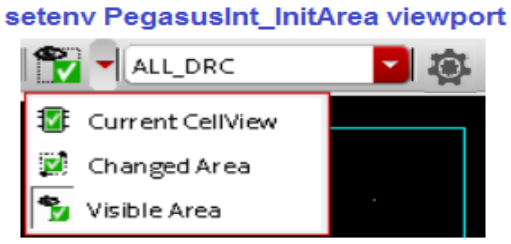
Verify-Design

Verify-Design is an area specific check. Therefore, you can set the default area check to any of following through `PegasusInt_InitArea`:

- Current CellView (Default)
- Changed Area
- Visible Area

For details of `PegasusInt_InitArea` setup, refer to the Virtuoso CDS Environmental Variables section.

Following example sets area of checking based on current visible area.



Default Active Snapshot

By default, items in the snapshot combo field are sorted alphabetically. The first item on the sorted list will be set as an active snapshot and displayed in the combo field. If you prefer to customize a default active snapshot other than tool's selection, you can define it through `PegasusInt_InitSnapshot`.

Once it is set, Pegasus Interactive first searches for the existence of this snapshot and displays it as an active snapshot after Pegasus Interactive toolbar initialization. If your preferred snapshot is not found, Pegasus Interactive will resolve to its default behavior.

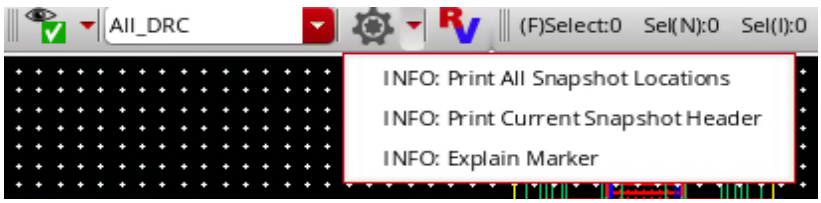
Following example shows Pegasus Interactive's default behavior on determining the active snapshot by setting the first item from the alphabetical sorted list. To override this behavior, `PegasusInt_InitSnapshot` is set to `FinBound` prior to launching Virtuoso. Once the layout is opened, `FinBound` is displayed in the snapshot combo field.

```
Default Active Snapshot: AllRules
```


```
setenv PegasusInt_InitSnapshot FinBound
```

Run Option

You can query the snapshot header information through a snapshot info inquiry utility. It is part of the *Run Option* icon drop-down list.



For details of snapshot info inquiry, refer to Viewing Snapshots Information.

Error Browsing Utility

Pegasus Interactive allows you to view violations through following options:

- Annotation Browser
- Pegasus Results Viewer (Pegasus RV)

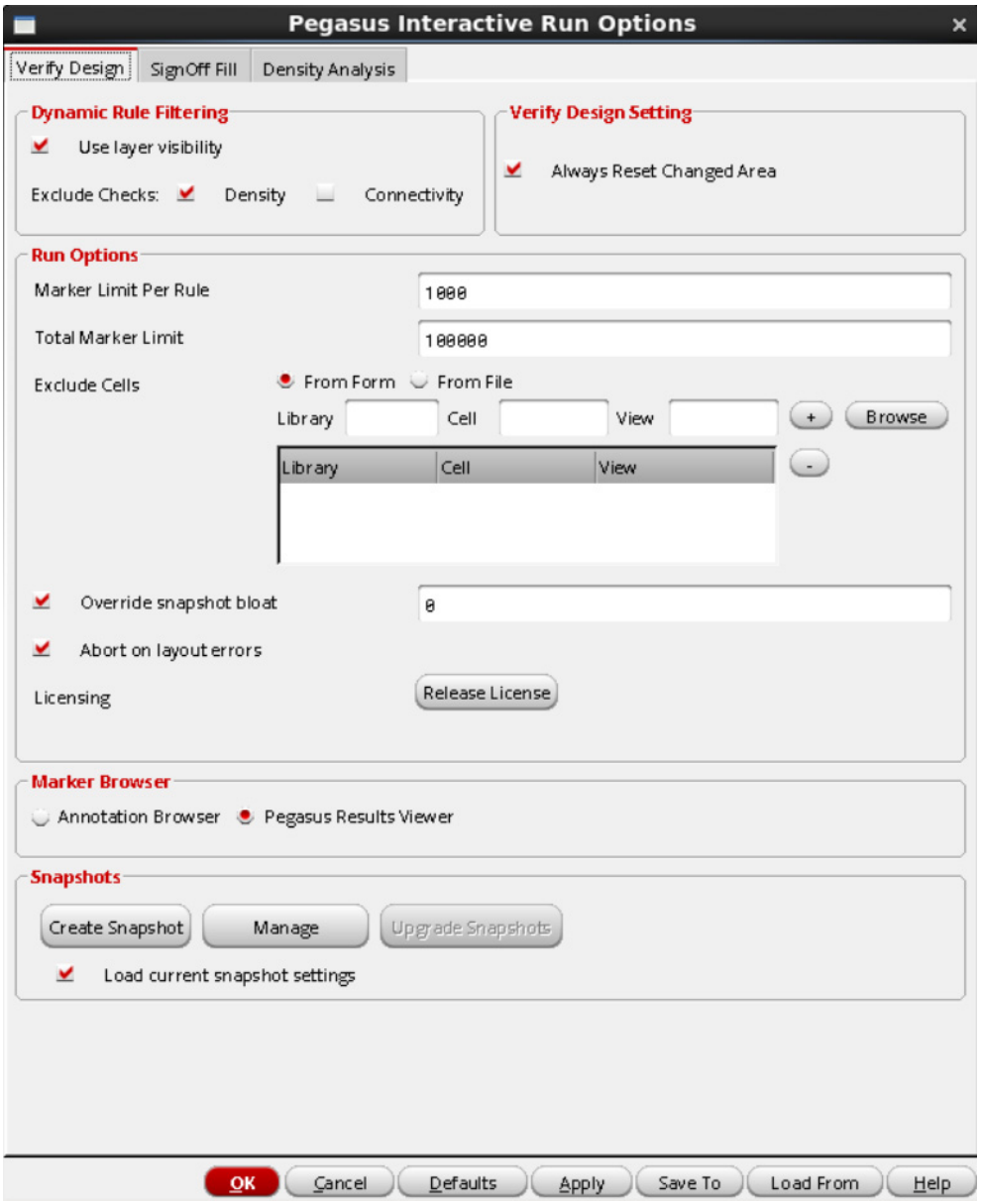
Pegasus RV is the default error browser. To customize the default browser, you can specify the setting through `PegasusInt_MarkerBrowser`.

setenv PegasusInt_MarkerBrowser PegasusRV

setenv PegasusInt_MarkerBrowser AnnotationBrowser

Pegasus Interactive Run Options

The *Pegasus Interactive Run Options* form contains three tabs: Verify Design, SignOff Fill and Density Analysis.



Related Topics

- Run Settings for Verify Design on page 29

Run Settings for Verify Design

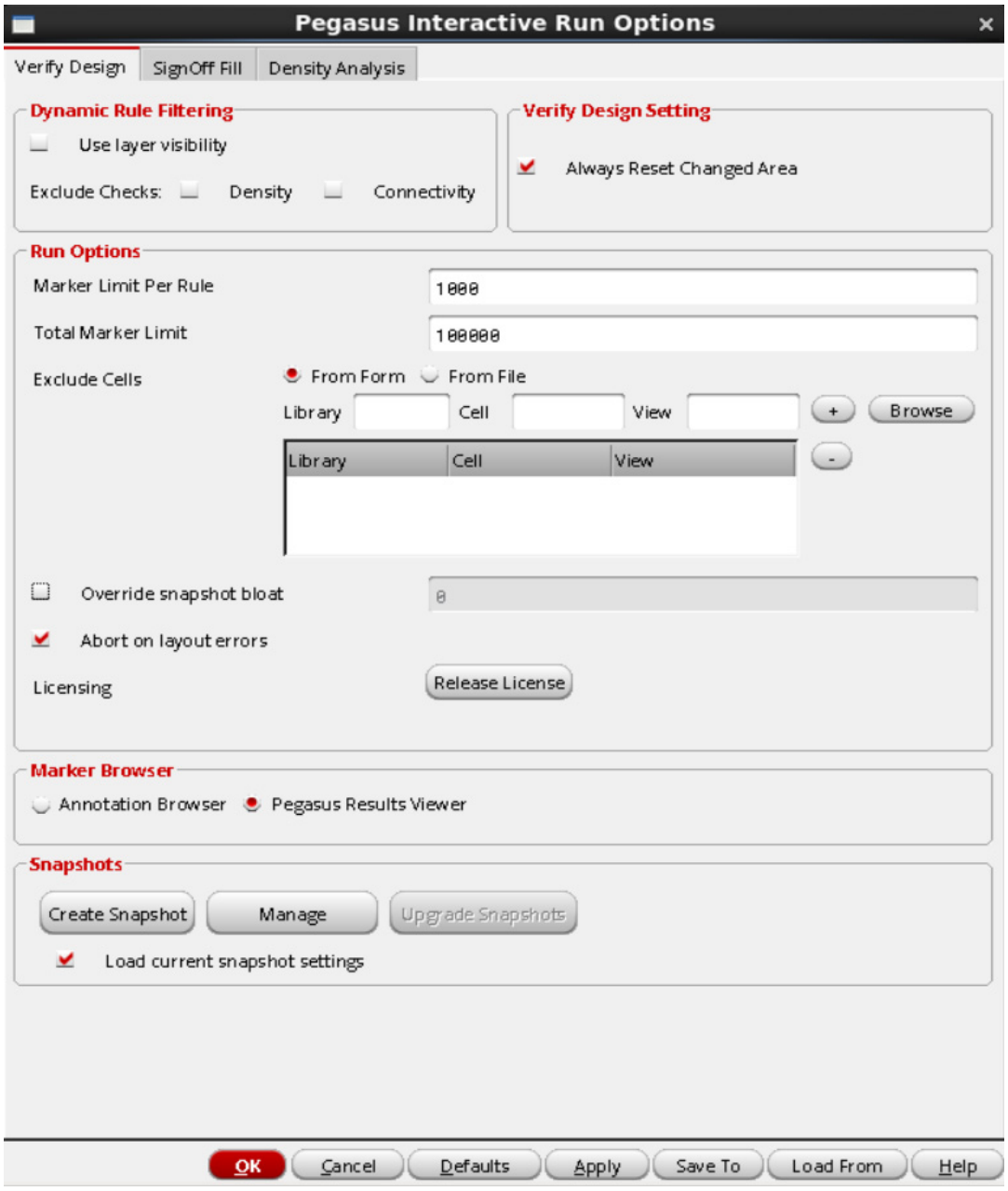
Following outlines the topics described in this section:

- Enable Dynamic Rule Filtering
- Customize Run options
- Set error browsing utility
- Snapshot Creation and Management

To interactively change current Pegasus Interactive behavior, you can invoke the *Pegasus Interactive Run Options* form by clicking on the *Pegasus Interactive Run Options* icon  on the toolbar.

All options described in this section are applicable to entire Virtuoso session only, and are not window specific. If new layout is opened within the Virtuoso session, it will inherit settings set

in the *Pegasus Interactive Run Options* form. Once Virtuoso session is closed, the options interactively set by you are not retained.



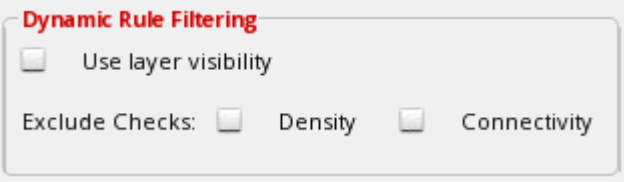
Dynamic Rule Filtering (DRF)

Pegasus Interactive has a built-in intelligence to select rules based on the layer visibility. In other words, you can ask Pegasus Interactive to select checks based on current visible layers

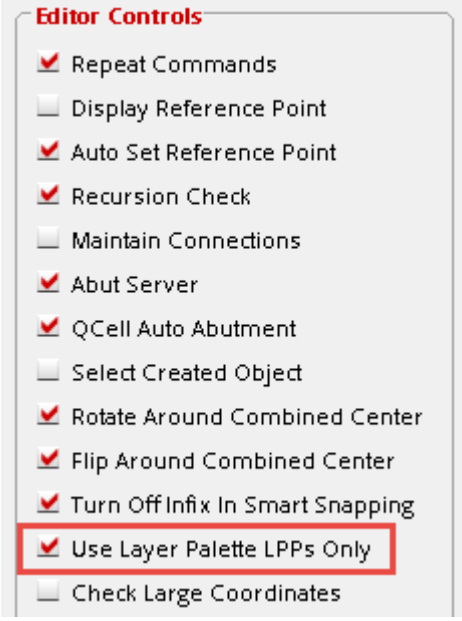
displayed in the *Palette*. To effectively use this feature, this dynamic rule filtering capability should be paired up with Virtuoso valid LPP utility. Then, Pegasus Interactive further performs rule selection based on current visible layers display status in the *Palette*. Essentially, it becomes a *What-You-See-Is-What-You-Check* utility when they are paired up. Tool makes the assumption that for any visible layer displayed in the *Palette* that matters to you, you only want to ensure editing does not introduce any unforeseen DRC violations which are avoidable during layout editing and planning.

In *Pegasus Interactive Run Options*, you can enable the *Use layer visibility* option located in the *Rules for Visible Layers* field. Once it is enabled, rules are selected based on visible layers to its adjacent layers, which may or may not be visible in the *Palette*. For example, if Metal1 is visible in the Palette, rules for Metal1 itself checks, Metal1-Vias checks as well as Metal1-Metal2 checks will be selected.

Furthermore, you can exclude *density* or *connectivity* checks by enabling *Exclude Checks: density* or *Exclude Checks: connectivity* options.



This option needs to be paired up with Virtuoso valid LPP utility. It can be set interactively through *Layout Editor Options* GUI. It is one of options under the *Editor Controls* field.



Its default option can also be set by using `validLppFilterOn` CDS environmental variable as following:

```
envSetVal("layout" "validLppFilterOn" 'boolean 't)
```

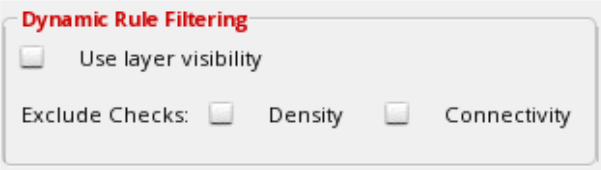
Furthermore, you can exclude *density* or *connectivity* checks by enabling *Exclude checks: Density* or *Connectivity* options. These options are independent of *Use layer visibility*.

Similarly, DRF default behavior can be controlled using `drdPegasusIntFunctionalPresets` CDS environmental variable as following:

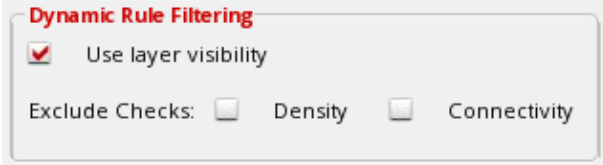
Note: Environment variable settings depend on Virtuoso version. If you are using ICADV123/ IC617 ISR21 or newer then `drdPegasusIntFunctionalPresets` environment variable

should be used. If you are using ICADV123/IC617 ISR20 or older then `drdPVSFunctionalPresets` environment variable should be used.

```
envSetVal("layout" "drcPegasusIntFunctionalPresets" 'string "Default")
```

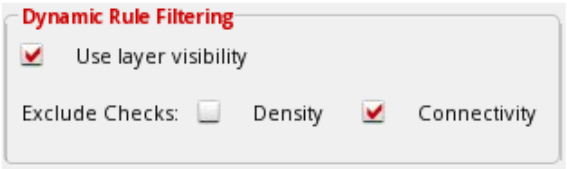



```
envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string "Any Visible")
```

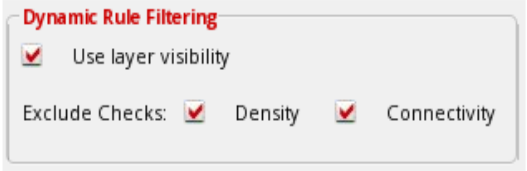



```
envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string "Any Visible nodensity")
```


```
envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string "Any Visible noconnect")
```

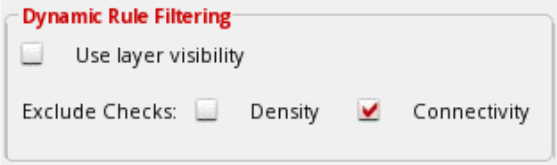



```
envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string "Any Visible noconnect nodensity ")
```



envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string " Ignore Visible nodensity")

envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string " Ignore Visible noconnect")



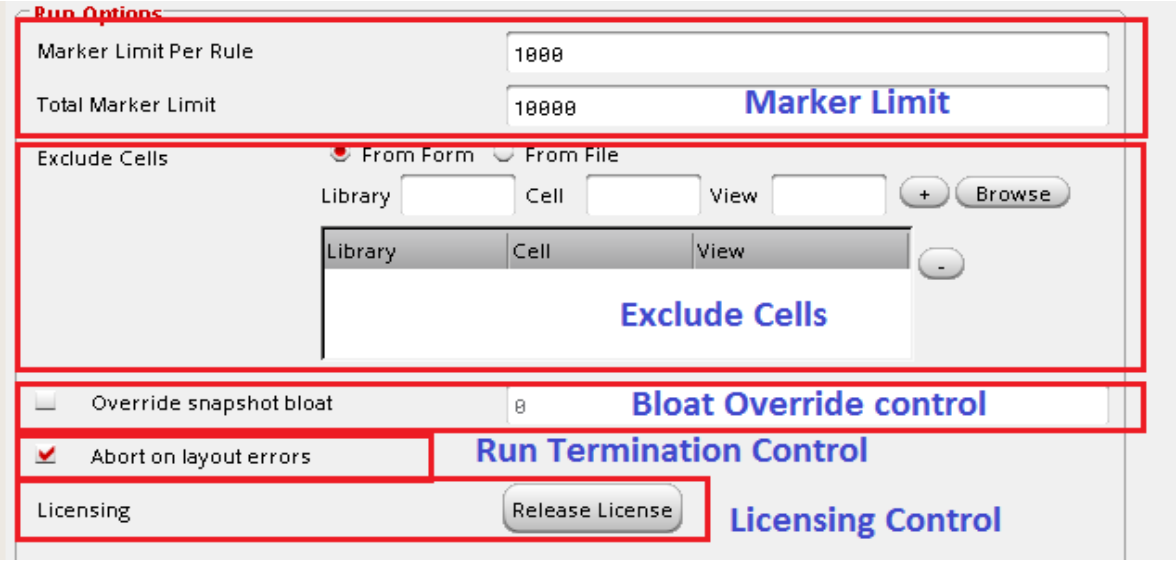
envSetVal("layout" "drdPegasusIntFunctionalPresets" 'string " Ignore Visible noconnect nodensity")

Run Options

Five elements are grouped under general run option field:

- Marker Limit
- Exclude Cells
- Bloat Override control
- Run Termination Control

- Licensing Control



These elements are described in following sections:

Marker Limit

For a given run regardless of operate modes, you can set a limit on the total violation counts in the *Total Marker Limit* text box. Once total count of markers exceeds this threshold, only maximum amounts of markers are generated and displayed in the layout. Furthermore, you can define maximum marker counts per rule interactively in the *Marker Limit Per Rule* text box.

In addition to the interactive setting of these marker limits, you can set the default values through following CDS environment variables:

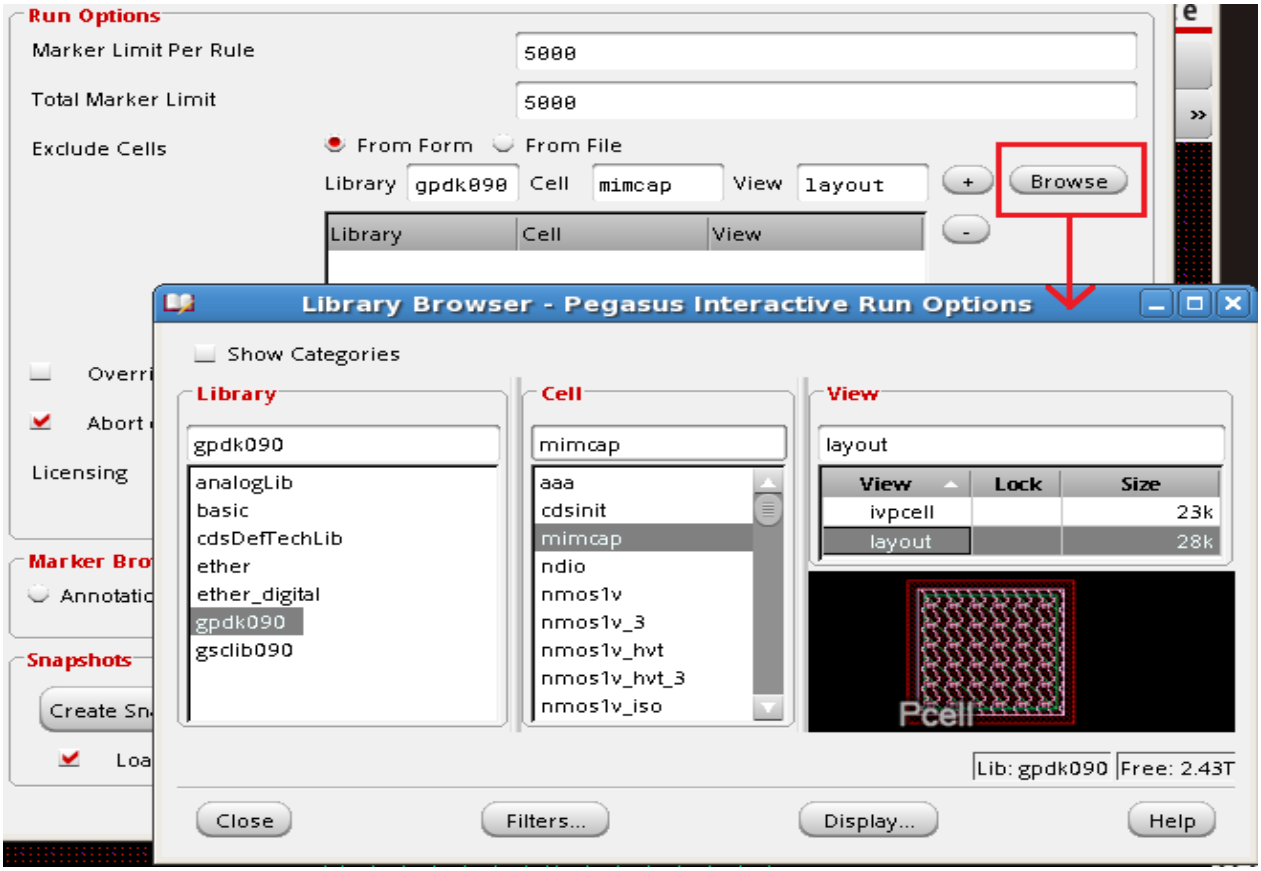
- `drdBatchVioLimit`
- `drdVioLimitPerRule`

Exclude Cells

You can also choose to exclude certain cells from a given run. There are two ways of declaring excluded cell:

- *From Form*: Library management user interface.
- *From File*: A file containing cellview information.

By selecting the *From Form* option, you can either fill cell information explicitly in the *Library*, *Cell* and *View* text boxes or through the *Library Browser* form. Latter automatically fills in the *Library*, *Cell* and *View* text boxes once you finalize selection in the library browser.

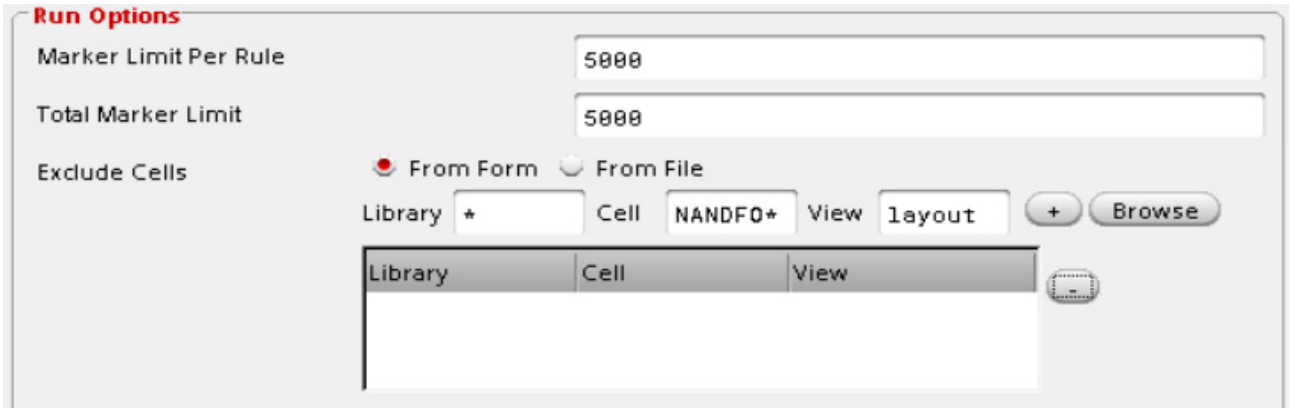


Also, if you would like to select a set of cells have similar naming convention, you can issue wildcard on any entry in *Library*, *Cell* and *View* fields.

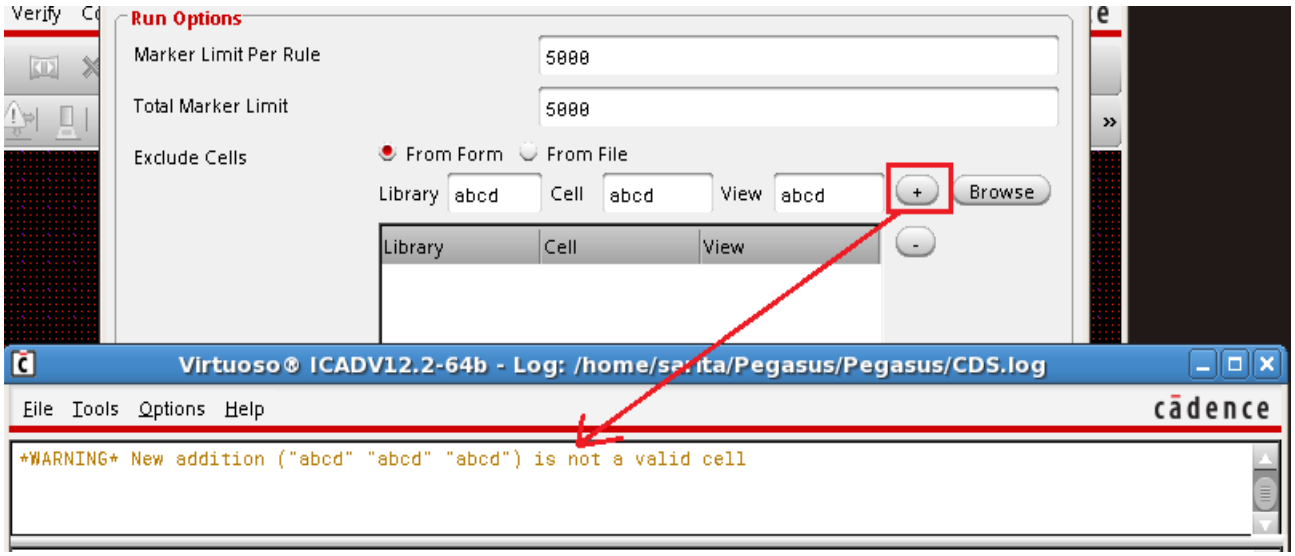
Two types of wildcards are supported:

- * : Match zero or more characters
- ? : Match exact one character

For example, you would like Pegasus Interactive to exclude all of cell has name starts with NANDFO in any library found, you can fill *Library*, *Cell* and *View* fields as following:

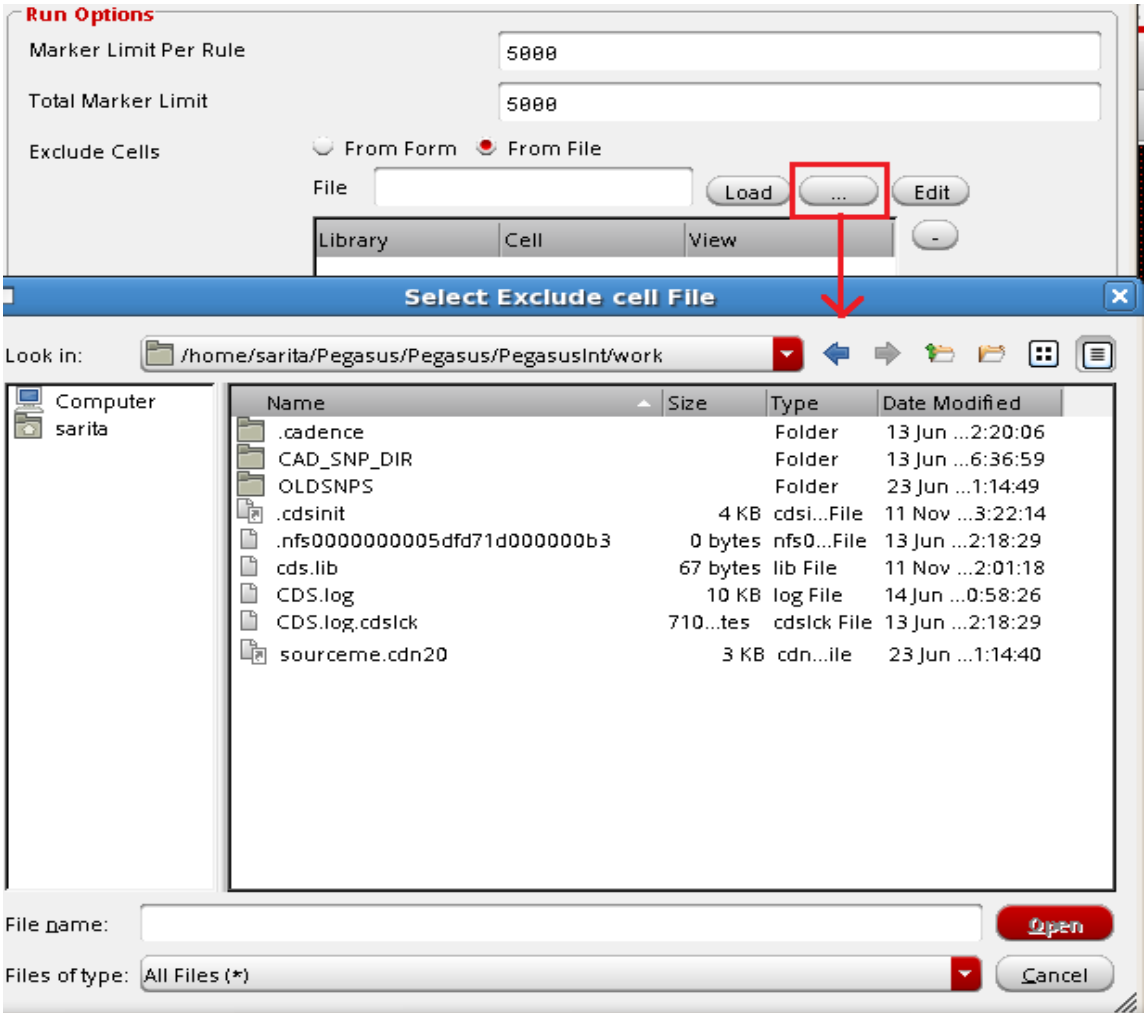


Commit the selection by clicking the plus sign (+) button. If the Information is insufficient, a warning message is issued in CIW. Following example shows entered cellview information is insufficient for Pegasus Interactive to process.



In general, Pegasus Interactive checks cellview information against libraries declared in cds.lib. If it is not in cds.lib, it is considered not accessible by Pegasus Interactive.

You can also declare a list of excluded cells in the file and then load the content through *From File* option. By clicking the "…" button. The *Select Exclude cell File* form is available for you to select the file interactively.



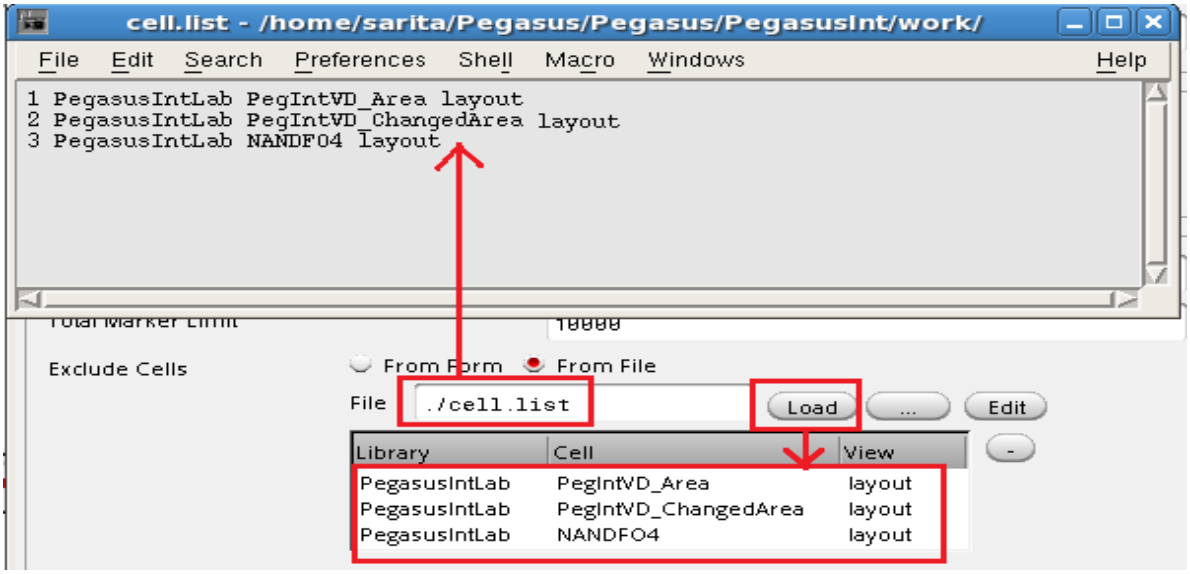
Once the file is selected, its path will be populated under *File* text box. To commit the cell selection via file, click *Load*. During the loading, Pegasus Interactive checks:

- Availability of the cell
- Cell duplication
- Cell declaration syntax

If any insufficient information is detected, warning messages are displayed in CIW.

Correct syntax for cell declaration in the file should be in the order of library, cell and view and they are delimited by space between declarations. Also, you can declare wildcard, either `?` or

`*`, to conduct batch cell selection. Following example shows correct syntax declared in `cell.list`. Complete cell information is loaded into exclude cell table after clicking *Load*.



You can sort column alphabetically in the *Exclude Cells* table by clicking the column header: *Library*, *Cell* or *View*.

To remove one or multiple cells from the *Exclude Cells* table, select one or more rows by clicking the minus sign (-) button.

You can declare default exclude cells file path through `PegasusInt_ExcludeCellFile`.

Bloat Override Control

Pegasus Interactive allows you to check DRC on certain area of the cellview. Basically the final area of checking is determined by the maximum bloat value found between the value stored in the snapshot.

However, you can override bloat value for current Pegasus Interactive session without re-generating of snapshot using *Override snapshot bloat* option of the *Pegasus Interactive Run Option* form:

By default, *Override snapshot bloat* is disabled. Once you enable it you can enter any positive float value including "0". If you enter "0" bloat value, it would mean that you want to check area "as-it-is".

If you specified override snapshot bloat value, then following message will be printed to Virtuoso CIW window prior Pegasus Interactive DRC run:

```
Pegasus Interactive: Bloat value for current snapshot (3.00) is now set to 5.00
(inspect Run Options form)
```

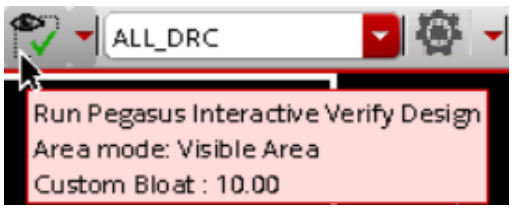
After Pegasus Interactive DRC run regardless operate modes, run complete message includes the bloat value used during the run. Example:

```
*INFO* Pegasus Interactive
Cellview Name: top
Snapshot: Default
Bloat Value: 5
Summary: 12 markers created
```

Also, enabling of *Override snapshot bloat* triggers the *Verify Design* icons update in the Pegasus Interactive toolbar for all of opened layout window:

Verify Design Area Modes	Override Snapshot Bloat	Toolbar Icon
Current Cellview	OFF	
Changed Area	OFF	
Visible Area	OFF	
Current Cellview	ON	
Changed Area	ON	
Visible Area	ON	

In addition, the bloat value is shown as tooltip of toolbar icon:

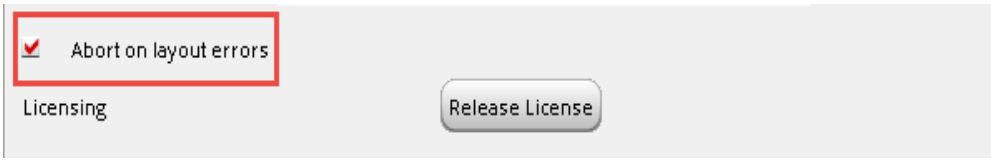


Run Termination Control

By default, Pegasus Interactive terminates the run if it detects following errors:

- Missing cell masters
- Mismatch rule precision values with the input database precision

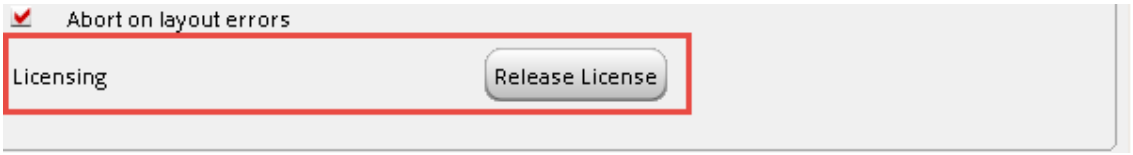
This default behavior can be altered through the *Abort on Layout Error* option.



To continue DRC despite of outcome of layout validity, deselect the check box. In this case, errors found during layout validity check will be treated as soft errors and run proceeds as it-is.

Licensing Control

As described in *Licenses Requirements and Behaviors*, Pegasus Interactive checks out licenses and holds them until termination condition is met. Within a Virtuoso session, you can release the license anytime through the *Release License* button. It terminates any existing Pegasus Interactive session. Only license(s) documented in snapshot header is released. License pertaining to Pegasus RV is controlled by its own activity process.



Marker Browser

This field allows you to switch between two different types of browsers. If there are any existing OA markers created by Pegasus Interactive or error highlighted on the layout, they are erased when you switch from Pegasus Results Viewer to Annotation Browsers or vice versa.

Since Pegasus Interactive retains last run result in an ascii format, you can reload this data any time by invoking Pegasus Results Viewer through the toolbar. On the contrary, you cannot view last run results via Annotation Browser on demand. For example, Pegasus Interactive's latest run generates OA markers on the layout. You would like to first view results in Pegasus Results Viewer and then use Annotation Browser for DRC fixing afterward. In this case, error

highlights are generated on the layout and errors are automatically reported in Pegasus Results Viewer when you toggle from Annotation Browser to Pegasus Results Viewer. However, OA markers are not restored once you toggle back to Annotation Browser from Pegasus Results Viewer.

For details on interactive setting browser preference and for details on each browser behavior and customization, refer to Error Viewing.

Snapshots

The content of snapshot setting field can be customized. The intent of this use model is to give CAD administrator control on the snapshot accessibility to target users. CAD administrator determines whether snapshot should be generated or maintained either globally or locally. Based on this decision, snapshot field content display can be controlled through following shell environmental variables:

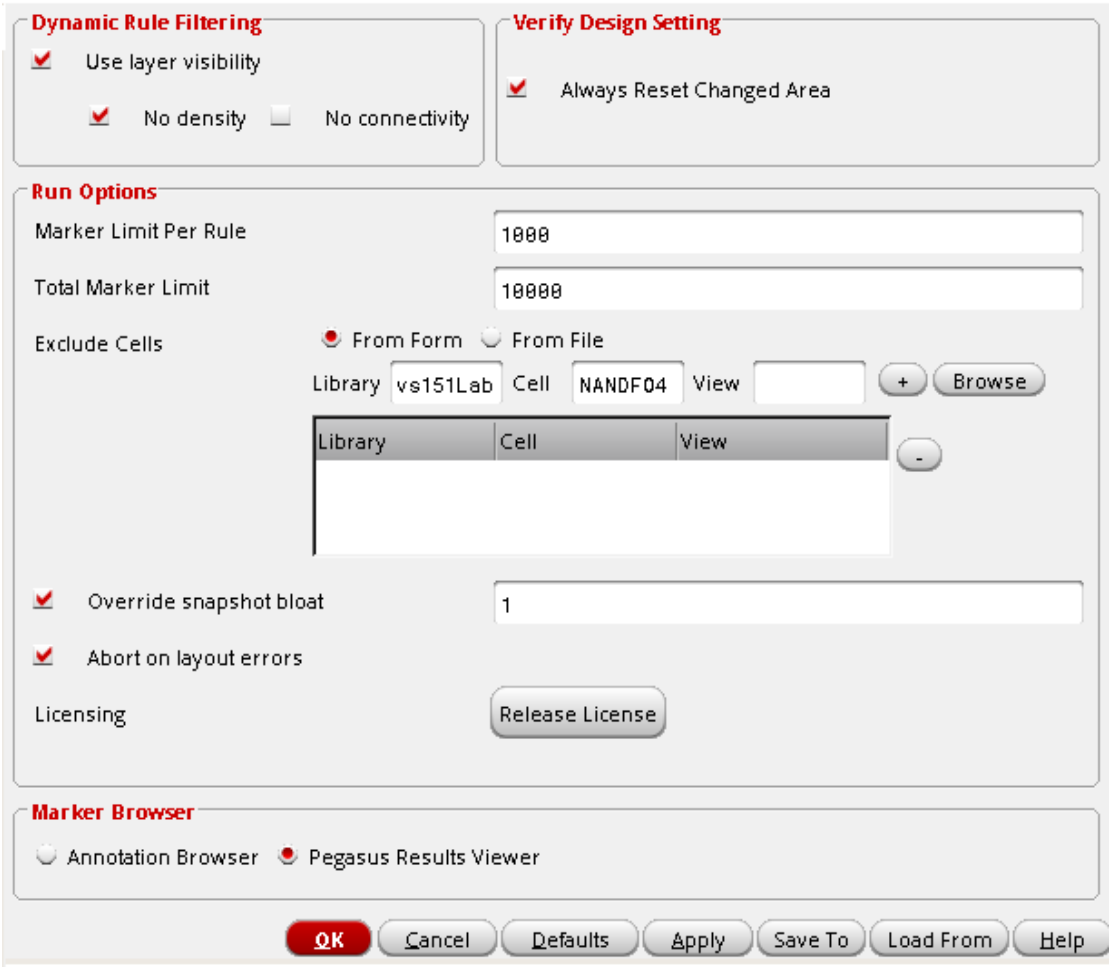
- `PegasusInt_Maintenance`
- `PegasusInt_EnableSNPCreation`
- `PegasusInt_EnableSNPManagement`

Condition One: Global Snapshot Control by CAD Only

In this case, CAD wants a full control of snapshot creation and maintenance. Since users are not allowed to create local snapshot, snapshot creation trigger can be hidden through `PegasusInt_EnableSNPCreation`. Depending on the number of snapshot directories, CAD can decide whether to disable or enable Snapshots management field through `PegasusInt_EnableSNPManagement`.

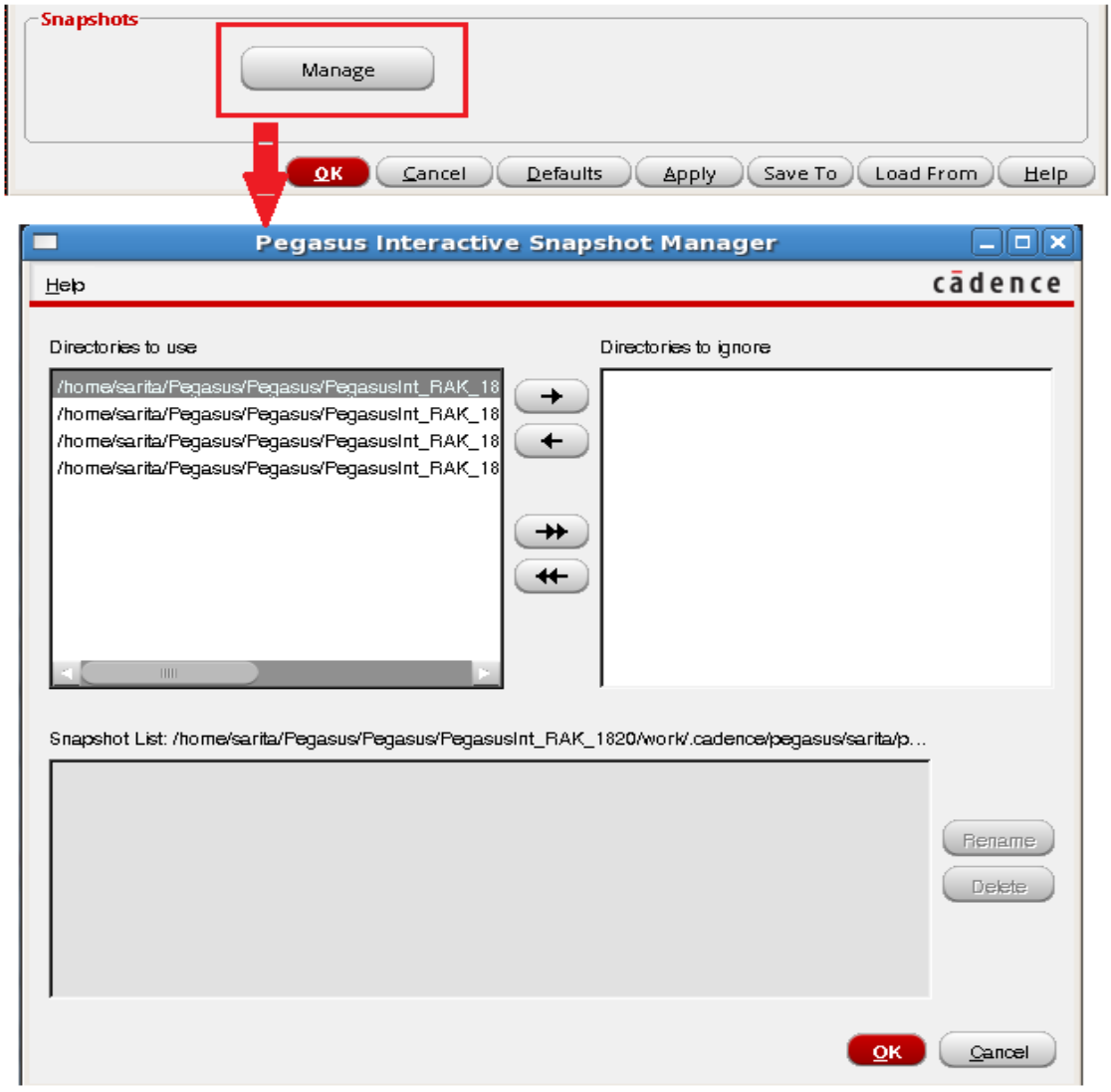
Following example shows the complete removal of the `Snapshots` field by setting both shell environment variables to `no`.

```
setenv PegasusInt_EnableSNPCreation no
setenv PegasusInt_EnableSNPmanagement no
```



If there are more than one snapshot directories maintained by CAD and end user can access them, the CAD can enable snapshot management utility as following:

```
setenv PegasusInt_EnableSNPCreation no
setenv PegasusInt_EnableSNPManagement yes
```

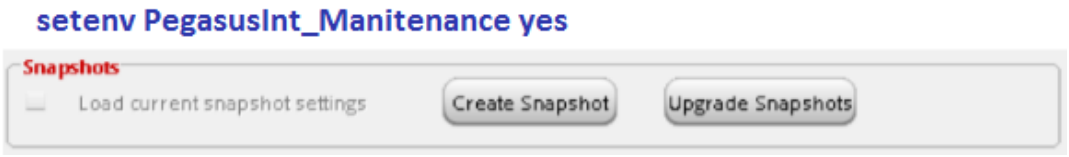


For details of snapshot management, refer to Manage Snapshots.

Scenario Two: Local Snapshot Control by User Only

CAD only maintains technology setup and user is responsible of creating and maintaining local snapshots. Snapshot management utility needs to be available for you to upgrade any

out of sync snapshots detected by Pegasus Interactive. This utility can be enabled through `PegasusInt_Maintenance`.

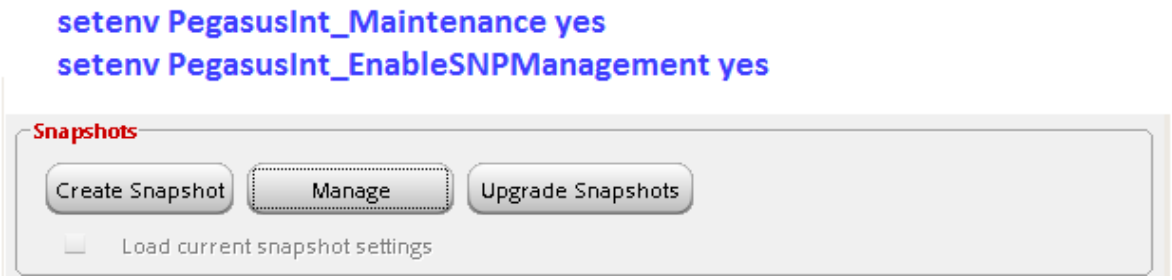


For details of creating snapshot, refer to Quick Start: Create Snapshot.

For details of upgrading snapshot, refer to Upgrading Snapshots.

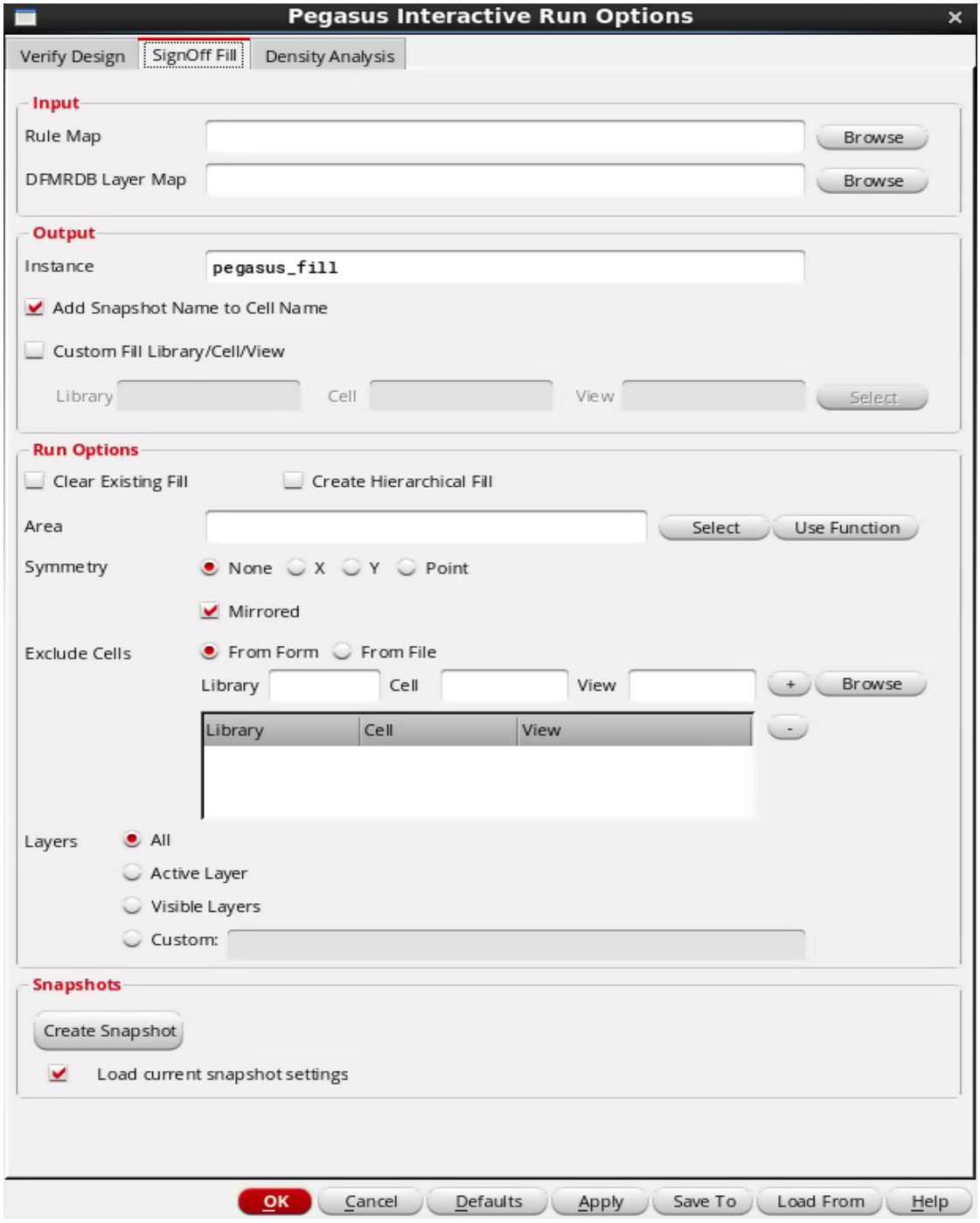
Scenario Three: Global and Local Snapshot Control by User

For this use model, CAD provides a set of generic global snapshots and user can create custom snapshots that are derived from these global snapshots.

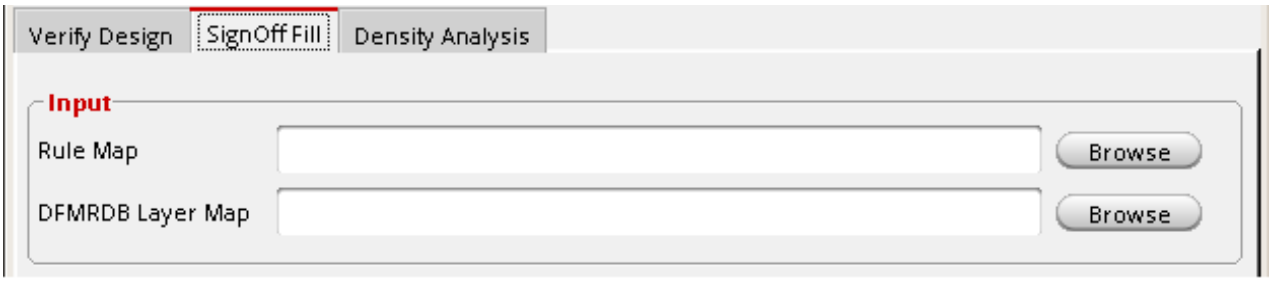


Run Settings for SignOff Fill

Click the *SignOff Fill* tab to define the run settings for SignOff Fill.



Input Section



- *Rule Map*: Enter the name of the sign-off rule map file or click *Browse* to select it. Use this file when fill rule deck uses the `output` PVL command to output metal fills. The syntax of the file is:

  ```
  <rulename> dfII_layer dfII_purpose
  ```

  Example:

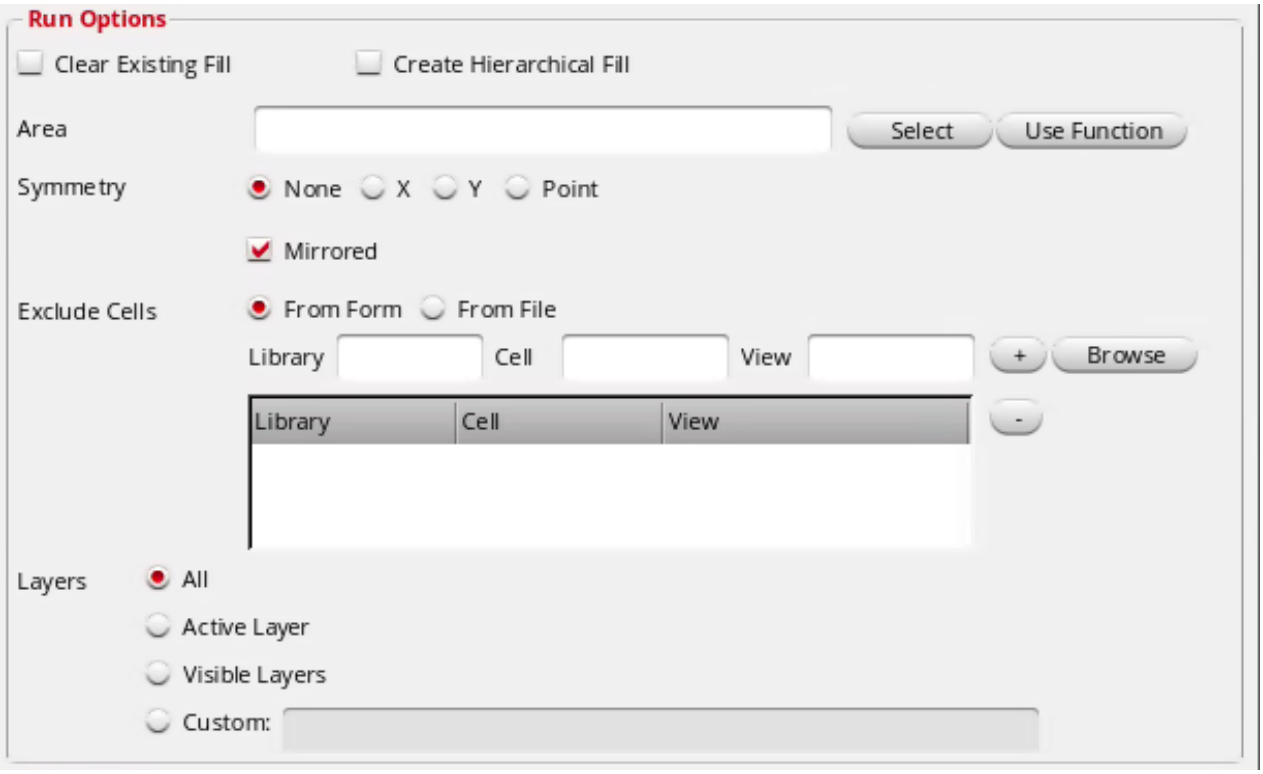
  ```
  fill.rul
  …
  rule m1_fill_output { copy m1_fill }
  output -drc m1_fill_output -gds 15 35 fill.gds -output all

  rule m2_fill_output { copy m2_fill }
  output -drc m2_fill_output -gds 17 35 fill.gds -output all

  rulemap.file:
  m1_fill_output Metal1 fill
  m2_fill_output Metal2 fill
  ```

- *DFMRDB Layer Map*: By default Pegasus Interactive SignOff fill uses technology *layermap* file to read generated dummy fills and map them into Virtuoso to specific layer/view purpose. You can redefine it using this field.

Output

- *Instance*: Specifies prefix of instance name of fill cell. Pegasus Interactive fill puts all generated fills into specific cell and then inserts this cell as instance into the block. Use this field to redefine instance name prefix. Default instance name prefix is `pegasus_fill`. Full instance name is `<prefix>_RUN_FILL`.
- *Add Snapshot Name to Cell Name*: Select the check box to add the snapshot name to the name of the cell with generated fills as secondary prefix. When enabled, the fill cell name is `pegasus_fill_<snapshot name>_<original cell name>`.
- *Custom Fill Library/Cell/View*: Select the check box to customize the library, cell and view of the cell with generated fills.

Run Options



- *Clear Existing Fill*: Select the check box to delete fills, generated by previous SignOff Fill runs before the new run. The tool deletes existing fills, generated by previous SignOff Fill runs. Do not select the check box to retain existing fills in each new run.
- *Create Hierarchical Fill*: Select the check box to create Hierarchical fills.
- *Area*: Enter the coordinates or click *Select* to generate dummy fills for the specified area. Dummy fills will not be generated outside this specified area. You can define `pgssUserSelectedFillArea()` function to select area to fill. If you have defined this

function then clicking on the *Use Function* button fills the area field with already defined coordinates. If you have not defined the function then an error message is displayed.

- *Symmetry*: Select the symmetry value from the following options based on which the fill shapes will be generated. To use any option other than *None*, the *Area* option must be selected. If the *Area* option is not selected and area is not defined, an error message is displayed.
  - *None*: This is the default mode. No symmetry operation will be performed in this case.
  - *X*: When X is specified, the symmetry operation is performed along the X-axis.
  - *Y*: When Y is specified, the symmetry operation is performed along the Y-axis.
  - *Point*: When point is specified, the symmetry operation is performed along the axis point.
  - *Mirrored*: When you select *Mirrored* check box, the *X*, *Y*, and *Point* options perform Mirrored symmetry operation. The following figure shows the result of mirrored and non-mirrored *Symmetry* option result:

Option	Non-Mirrored	Mirrored
X	$\begin{vmatrix} R0 \\ 0 \end{vmatrix}$	$\begin{vmatrix} R0 \\ MX \end{vmatrix}$
Y	$\begin{vmatrix} 0 & R0 \end{vmatrix}$	$\begin{vmatrix} MY & R0 \end{vmatrix}$
Point	$\begin{vmatrix} 0 & R0 \\ 0 & 0 \end{vmatrix}$	$\begin{vmatrix} MY & R0 \\ R180 & MX \end{vmatrix}$

- *Exclude Cells*: Use this option to exclude cells from fill insertion. The *Exclude Cells* option is similar to the *Exclude cells* option of *Run Settings for Verify Design*. For detailed description, see: Exclude Cells on page 35.
- *Layers*: Use this option to define layers to be used in the run.
  - *All*: Select this radio button to process all layers.
  - *Active Layer*: Select this radio button to perform the run on the currently active layer on Virtuoso layer palette.
  - *Visible Layer*: Select this radio button to generate dummy fills for layers, which are visible on Virtuoso layer palette.
  - *Custom*: Select this radio button to generate dummy fills for the custom layers defined in the text box. The syntax for specifying layer in the *Custom* layer field is the following:

`<layername> ... <layername>`

Wildcards are supported.

Example 1:

```
Metal1
```

Fills will be generated for `Metal1` layer.

Example 2:

```
Metal1 Metal2 Metal3
```

Fills will be generated for `Metal1`, `Metal2`, and `Metal3` layers.

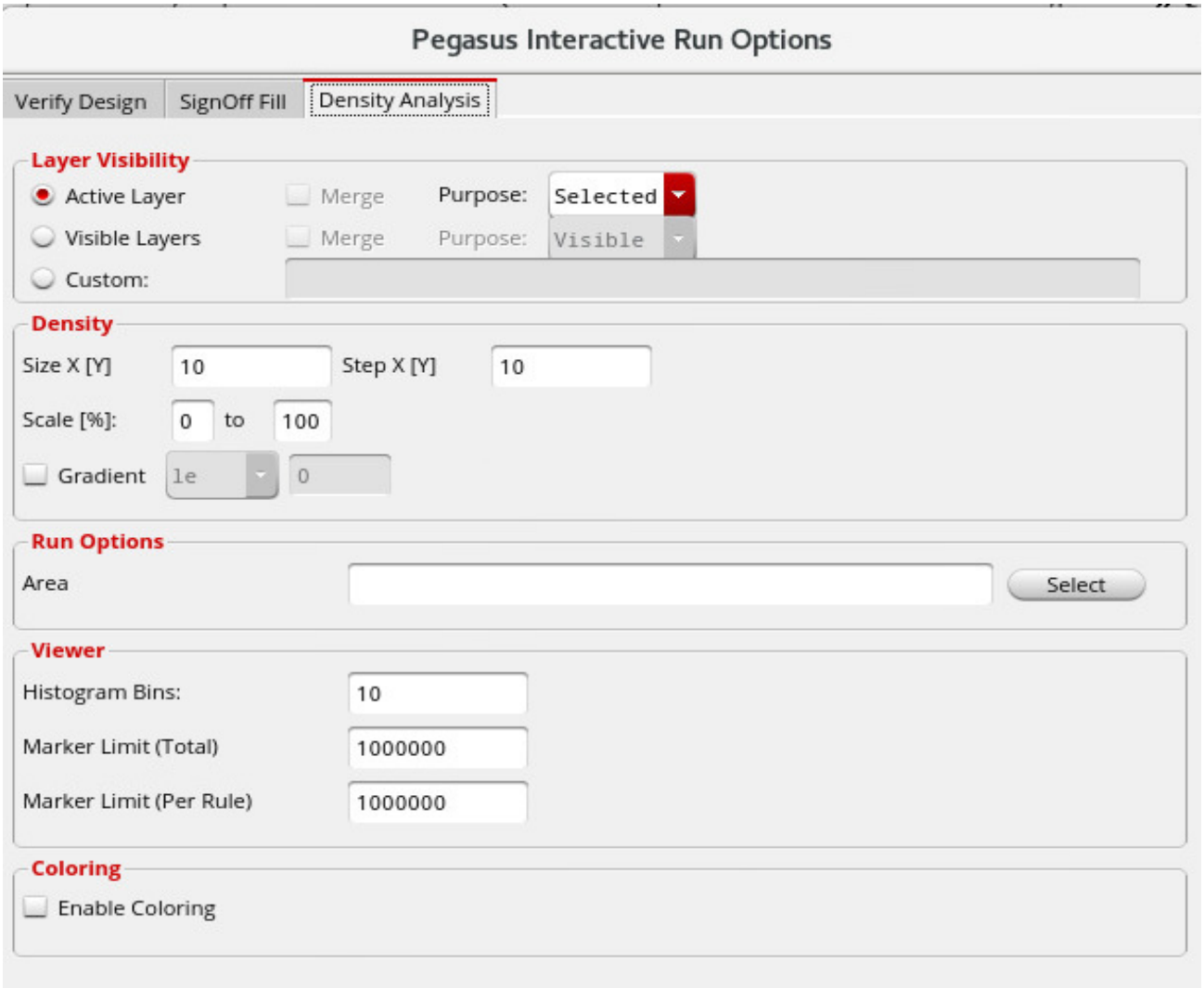
Example 3:

```
Metal*
```

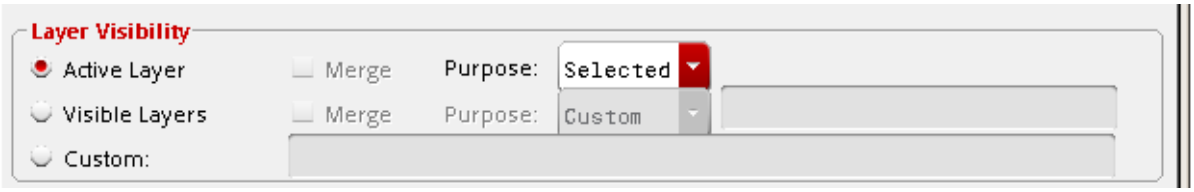
Fills will be generated for all layers, which start with `Metal`.

Run Settings for Density Analysis

Density analysis generates density results from Pegasus Interactive toolbar for layers of interest. Rule deck is not required. Results are shown in Pegasus Results Viewer as heatmap and histograms.



Layer Visibility



- *Active Layer*: Select this radio button to perform density analysis on the active layer on Virtuoso Palette.
  - *Purpose*: By default density results will be generated only for layer/purpose pair, selected on Virtuoso Layer palette. Use the *Purpose* drop-down list to redefine list of layer purpose.
    - *All*: Density results will be generated for all purposes of selected layer. Each layer/purpose result will be shown in separate tabs of DRC RV.
    - *Selected*: Density results will be generated for selected purpose only.
    - *Custom*: Density results will be generated for purposes defined in the custom field. Each layer/purpose results will be shown in separate tabs of DRC RV.
  - *Merge*: This option is valid when the *Purpose* field is either *All* or *Custom*. Select this check box to merge the result for layer purposes into a single tab. This is useful when you have metal and fills in different purposes (Metal1/drawing and Metal1/fill) and want to generate density results for merged drawing and fill purposes.
- *Visible Layers*: Select this radio button to perform density analysis for layers, which are visible on Virtuoso layer palette.
  - *Purpose*: By default density results will be generated only for layer/purpose pair, visible on Virtuoso Layer palette. Use the *Purpose* drop-down list to redefine list of layer purpose.
    - *All*: Density results will be generated for all purposes of selected layer. Each layer/purpose result will be shown in separate tabs of DRC RV.
    - *Visible*: Density results will be generated for visible purposes only.
    - *Custom*: Density results will be generated for purposes defined in the custom field. Each layer/purpose results will be shown in separate tabs of DRC RV.
  - *Merge*: This option is valid when the *Purpose* field is either *All* or *Visible*. Select this check box to merge the result for layer purposes into a single tab. This is useful when you have metal and fills in different purposes (Metal1/drawing and Metal1/fill) and want to generate density results for merged drawing and fill purposes.
- *Custom*: Select this radio button to perform density analysis for the custom layers defined in the text box. The syntax for specifying layer/purpose in the *Custom* layer field of the *Density Analysis* form is the following:

  Example 1:

  ```
  Metal1/drawing
  ```

  Density Analysis of `Metal1/drawing` will be output.

Example 2:

```
(Metal1/drawing Metal2/drawing) Metal3/drawing
```

Density Analysis histograms of `Metal1/drawing` and `Metal2/drawing` will be merged to create one tab in the viewer. Density Analysis histogram of `Metal3/drawing` would be created as a separated tab in the viewer.

Example 3:

```
Met*/drawing
```

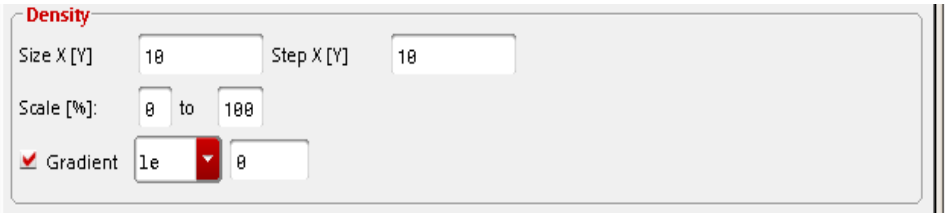
All histograms of layers having "Met" in their name and having drawing purpose will be created as separate tabs.

Example 4:

```
(Met*/drawing)
```

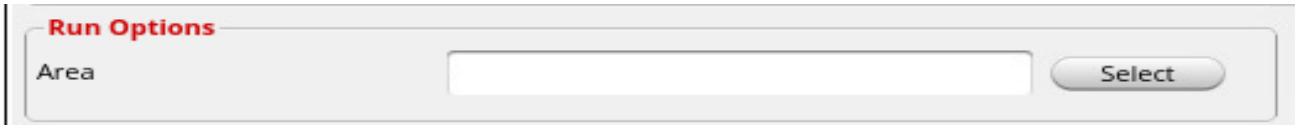
All histograms as explained in Example 3 will be merged in a single tab.

Density



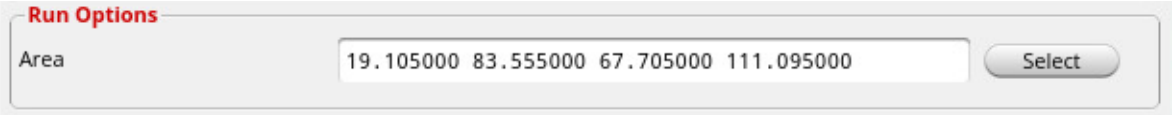
- *Size X[Y]* and *Step X[Y]*: Specify the size of the rectangle window, which is used to calculate density. This is equivalent of the `-window` option of PVL command `density`. Default window size is 10x10.
- *Scale [%]*: Specify the scale range between which you want to check the density.
- *Gradient*: Optional keyword specifies the gradient value that windows must satisfy. This is equivalent of the `-gradient` option of PVL command `density`. For more details, refer to *Pegasus Developer Guide*.

Run Options



- *Area*: Enter the coordinates or click *Select* to select the area to perform density analysis on. Density results will not be generated outside this specified area.

When you complete the selection, the *Area* field is populated as shown below:



Viewer

- *Histogram Bins*: Specify the number of histogram data bins.
- *Marker Limit (Total)*: Specify the number to set a limit on the total violation counts shown in the histogram.
- *Marker Limit (Per Rule)*: Specify the number to set a limit on the total violation counts per rule shown in the histogram.

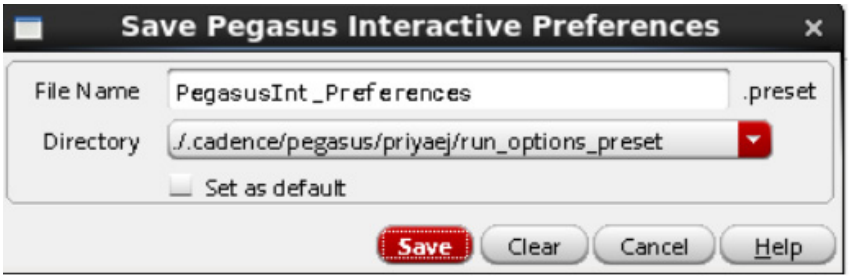
Coloring

- *Enable Coloring*: For a given Layer-Purpose pair, a set of polygons can have additional color attributes assigned on them: color and color state in the layout. By selecting *Enable Coloring* check box, Pegasus Interactive understands color implications set on these polygons.

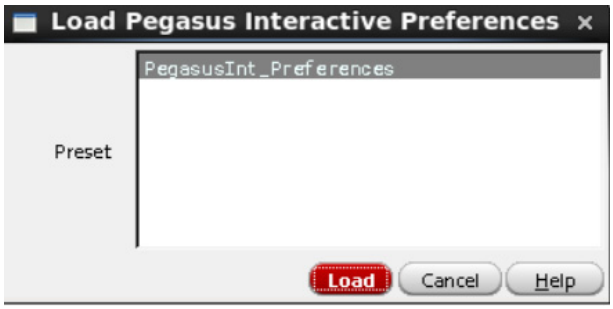
Buttons at the Bottom of the Form

- *OK* - Closes the Preferences form, saving any changes you have made.
- *Cancel* - Closes the Preferences form without saving any changes.
- *Defaults* - Loads the default form setting.

- *Apply* - Applies any preference changes that you have made. This does not close the form.
- *Save to* - Saves the changes you have made to a preset file. You can use that preset whenever you want to use the settings set in the file. Clicking the button opens the *Save Pegasus Interactive Preferences* form:



  - *File Name*: Enter the file name for the Pegasus Interactive Run Options form settings you have made.
  - *Directory*: Select the location where you want to save the file. The *Directory* drop-down list allows you to set the directory path that preset is going to save to. By default the *Directory* field allows you to select two locations: the local `.cadence/pegasus/$USER` directory and `.cadence/pegasus/$USER` at the `$HOME` directory. However, its drop-down items can be expanded using `CDS_WORKAREA` environment variable.
  - *Set as default*: Select this check box to save the settings as default.
  - Click *Save* to save the settings.
- *Load From* - Loads the form settings from an already saved preset file. Clicking *Load Form* opens the *Load Pegasus Interactive Preferences* form:



  The form lists presets saved by user in different locations. These locations and search order are defined by setup.loc file, which

Cadence provides in $PEGASUS_INSTAL_DIR/share/cdssetup/setup.loc. Default setup.loc file defines search order as follows:

. : cwd should always be searched first.

$CDS_WORKAREA: User workarea if defined.

$CDS_SEARCHDIR: This is set by various tools during tool startup.

$HOME

$CDS_PROJECT: Project storage area, ignored if not defined.

$CDS_SITE : Site Setup Info - default is $CDS_INST_DIR/share/local.

$(compute:THIS_TOOL_INST_ROOT)/share: Cadence Default Setup Info.

- *Help* - Brings up context-sensitive help.

3

All About Design Rules: Snapshot

This chapter covers basic concepts of snapshots, its definition and the type of snapshot directories. The content of this chapter is organized as follows:

Snapshot Definition and Files

A snapshot captures design rules in a compiled binary format, which only Pegasus understands. This format allows Pegasus the direct access to design rules during DRC without re-compiling design rules from the rule deck. The advantages of using snapshot over rule deck are:

- Run time improvement since rule compilation are eliminated
- Fast switching between different snapshots
- Ensure rules stored in the snapshot reflect the creator's initial intent
- Re-usability of custom rule sets

Following table gives an overview of these snapshot files.

Snapshot Files Outline

Snapshot Files	Description
SnpName.preset	■ Preset file that contains Snapshot Creator GUI configuration setting for a given snapshot
SnpName.tag	■ Reference for Dynamic Rule Filtering
SnpName.cap	■ Cap file contains default rule selectivity, rule caption, rule types and rule description. ■ Reference for Dynamic Rule Filtering
SnpName.snp	■ Snapshot: the compiled version rule deck
SnpName.layermap	■ Technology layer mapping file
SnpName.objectmap	■ Pegasus Interactive uses this reference to lookup OA specific object to its equivalent stream pair
SnpName.sum	■ Pegasus Interactive uses this file to check the validity of overall snapshot reference files.

In order for Pegasus Interactive to work properly, a set of snapshot related files are created by the tool either through interactive or batch snapshot creation. Seven files are created by Pegasus Interactive and the filename is formatted as snapshot name appended with specific file type suffix: .preset, .tag, .cap, .snp, .layermap, .objectmap and .sum.

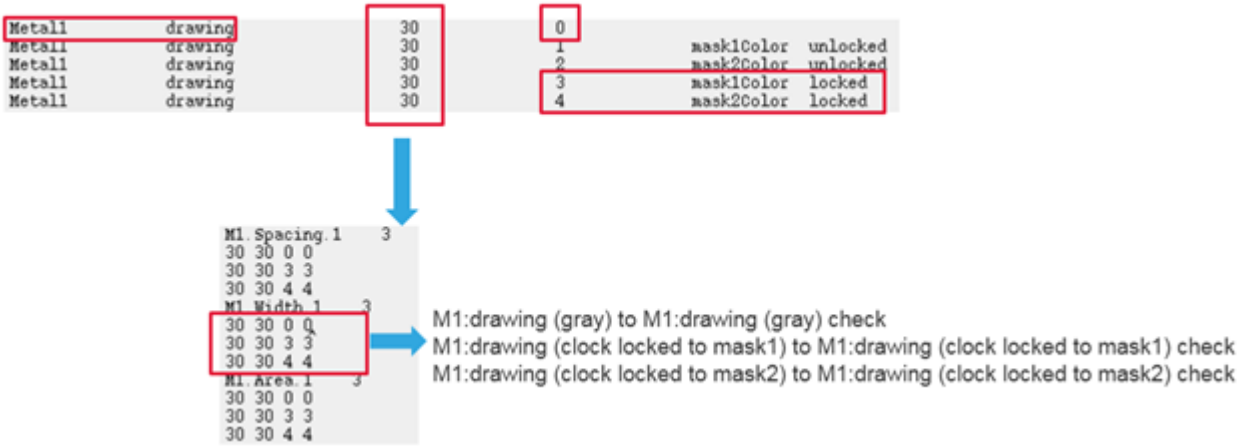
For dynamic rule selection, Pegasus Interactive relies on information collected from .tag, .cap and .layermap /.objectmap. Since polygons drawn in Virtuoso are represented by Layer-Purpose pairs defined in the technology file, Pegasus Interactive transfers these LPP specific polygons to their Layer-Datatype stream pair equivalents and then passes them directly to Pegasus. This is a direct communication between Virtuoso and Pegasus through Pegasus Interactive interface socket. In addition to transporting geometry data, Pegasus Interactive performs rule selection based on layer visibility by relying on information stored in the .cap and .tag files. The .tag file contains rule specific layers check lookup table. All the layers are represented by Layer-Datatype stream pair. Following is the general syntax declared in .tag file.

```
<Rule Caption> <# of different layer pair's check>
<LayerStream1#> <LayerStream2#> <Datatype1#> <Datatype2#>
:
```

Following example shows three different rule captions: M1.Spacing.1, M1.Width.1, and M1.Area.1. There are also three layer check conditions:

- Gray color M1:drawing to Gray M1:drawing
- Mask1 specific M1:drawing to Mask1 specific M1:drawing
- Mask2 specific M2:drawing to Mask2 specific M1:drawing

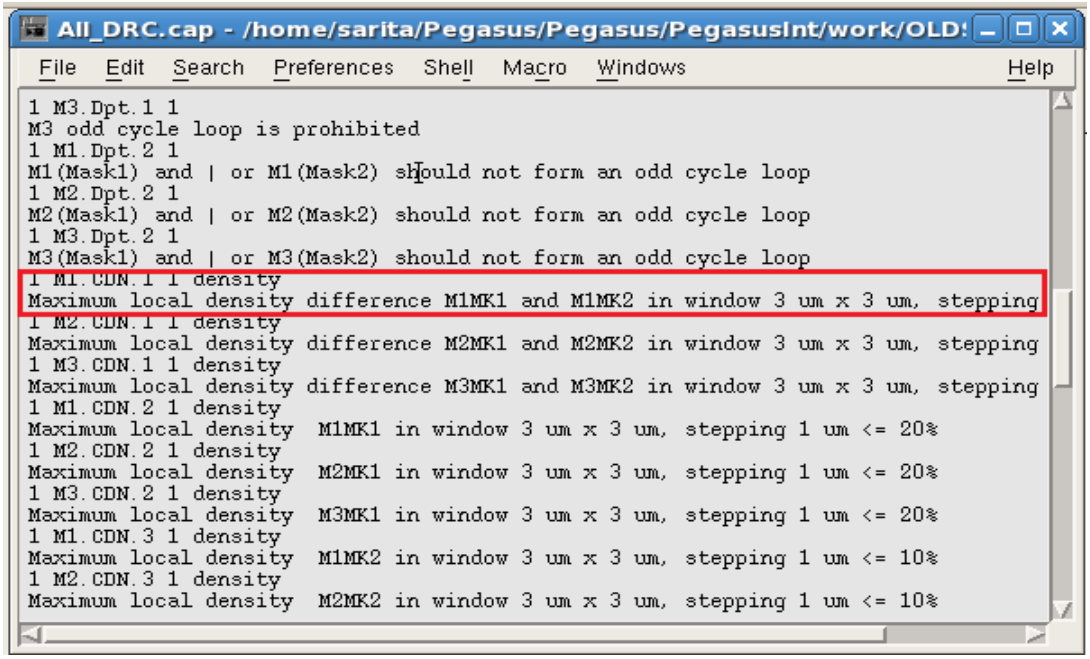
Rule is selected for DRC if any of the above interlayer check combinations is detected in the Palette.



For the .cap file, it contains rule types that Pegasus Interactive refers to for either density or connectivity type rules exclusion. Below example shows a typical content of .cap file. Its rule status' syntax is composed as following:

```
<Select Tag> <Rule Name> <Rule Caption's Line Count> <Rule Type>
```

```
<Rule Caption>
:
```



The .preset file is the configuration file that contains the *Snapshot Creator Form* setting. It is only for GUI application and is not understood by Pegasus engine. The advantage of having this .preset file is allowing you to load settings that generate current active snapshot while launching Snapshot Creator GUI. In addition to .tag, Pegasus Interactive refers to .cap file to ensure rule selection only on rules existed in the snapshot (.snp).

Finally .sum is for Pegasus Interactive to check snapshot files completeness. For a given snapshot validity check, Pegasus Interactive does not populate it in the snapshot combo field if file information recorded in .sum mismatches from its detected files.

Snapshot Directory Types and Customizations

For the interactive snapshot creation, Pegasus Interactive by default stores generated snapshot files under `.cadence/pegasus/<userid>/presets/pegasusint/snapshot`. This is defined as the local snapshot directory. Furthermore, the local snapshot directory is explicitly created per `userid`. For a given working space, there may be more than one user specific snapshot directories stored under `.cadence/pegasus`. However, Pegasus Interactive only checks and loads snapshots from the snapshot directory matched current user information that is queried through `'echo $user'` or `'whoami'`.

During Pegasus Interactive initialization, it first examines the content of this snapshot directory. If any pre-existing snapshot is found, it proceeds to perform snapshot files validity and version checks. If the potential snapshot passes these checks, it is included in the Pegasus Interactive's snapshot drop-down list. Otherwise, Pegasus Interactive issues warning messages in CIW and this non-compliant snapshot is excluded from the snapshot drop-down list.

In general, snapshot creation is a one-time setup. Re-usability of pre-existing snapshots in the local snapshot directory is guaranteed as long as the snapshot passes snapshot compatibility and file validity checks.

Pegasus Interactive allows you to define one or multiple snapshot directories through `PegasusInt_SnapshotsDirs`. Once this shell environmental variable is set, Pegasus Interactive checks and honors the contents of snapshot directories declared through this shell variable. If the snapshot directory is not available, Pegasus Interactive creates the snapshot directory automatically. However, this setting overrides the default snapshot directory. In other words, if default local snapshot directory path is not part of `PegasusInt_SnapshotsDirs`, Pegasus Interactive excludes it from mandatory checks.

To ensure the default local snapshot as part of user-defined snapshot directories, you can include it as part of declaration as shown below in .cdsinit:

```
;Set Snapshot directory
localsnpdir = strcat(getShellEnvVar("PWD") "/.cadence/pegasus/" getShellEnvVar("USER") "presets/pegasusint/snapshot")
globalsnpdir = "./CAD_SNP_DIR"

setShellEnvVar(PegasusInt_SnapshotsDirs" strcat(localsnpdir ":" slobalsnpdir))
```

`PegasusInt_SnapshotsDirs` supports path inputs containing following syntax:

- Relative Path
- Environmental variables

Pegasus Interactive checks path validity and displays the absolute snapshot directory path in the *Pegasus Interactive Snapshot Creator and Pegasus Interactive Manage* field. The order of directory paths in `PegasusInt_SnapshotsDirs` dictates its display order in both of *Pegasus Interactive Snapshot Creator and Pegasus Interactive Manage* field.

Quick Start: Create Snapshot

This section introduces different ways of creating snapshots

- Interactive Creation for the first time via Pegasus Interactive Snapshot Creator
- Create a snapshot based on a pre-existing preset file

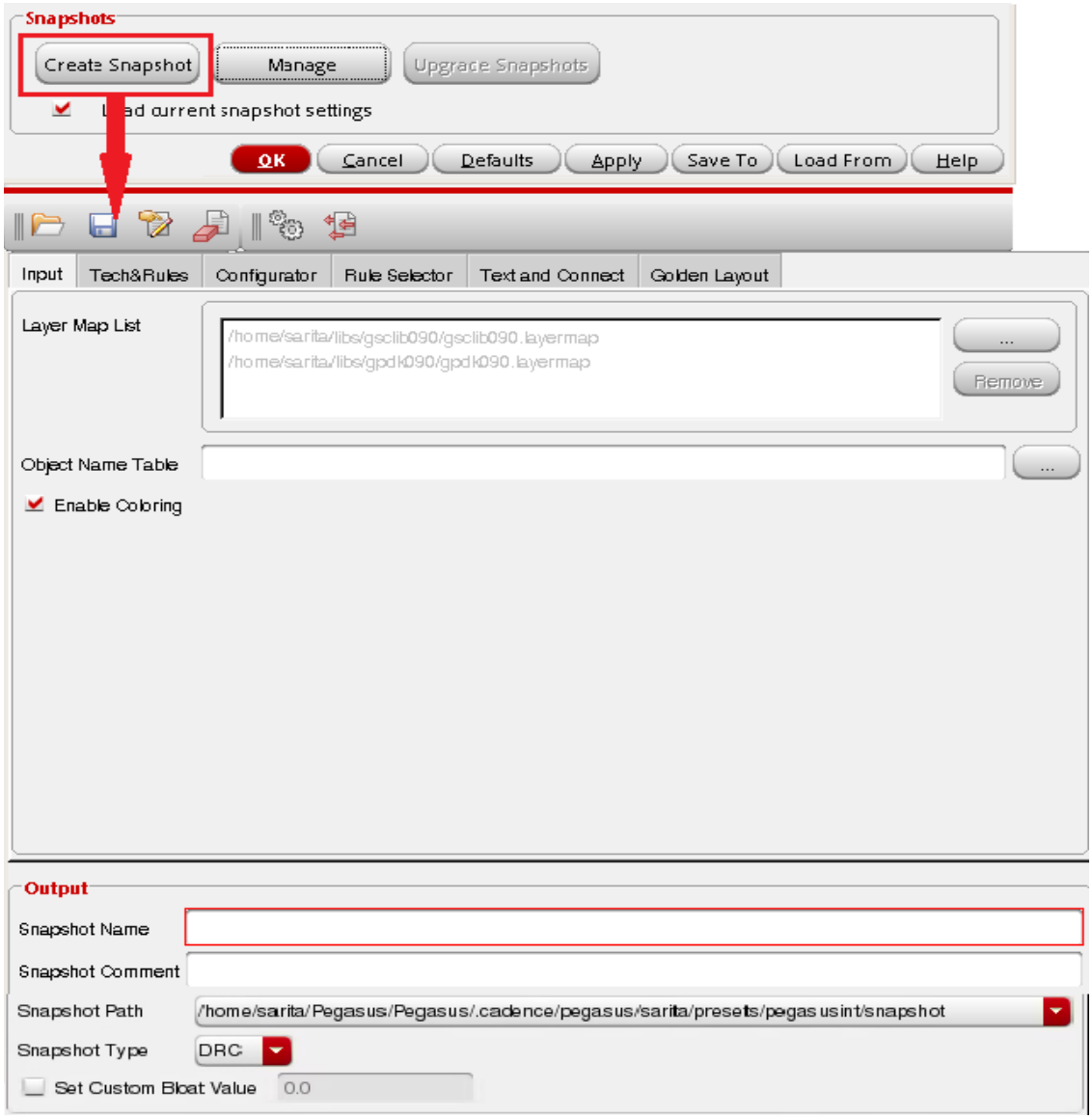
- Create a snapshot from current snapshot setting

Related Topics

- Creating Snapshot for First Time on page 63
- Creating Snapshot From Preset on page 65
- Creating Snapshot From Current Active Snapshot Setting on page 67

Creating Snapshot for First Time

Pegasus Interactive Snapshot Creator allows you to interactively create the snapshot. It can be invoked through the *Create Snapshot* button under the *Pegasus Interactive Run Options*.

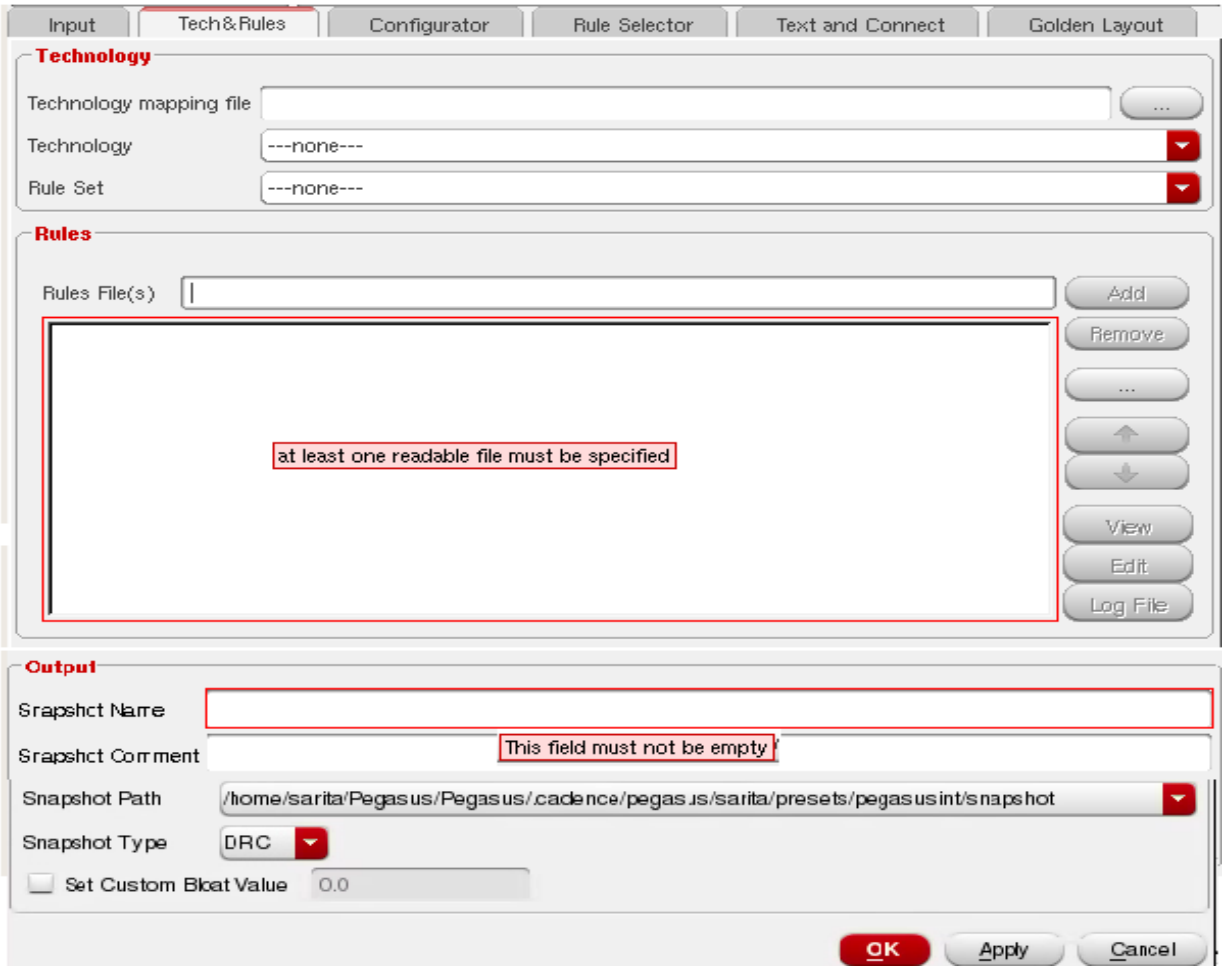


To generate a snapshot using Pegasus Interactive Snapshot Creator, three minimum inputs are required:

- Layermap file
- Rule Deck

- Output Snapshot name

Assuming no technology is set up, Pegasus Interactive flags missing required inputs visually by boxing fields with the red highlight. Once the cursor hovers over the red-box, error message is displayed as a tooltip. Following example illustrates error dictated by the interface. Till required inputs are incomplete, GUI prevents you from creating a snapshot.



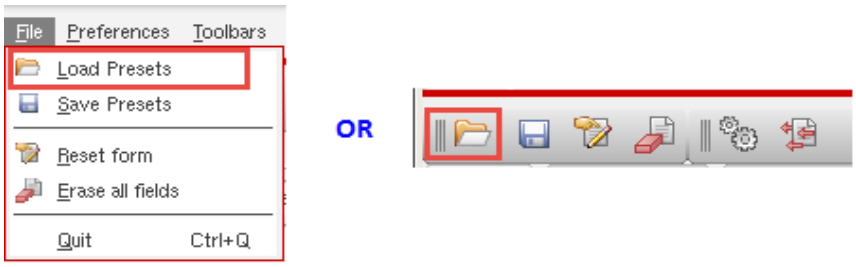
To generate a snapshot, enter the required inputs in Pegasus Interactive Snapshot Creator. Then click *Apply* or *OK* to invoke snapshot creation. During Snapshot Creation, following popup box indicates the progress of snapshot generation:

The newly created snapshot is automatically populated under Pegasus Interactive toolbar's snapshot combo field.

Creating Snapshot From Preset

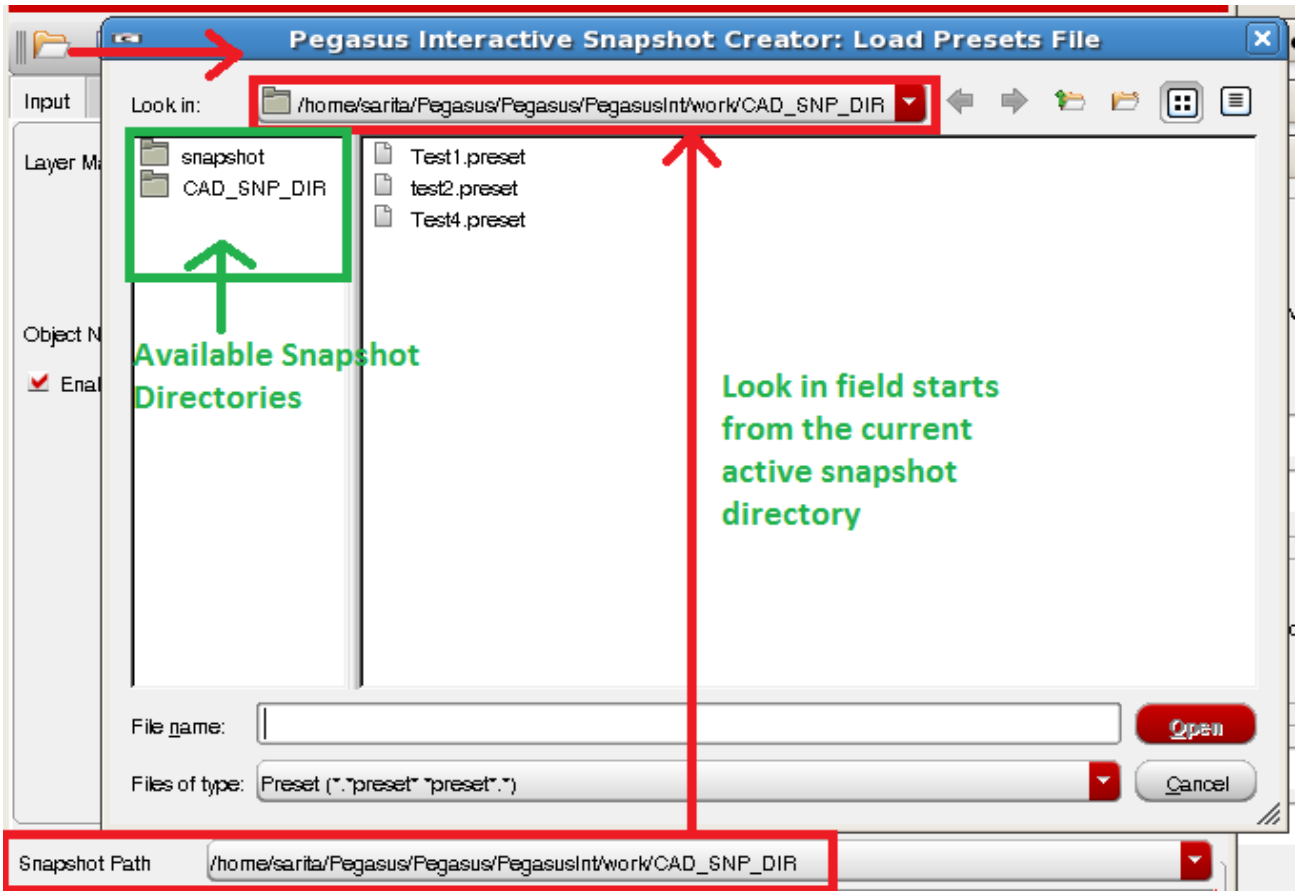
The preset file contains the configuration setup that can be loaded into Pegasus Interactive Snapshot Creator. As long as the syntax in the preset file is supported by Pegasus Interactive, its setup value will be populated in the GUI.

To load the preset, you can interactively click the *Load Presets* icon on the *File* toolbar or *File -> Load Presets*.

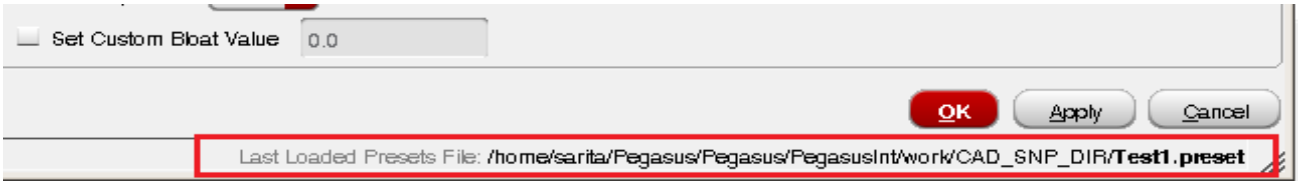


It opens a file searching form. For the current *Look in* directory, files' name with suffix `.preset` are populated in the window. Current *Look in* directory is initially set to current snapshot directory populated under the *Snapshot Path* field. You can navigate to any of available snapshot directory content by either selecting it from the left directory list field or

through file searching utility provided by this GUI. These available snapshot directories are determined by `PegasusInt_SnapshotsDirs`.

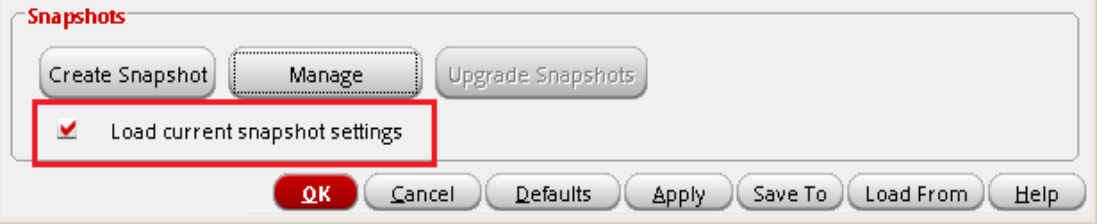


Once the desired preset file is located, click *Open*. It loads the *Pegasus Interactive Snapshot Creator* contents based on the selected preset file. After successfully loading the preset, the loading status is displayed at the bottom of *Pegasus Interactive Snapshot Creator*.



Creating Snapshot From Current Active Snapshot Setting

Similar to creating snapshot from a preset file, Pegasus Interactive allows you to load current active snapshot setting during Snapshot Creator invocation. The advantage of this feature is to allow you to further create a snapshot from a reference snapshot setting.



This option is enabled by default when Pegasus Interactive identifies the current active snapshot.

Pegasus Interactive Snapshot Creator Form

This form is another derivation of Pegasus run submission form. However, its contents are specific to Pegasus Interactive snapshot creation only. You can invoke this GUI within Virtuoso session or standalone through `pegasusgui` invocation utility. You can refer to Appendix A - Preset to Snapshot Batch Migration for standalone GUI invocation.

Similar to Pegasus submission form, Pegasus Interactive Snapshot Creator GUI has a built-in intelligence that checks on input validity before generating snapshots. If any required input is missing, fields in question are enclosed with red box highlights. The error message is displayed as a tooltip when the cursor is hovered over each of the red boxed field.

This GUI fields are grouped as following:

- Snapshot content customization
- Snapshot Output

Related Topics

- Customizing Snapshot Content on page 68
- Customizing Snapshot Output Field on page 87

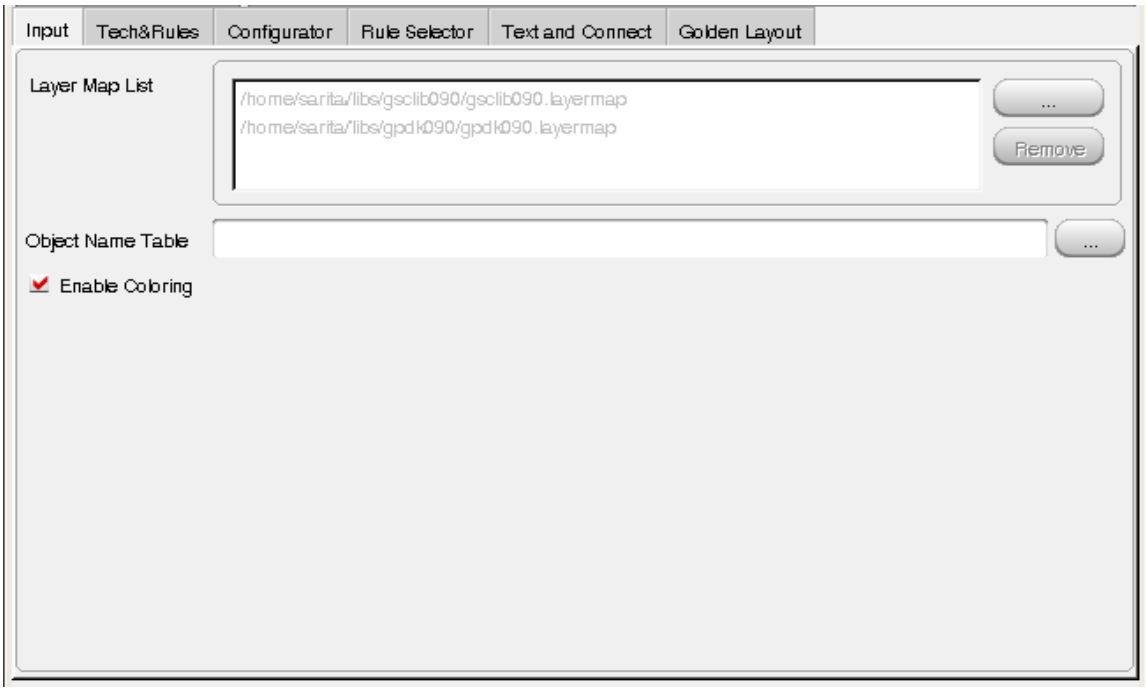
Customizing Snapshot Content

Right above the *Output* field, six tabs are available for customization. These options allow you to specify DRC criteria that will be stored in the snapshot.

- *Input*: Defines polygon handling information used between Virtuoso and Pegasus.
- *Tech & Rules*: Defines technology design rules deck.
- *Configurator*: Part of technology setup. It is used to override switch based default setting defined in the rule deck.
- *Rule Selector*: Interactive rule selection.
- *Text and Connect*: Defines text based connectivity criteria for Pegasus Interactive to follow.
- *Golden Layout*: Defines Golden layout pattern files for graphical DRC.

Input

The *Input* tab contains information used for Pegasus Interactive to do correct polygon translation and packaging.



The layer handling between Virtuoso and Pegasus is different. Pegasus Interactive is built as an active bi-directional communication channel that transports polygons between two platforms. Thus, it handles direct layer mapping between Virtuoso and Pegasus. To ensure the accuracy of layer translation, mapping file is mandatory for Pegasus Interactive while is optional for signoff Pegasus DRC.

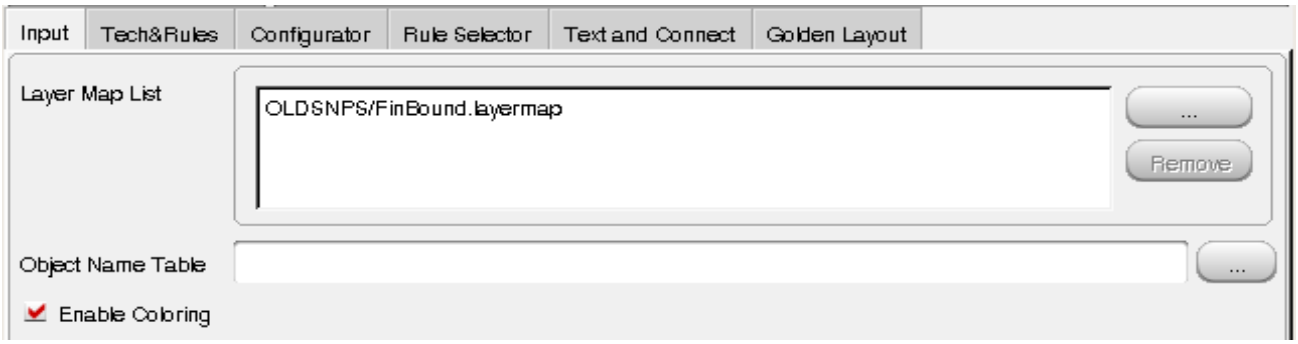
There are two types of mapping files: layer and object. The former contains mapping between Layer-Purpose pair and Layer-Datatype stream pair while latter maps Virtuoso specific object attributes to Layer-Datatype stream pair. Object mapping file is optional and layer mapping file is given by the same source distributes rule deck.

You can specify the layermap file by clicking the (...) button and browse to the location to select it. To remove a layer map file, select the file and click *Remove*.

Similarly, you can specify the object mapping file by either of the following ways:

- Entering file path in the *Object Name Table* text box

- Clicking the (...) button and browse to the location to select it.

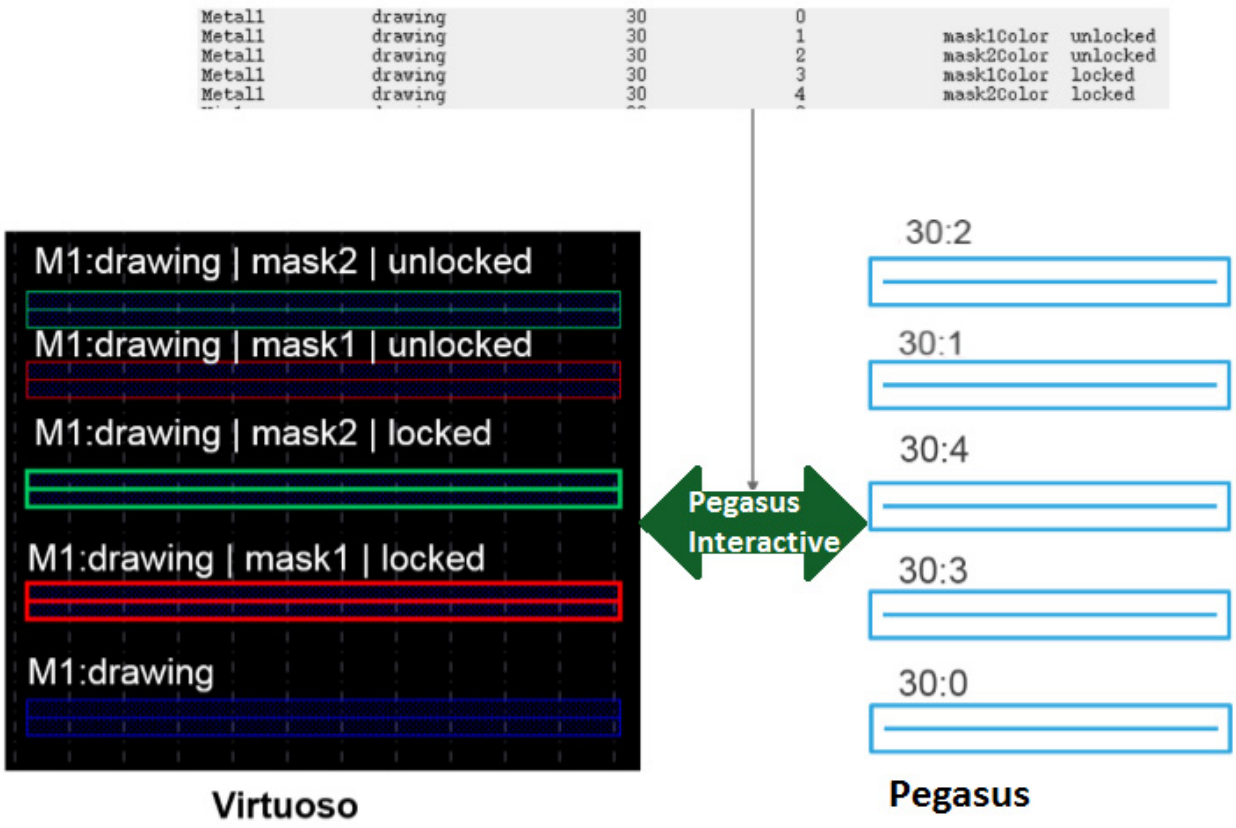


Since layermap file is one of required inputs, you can specify its initial value in the *Layer Map List* text box through the `PegasusInt_DefaultSNPLayerMap` shell variable before launching Virtuoso. If design library is attached or is referencing the technology library, then the *Layer Map List* field is automatically populated within layermap file being found in technology library.

Depending on the version of Virtuoso release, *Enable Coloring* and *Map All Colored To Locked* options are available to specify color mapping criteria.

Starting from ICADV12.1, for a given Layer-Purpose pair, a set of polygons can have additional color attributes assigned on them: color and color state in the layout. By selecting *Enable Coloring* check box, Pegasus Interactive understands color implications set on these polygons. It first translates them to equivalent Layer-Datatype stream pair based on layer mapping definition and then transports them to Pegasus prior to DRC. Similarly data preparation is done after DRC during reporting violations transport from Pegasus to Virtuoso.

Following example shows how Pegasus Interactive refers layer mapping definition to translate data between Virtuoso and Pegasus.



Map All Colored To Locked is a sub-option of *Enable Coloring* and its enabled once you select the *Enable Coloring* check box. It is only available in ICADV122 or later. When this option is selected, Pegasus Interactive treats all colored polygons in the layout as locked and maps them to equivalent Layer-Datatype stream pair accordingly. By default, *Map All Colored To Locked* field is hidden. Use environment variable: `envSetVal("xstream" "xstHideMapAllColorToLocked" 'boolean nil)` or shell environment variable `PegasusInt_HideMapAllColorToLocked N` to make it visible on the form.

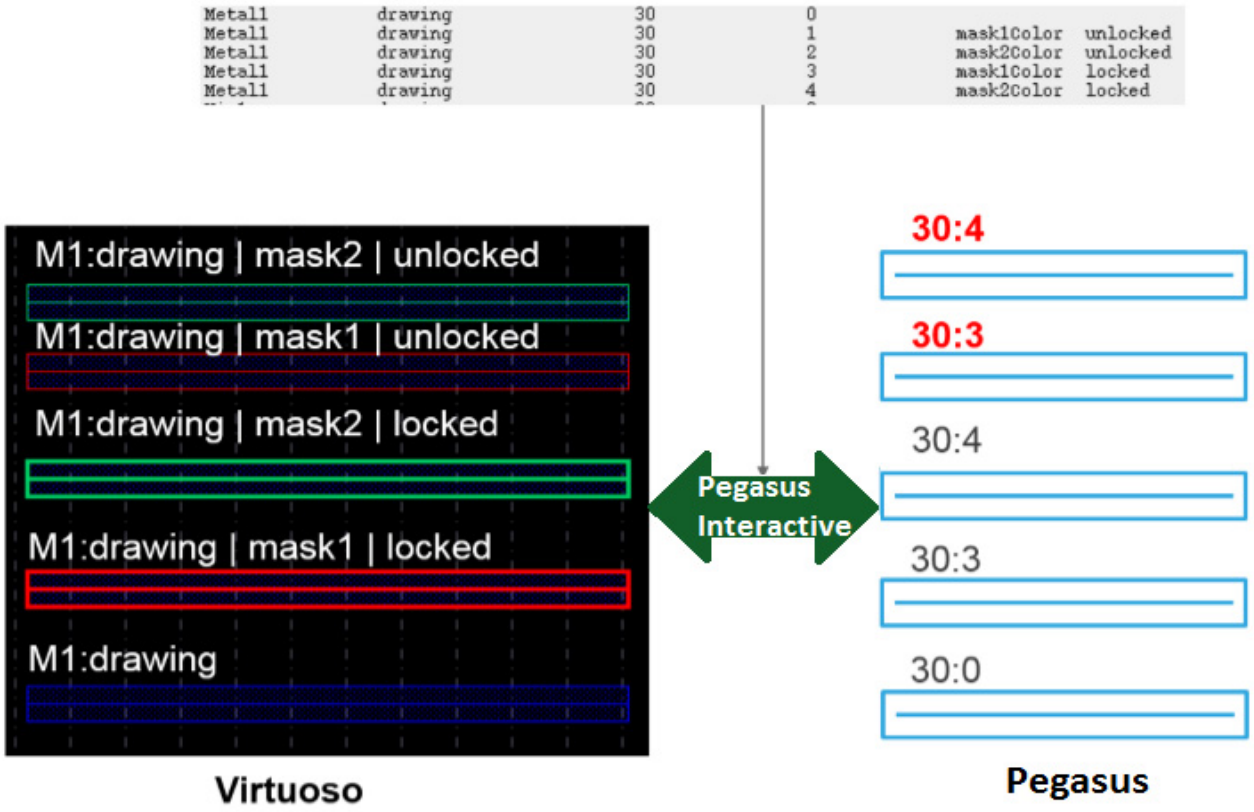
The priority of controls is the following:

1. `envGetVal("xstream" "xstHideMapAllColorToLocked")` if defined.

2. If not defined, shell envvar `"PegasusInt_HideMapAllColorToLocked"`.

3. If both are not defined - `default:hide`.

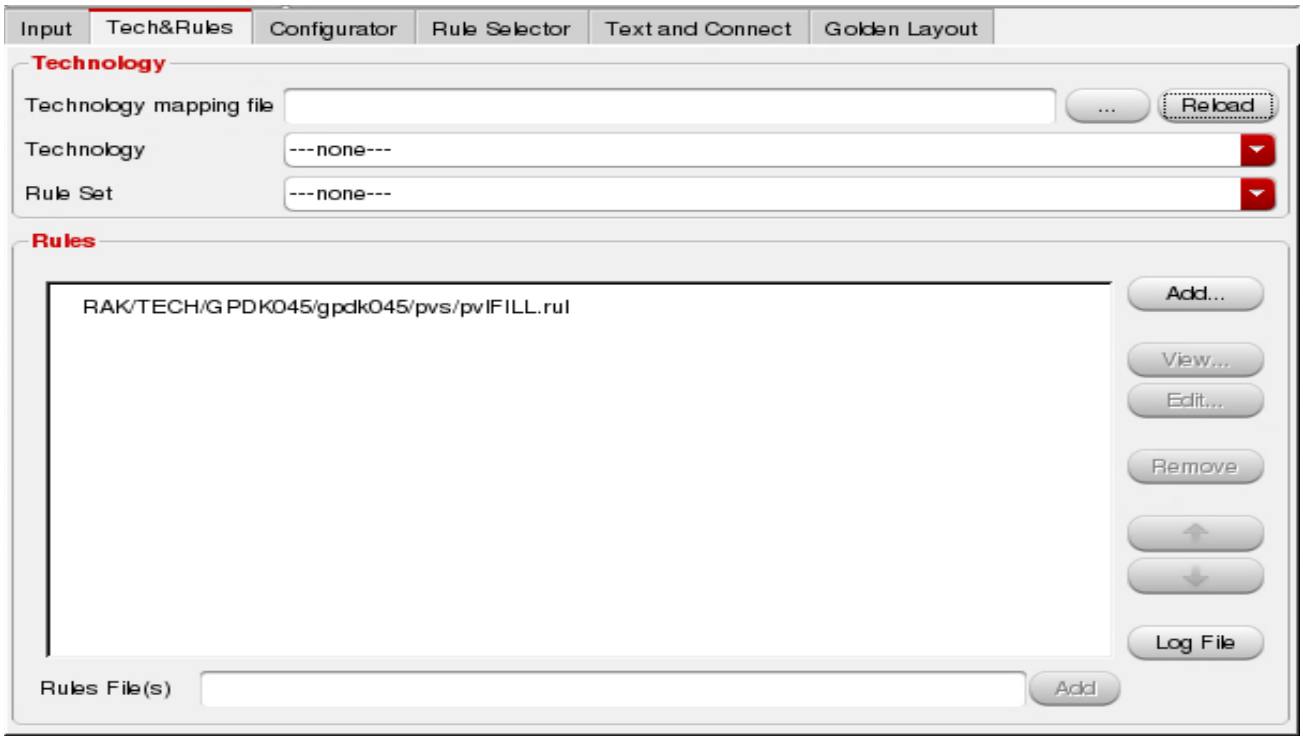
Following example shows Pegasus Interactive data preparation when *Map All Colored To Locked* is selected.

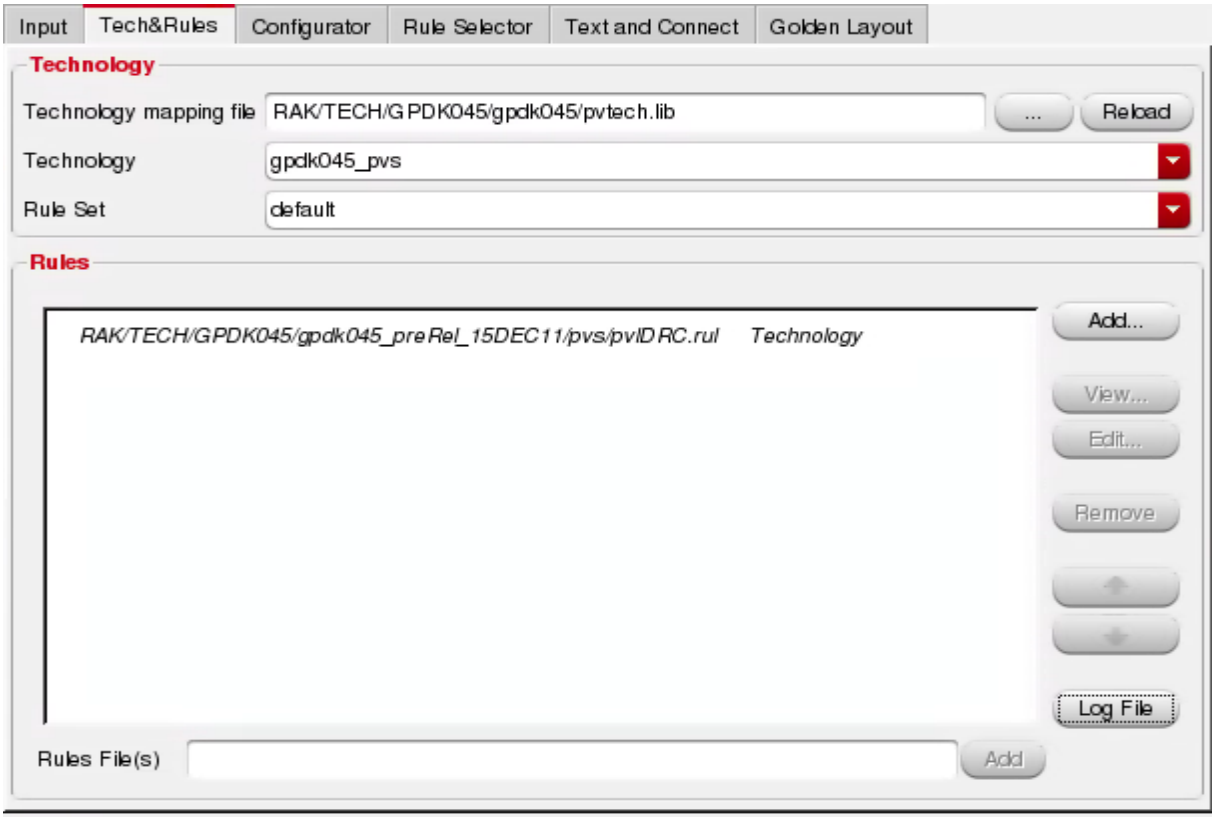


Tech&Rules

This tab defines the source that snapshot is generated from. You can either add the rule deck path under the *Rules* field or refer to the rule set available under the *Technology* field. In

general, these settings under this tab should be controlled by CAD. Following examples show two ways of setting up the technology specific DRC deck information:





Technology

- *Technology Mapping File* - Enter the name of the technology mapping file containing the technology library names and locations. Alternatively, click the (…) button and browse to the location where the files are stored and select the file.
- Click the *Reload* button to reload the technology file and refresh technology and rule set details accordingly. This is required when the technology file specified earlier has changed.
- *Technology* - Select the technology you want to use in the run. The list shows the technologies that are defined in the Technology Mapping File.
- *Rule Set* - Select the rule set from the list of rule sets defined for the technology specified in the technology field. The rules are defined in the file techRuleSets in a technology directory.

  If the rule set contains one or more applicable preset files, then following label is displayed on the form:

  Contains GuiPreset file[s]

If the rule set contains one or more applicable configurator files, then following label is displayed on the form:

Contains GuiConfig file[s]. Check the Configurator tab

If the rule set contains both preset and config files, then following label is displayed on the form:

Contains GuiPreset file[s] and GuiConfig file[s]. Check the Configurator tab

The selected rule is displayed in the *Rules* section with keyword *Technology*. You can view the contents of this rule set by clicking the *View* button. However, you cannot edit the contents of the file. If you use additional rule files, then technology rules are listed at the bottom of the rule files list. This order cannot be changed.

Similarly, you can add more rule-deck files to your run.

- To remove a rule file, select the rule file and click the *Remove* button.
- To view a rule file, click the *View* button.

  You can use the *Find* option to find a required keyword or sentence in the rule file. Enter the keyword to be searched in the text box and the keyword is highlighted in the report.

  Click *Next* and *Previous* to navigate in the form to see occurances of the keyword searched.

  Select the *Highlight* check box to highlight the search result. To customize the search, you can select the *Matchcase* check box*, Whole word* check box or both.

  Click *Help* to open the relevant help topic for this form in Cadence help and click *Close* to close the form.
- To edit contents of a rule file, click the *Edit* button. It will open a runtime editor to modify the rule file.
- To view the contents of log file, click the *Log File* button. It opens the Log file showing details of the run after checking the rule files added to the *Rules* block. Use this button to review log file when you see a red border around the rule, which means that rule deck has errors.

Since DRC deck is one of required inputs, you can specify its initial value in *Rule File(s)* field through the `PegasusInt_DefaultSNPRuleDeck` shell variable prior to Virtuoso launch. For details of setting rule set using Technology, see *Appendix C - Handling Technology Data* in Pegasus in the Pegasus User Guide.

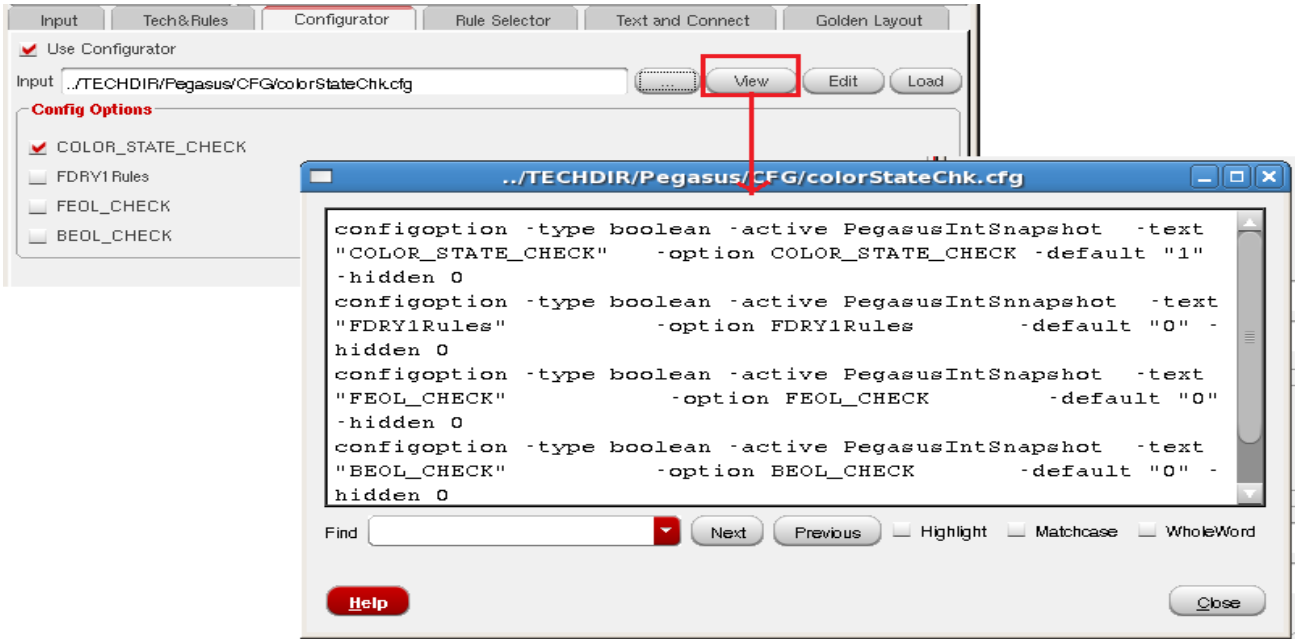
Configurator

For a given rule deck, it may contain a set of switches controlling the rule selection. In the configurator, you can change switch's value to trigger rules selection that is different from the default content in the rule deck. Due to the amount of switches defined in the rule deck, manual changing rule deck content can be difficult to manage and track the changes. Thus, configurator flow allows you to select or override setting without manually editing rule file content.

By default, configurator is disabled. Select *Use Configurator* check box to enable it. Once it is enabled, it asks for a configuration file to be loaded.

After loading the configuration file, the graphical representation of options are populated under the *Config Options* field. Following example shows any of four switches defined through `#define` in the DRC deck can be overridden interactively without physically modifying the DRC deck itself.

At the same time, both of View and Edit buttons are activated. These buttons allow you to view configurator file content and edit its content on the fly. Following figure shows how to view configuration file content interactively.



For detail of configuration file syntax, refer to the *Pegasus Configurator* chapter in the *Pegasus User Guide*.

Generally, Configuration file is maintained globally by CAD. CAD can declare its initial value in the *Input* field through the `PegasusInt_DefaultSNPConfigFile` shell variable before launching Virtuoso.

Rule Selector

The initial content of *Rule Selector* reflects current rule selection found in the loaded rule deck. Essentially, it is the result of rule deck compilation done through Pegasus. The *Rule Selector* section enables you to individually select or deselect rule sets or rule groups.

In Rule Selector tab, two view modes are available: *Groups* and *Plain*.

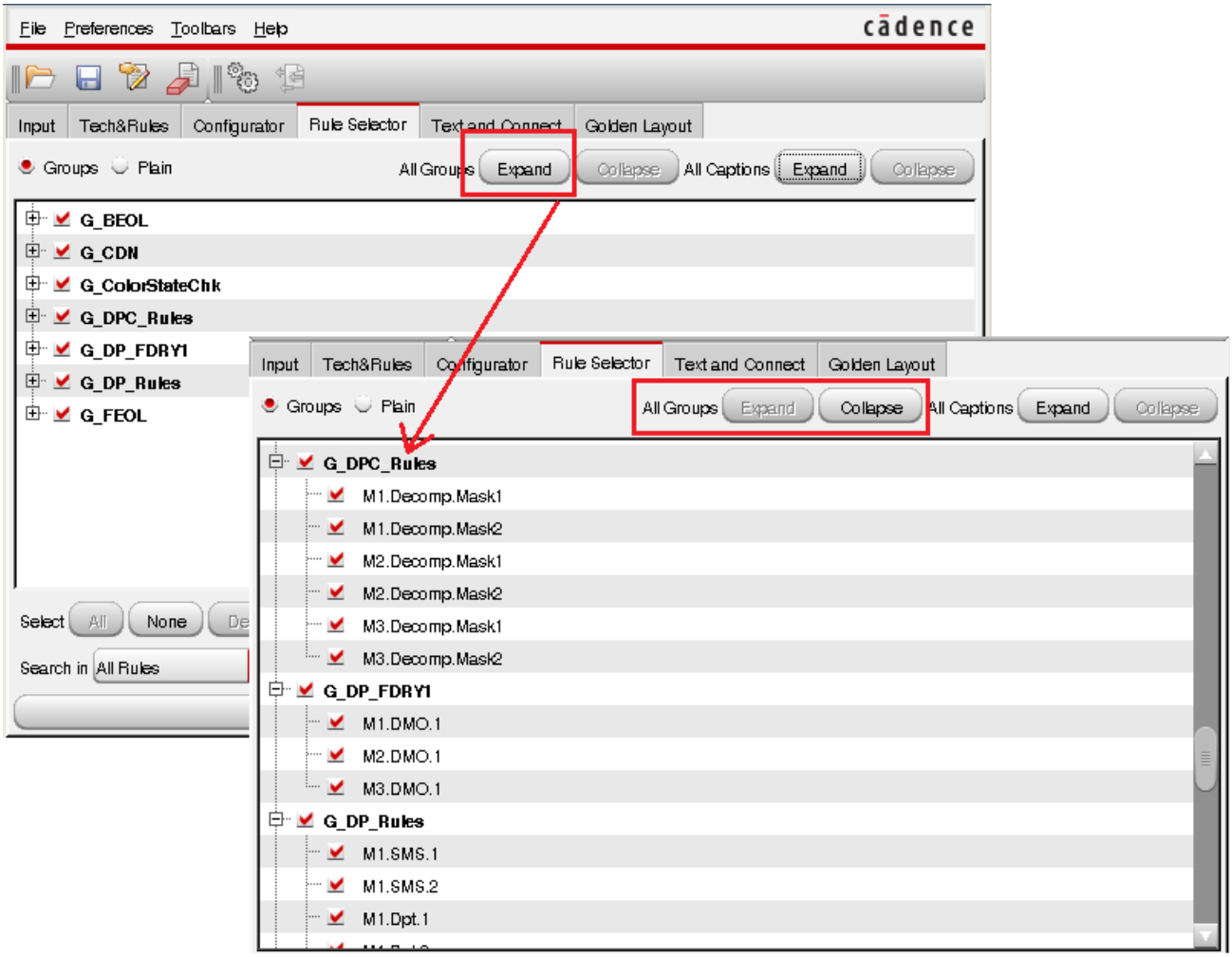
The default view mode is *Groups*. In this default view mode, rules are displayed either by the group they belong to or by *Ungrouped Rules*. Following example shows how groups'

representation in *Rule Selector* is declared in the rule deck. All `G_*` rule groups are defined through the `group_rules` command in the rule deck.

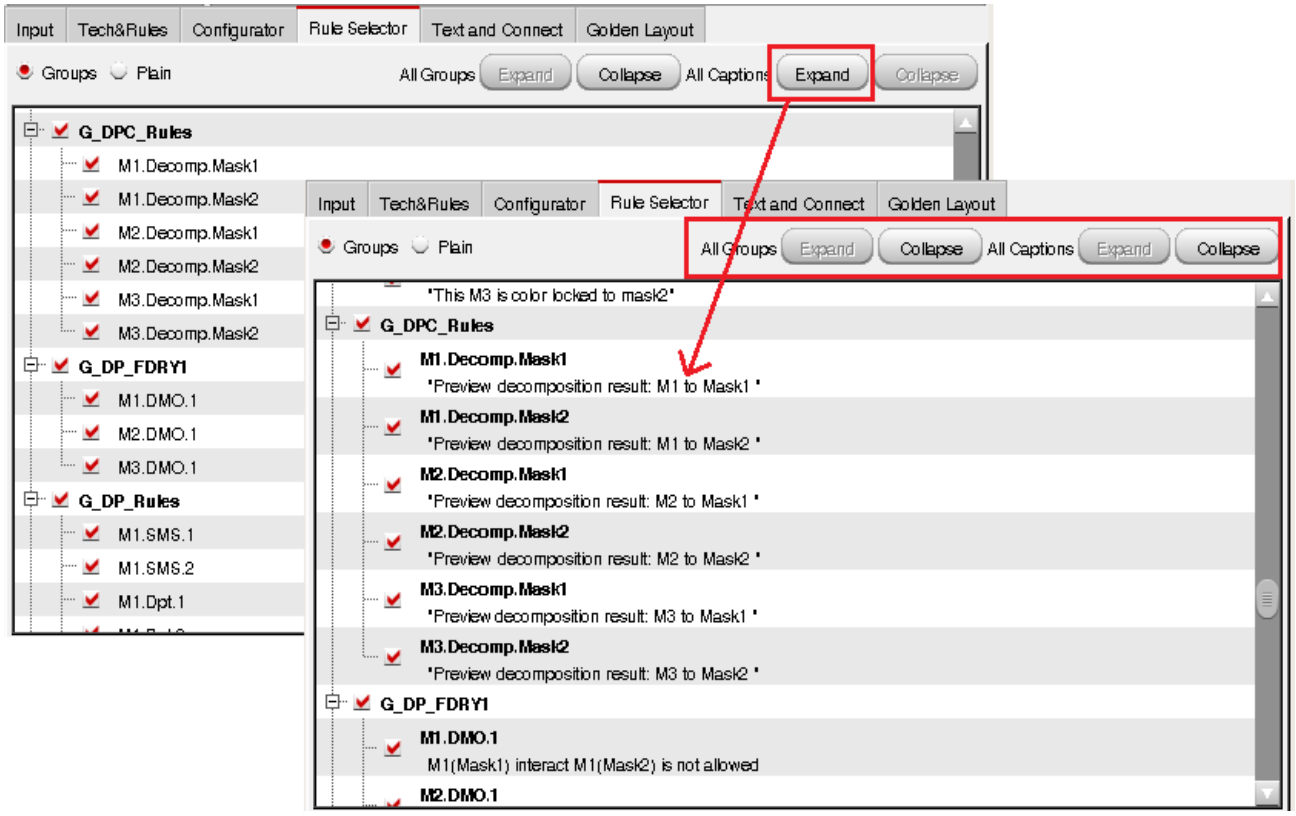
Within Groups, rule information is constructed based on its rule name and description. Rule name is defined through the `rule` command while rule description is defined through the `caption` command. You can expand or hide all of rule names and descriptions using:

- All Groups: Expand | Collapse
- All Captions: Expand | Collapse

Following example shows what happens when you click the *All Groups: Expand* button. All the rules that belong to a group are displayed.

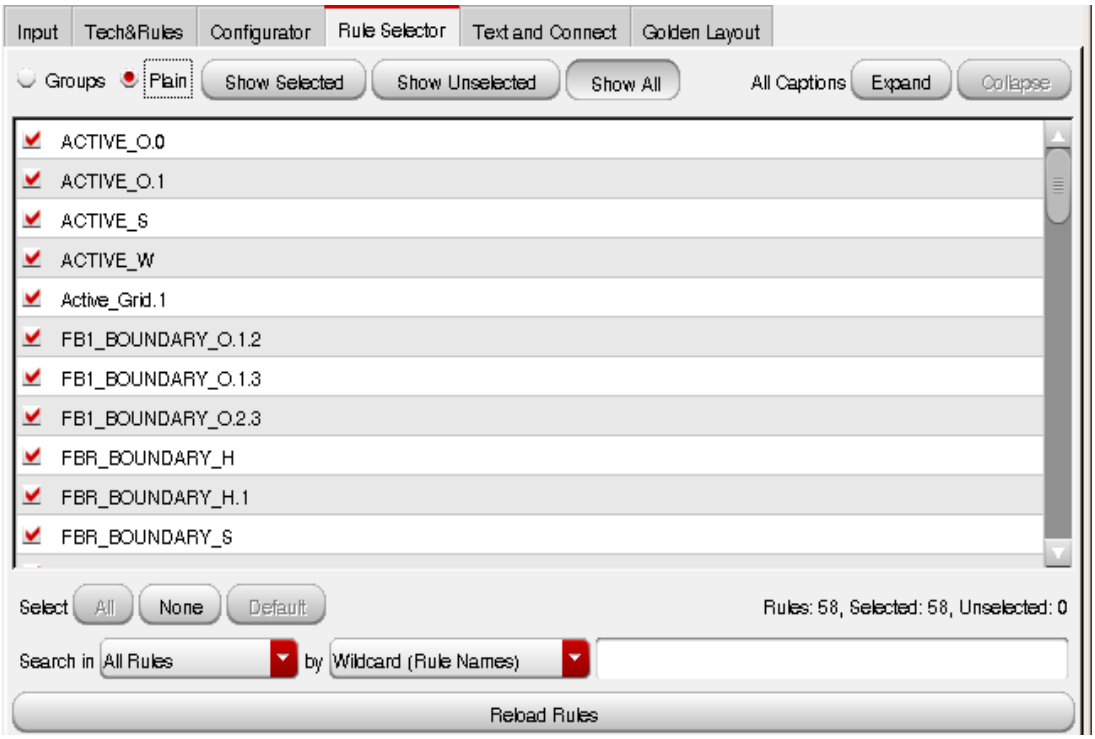


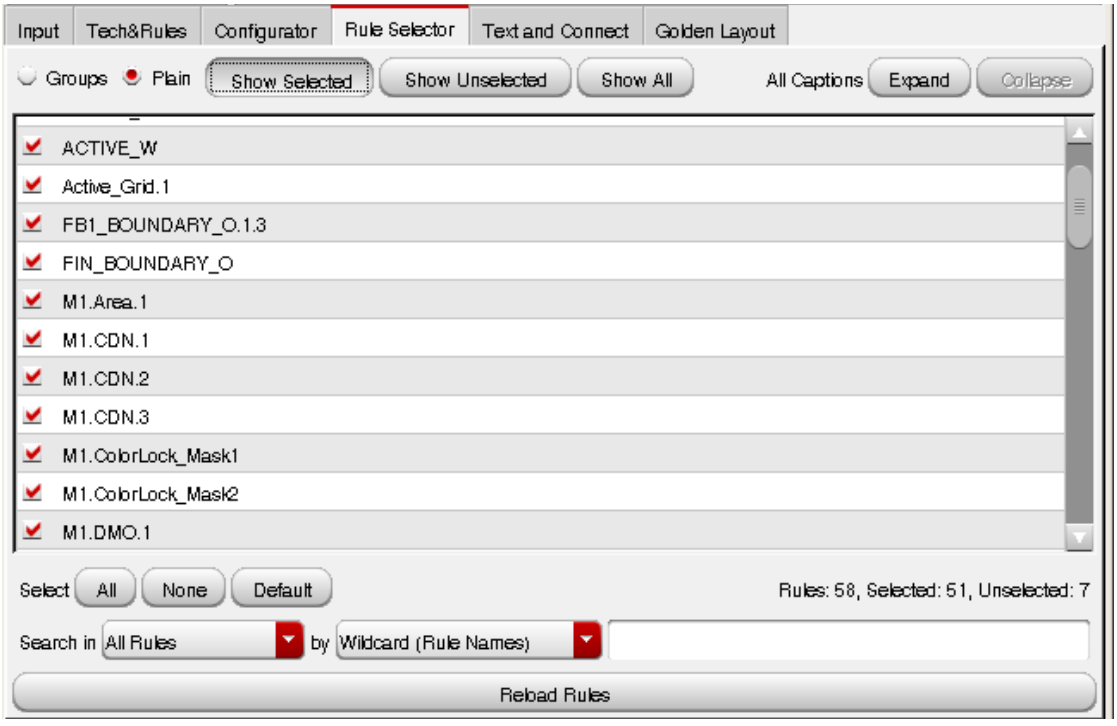
To see the description of each rule, clicking the *All Captions: Expand* button shows each rule's description as its child hierarchy. Following example shows the rule's content after clicking the *All Captions: Expand* button.



The *Plain* view mode allows you to see rules based on their current selection status. It is a flat display by rule name. Once the *Plain* mode is selected, additional three buttons are available for you to set rule display based on its current selection status.

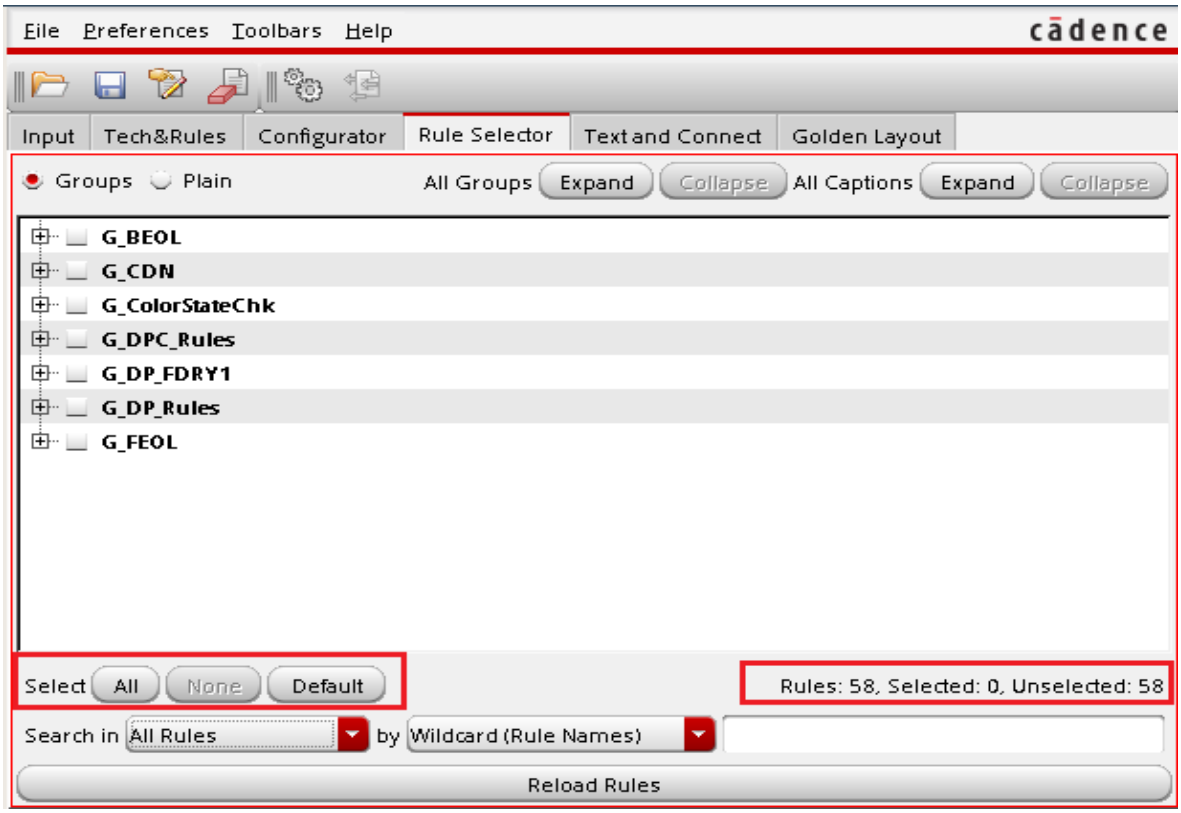
By default all rules are displayed. You can further filter unselected rules by clicking *Show Selected* to customize the rule display. Rule by default are displayed by rule names. However, you can expand or hide its rule caption using *All Captions: Expand/Collapse*.





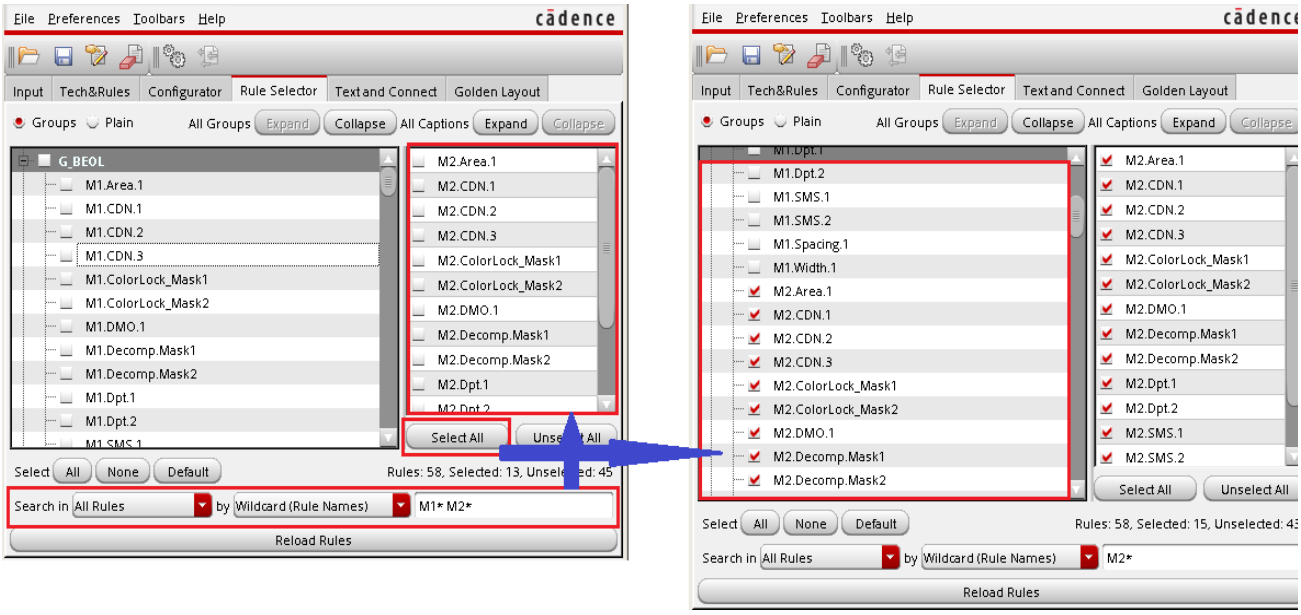
When the *Rule Selector* tab is activated to perform rule selections, all the selection options are enabled. For batch rule selection, you can select or deselect all the rules through *All* and *None* buttons. You can also reset the rule selection back to default through *Default* button.

Following example shows results after clicking the *None* button. Note that rule selection summary indicates 0 rule selection.

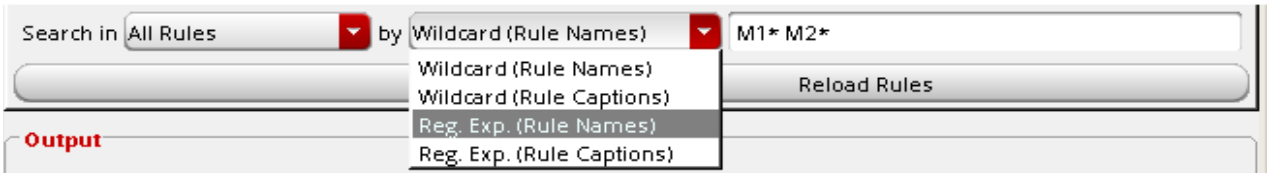


For search based rule selection, you can first set the search filter and then further define searching criteria by declaring searching condition in the entry field. Once matched rule names or captions are found, they can be pre-viewed. You can then perform rule selection by

clicking the *Select* or *Unselect* buttons. The preview column is only displayed where there is searching condition declared in the entry field.



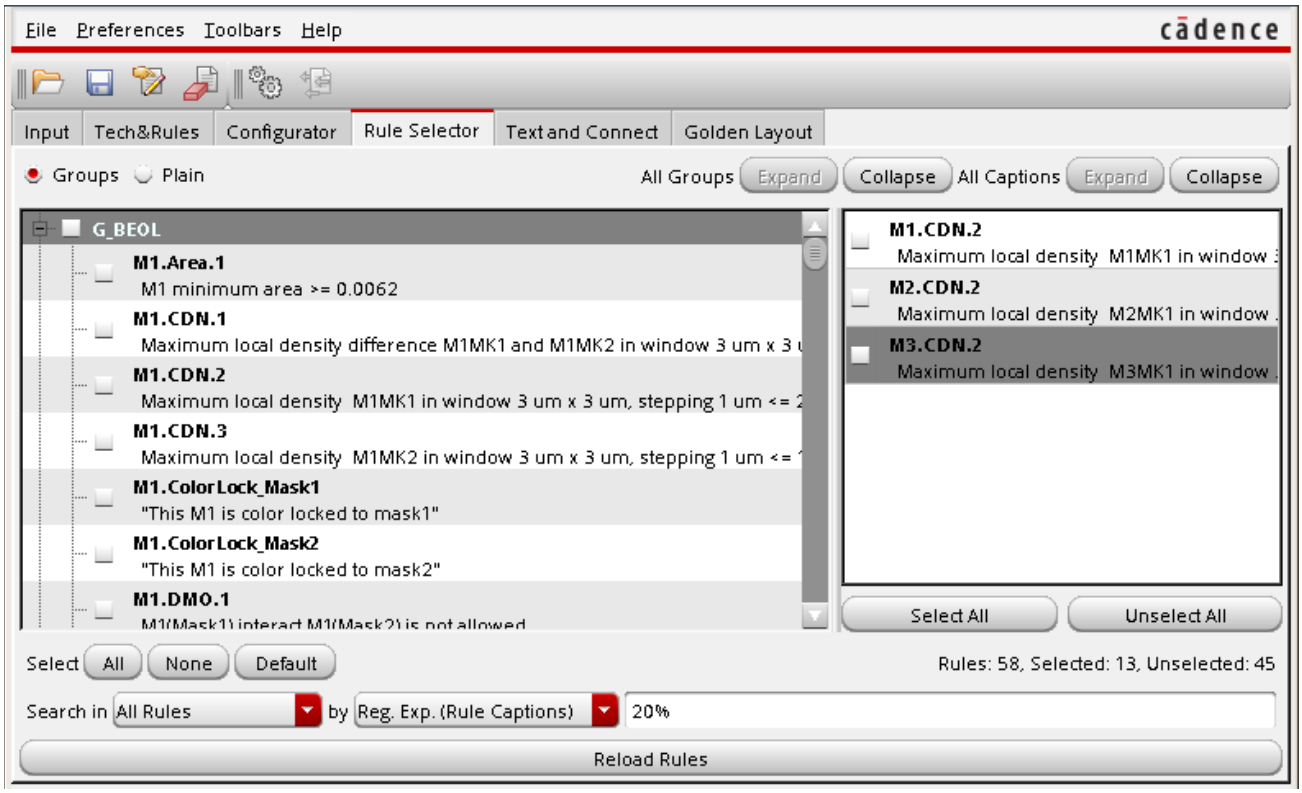
Pre-defined search filter allows you to specify where and how the search should be conducted. You can customize your search using any of the search criteria listed in the *Search in* drop-down list. The options are: *All Rules*, *Connectivity Rules* and *Density Rules*. The *by* field is related to the *Search in* field and the drop-down list provides second level of filtering. You can narrow down the search results using options listed in the drop-down list. The options are: *Wildcard (Rule Names)*, *Wildcard (Rule Captions)*, *Reg. Exp. (Rule Names)*and *Reg. Exp. (Rule Captions).*

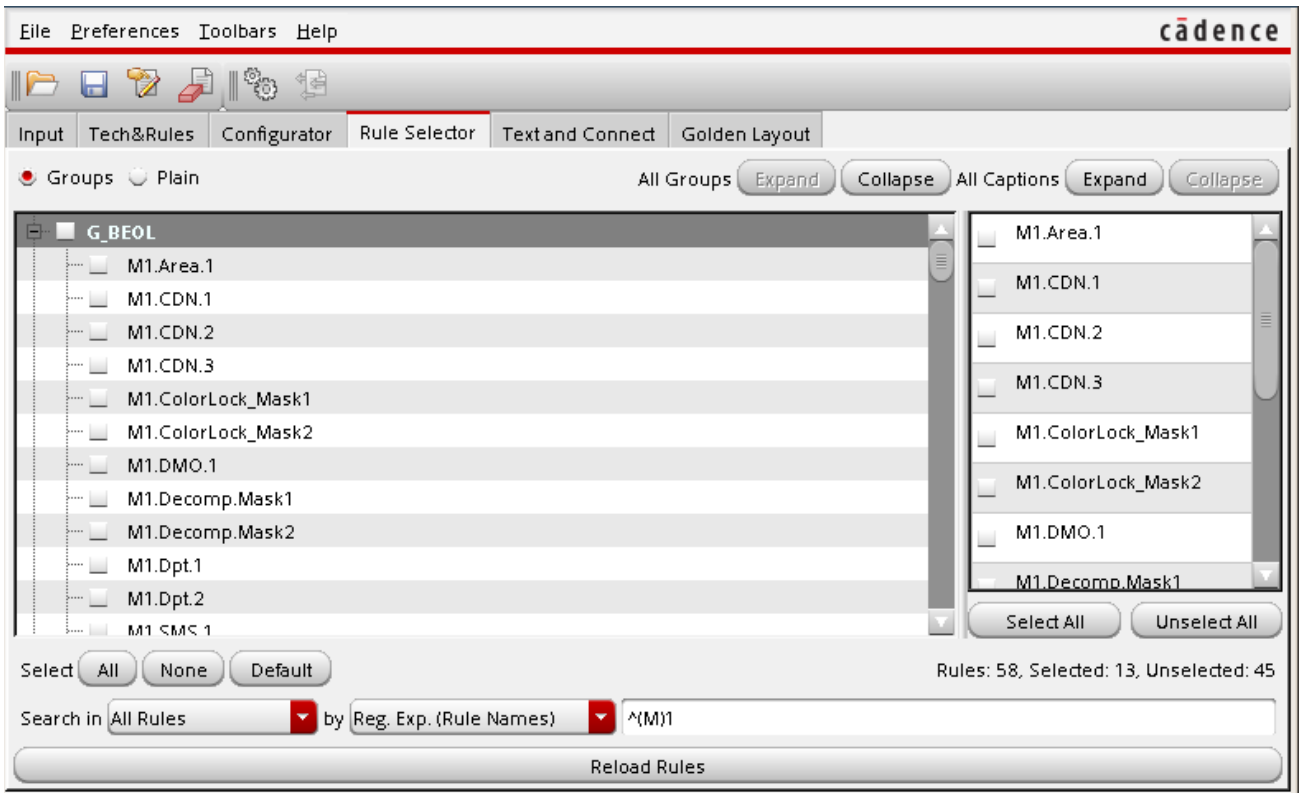


Reg. Exp. (Rule Name) and *Reg. Exp. (Rule Captions)* search filters support regular expression syntax.

You can specify multiple rule names separated by space as shown below:

Following example shows finding rule description containing exact match of `20%` through *Reg. Exp. (Rule Captions)* search filter.

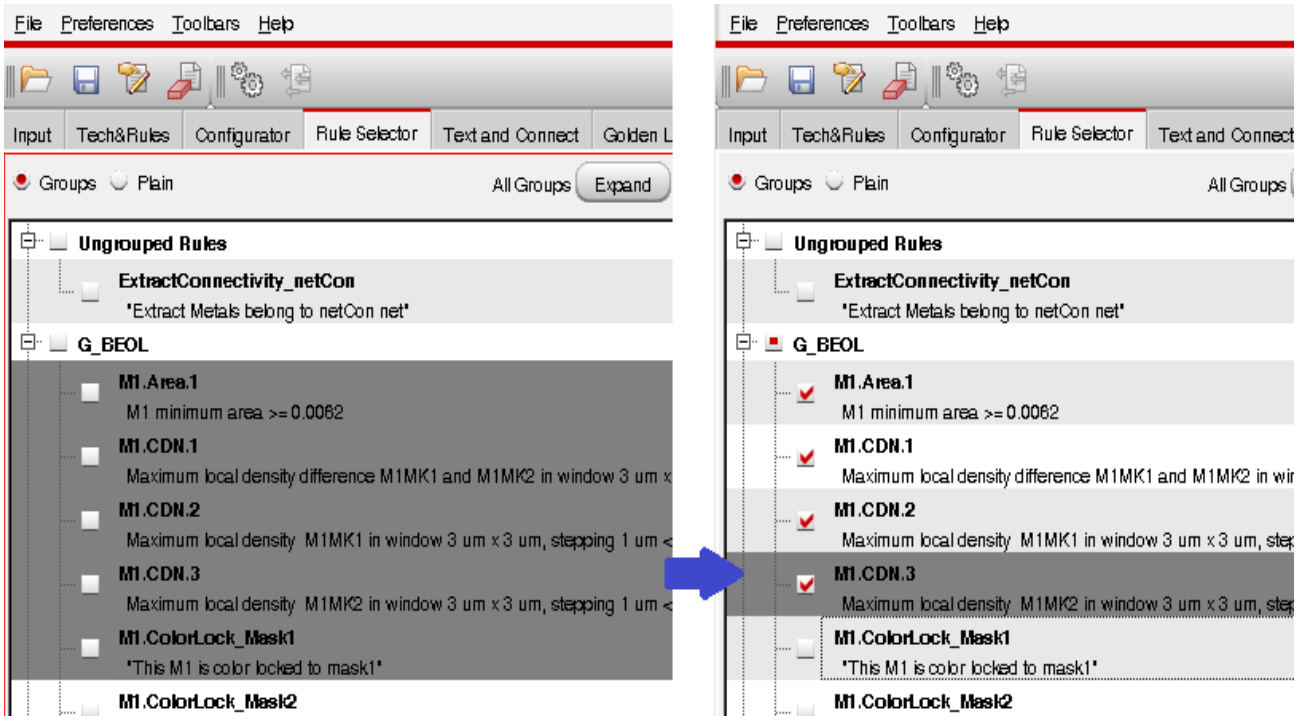




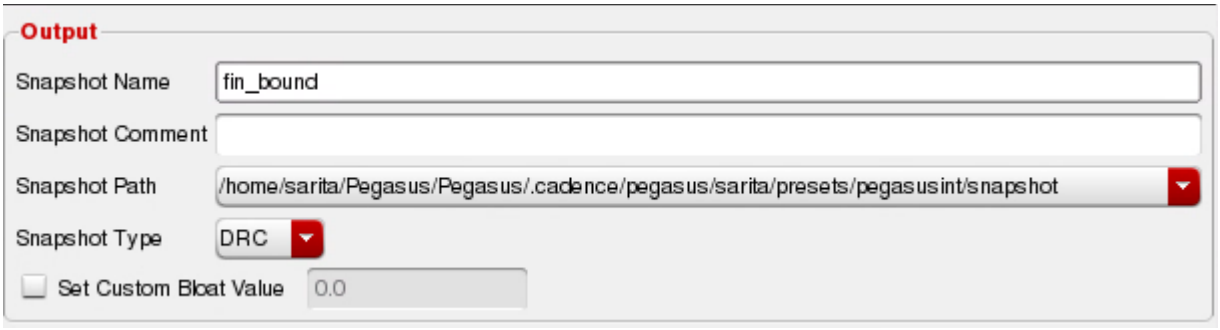
Search options *Connectivity* and *Density* show all rule names based on the connectivity and density respectively.

The *Reload Rules* button re-compiles rules contents from the rule deck. After the compilation, rules contents are refreshed based on compilation results. This is different from the *Default* button in *Select*. *Reload Rules* deletes all the rule compilation results initially obtained by the tool and re-generates rules based on the current active rule deck while *Select: Default* only resets rule selection status based on the original compilation results.

Finally, you can interactively select rules through the *rule display* field by clicking the check box located in the front of rule names or group names.



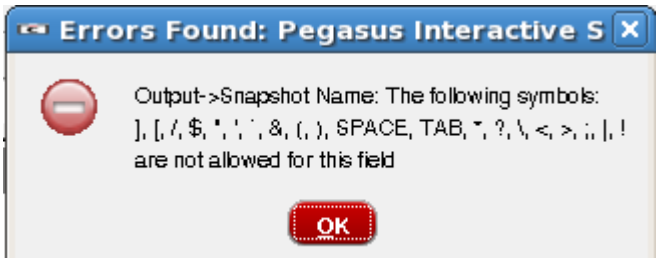
Customizing Snapshot Output Field



The *Output* field contains final snapshot name declaration, snapshot comment, the snapshot directory path, snapshot Type and check box for defining custom bloat value.

In the *Snapshot Name* field, you can enter the snapshot name. The naming syntax should not include any special characters, SPACE or TAB. If any special character is detected in the

snapshot name input, following pop-up message box appears and requests you to modify the input.

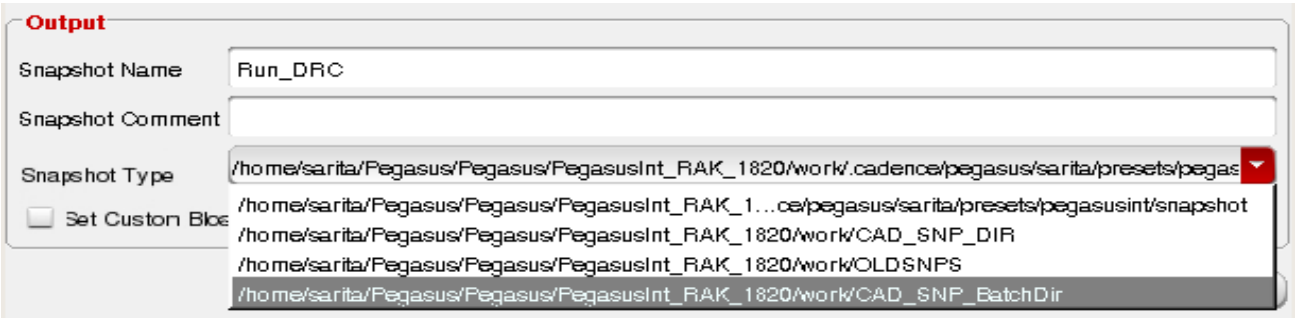


If the snapshot name you have defined already exists, then a warning message that snapshot already exits is displayed.

The *Snapshot Comment* field specifies a description comment for the snapshot.

The *Snapshot Path* drop-down list allows you to set the directory path that snapshot is going to save to. The *Snapshot Path* field is set to the local snapshot directory by default. However, its drop-down items can be replaced through `PegasusInt_SnapshotsDirs`. If `PegasusInt_SnapshotsDirs` is specified, paths in this shell variable is parsed and stored as a list for the *Snapshot Path* combo field. The order of the snapshot directory path in this list is based on the order of declaration in `PegasusInt_SnapshotsDirs`. Thus, the active snapshot directory path is the first item in the list. Also, path declared in `PegasusInt_SnapshotsDirs` is expanded and displayed in its absolute path format.

Following example shows all four paths declared in `PegasusInt_SnapshotsDirs` and how they are displayed under the *Snapshot Path* drop-down list.



The *Snapshot Type* drop-down list allows you to select the snapshot type you want to create: DRC or Fill.

The *Set Custom Bloat Value* check box allows you to specify the custom bloat value. By default the option is disabled. In this case bloat value is defined by the Pegasus engine based on selected checks and this information is outputted as part of snapshot header after the

snapshot is created. When you select the check box and specify the custom bloat value, this value is stored as part of the snapshot header.

You can enter any positive float value including “0”. However, If you enter “0” bloat value, it would mean that you want to check area “as-it-is”. When this option is off, bloat value is determined by Pegasus engine based on selected checks and output this information as part of the snapshot header. Otherwise, custom bloat value declared through this option is stored as part of snapshot header.

Manage Snapshots

Pegasus Interactive allows you to create unlimited snapshots and store them in various snapshot directories. However, accessing snapshots can be cumbersome when multiple snapshots created using different sources are available. To effectively manage snapshots, following utilities are available:

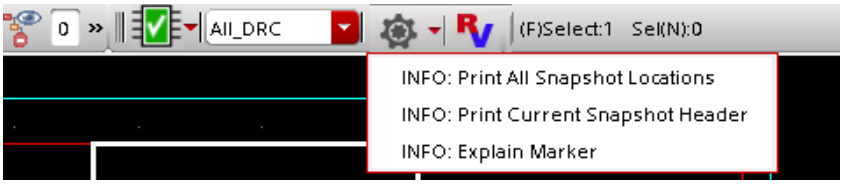
- Snapshot information inquiry
- Snapshot display management

Related Topics

- Viewing Snapshots Information on page 89
- Managing Snapshots on page 93
- Upgrading Snapshots on page 96

Viewing Snapshots Information

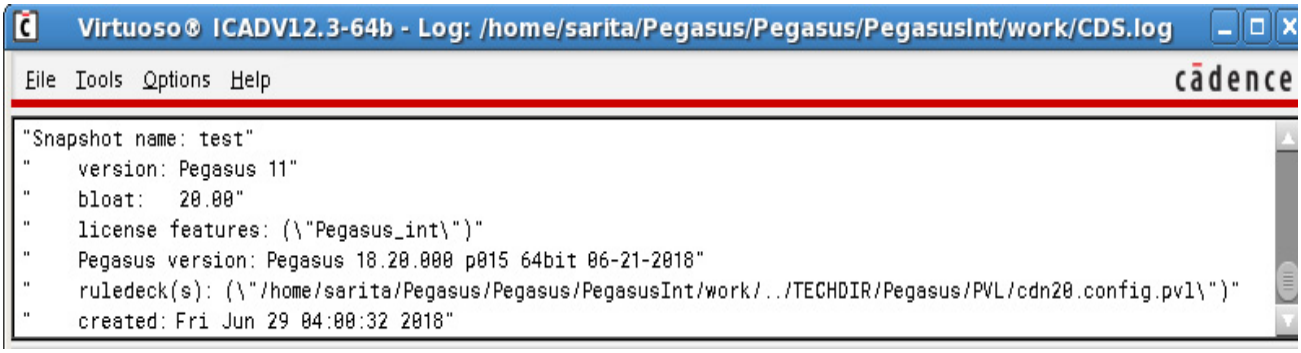
Snapshot content is only readable by Pegasus. Therefore, this feature not only allows you to query the overview of the active snapshot, but also generates a comprehensive snapshot storage location overview.



This inquiry utility contains three drop-down items on the Pegasus Interactive *Run Option* icon:

- INFO: Print All Snapshot Locations
- INFO: Print Current Snapshot Header
- INFO: Explain Marker

To view the header of the current active snapshot, info can be printed out in CIW after clicking *INFO: Print Current Snapshot Header*.



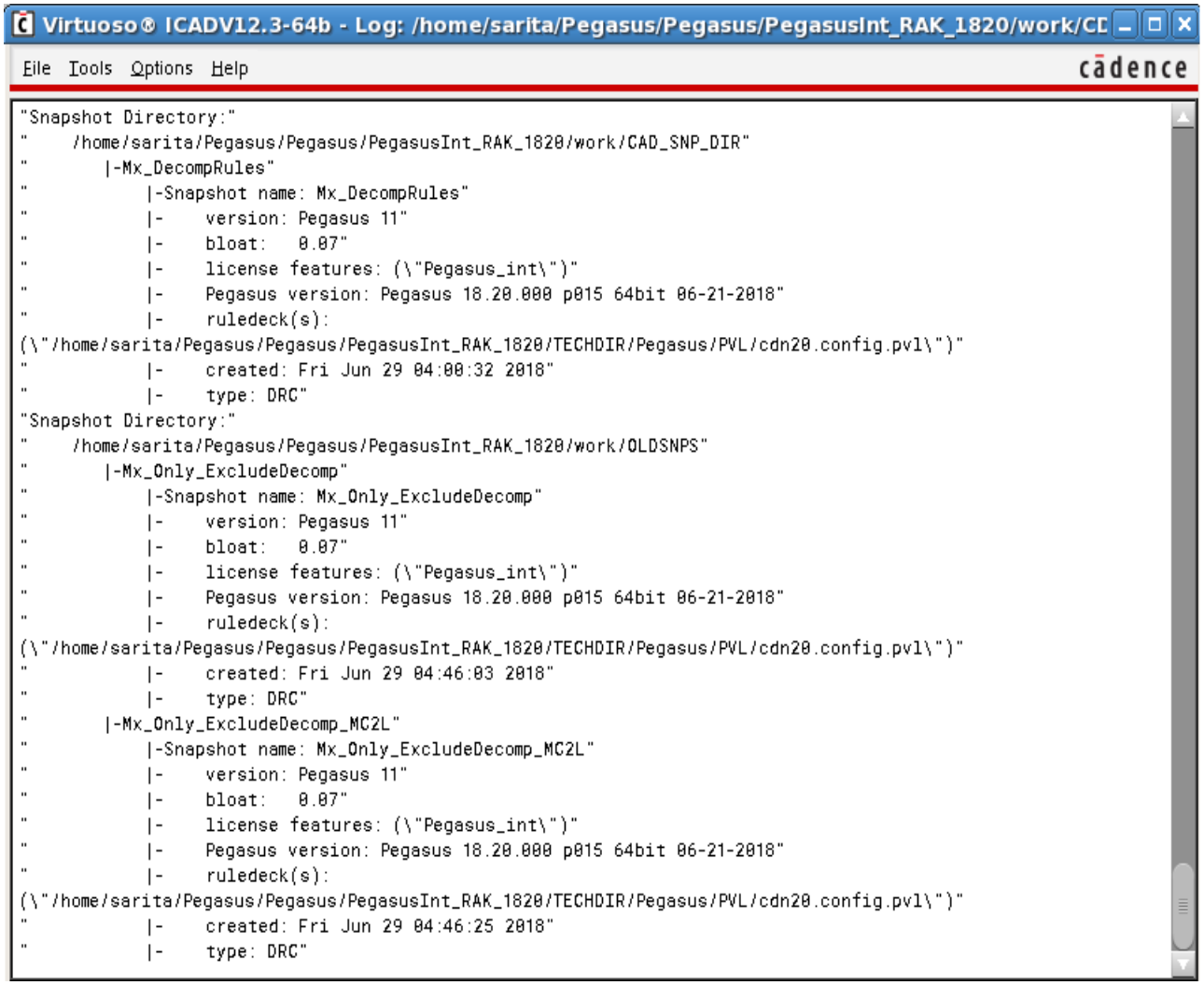
For each component listed in snapshot header, the detail of usage is documented in the following table:

Snapshot Header Overview

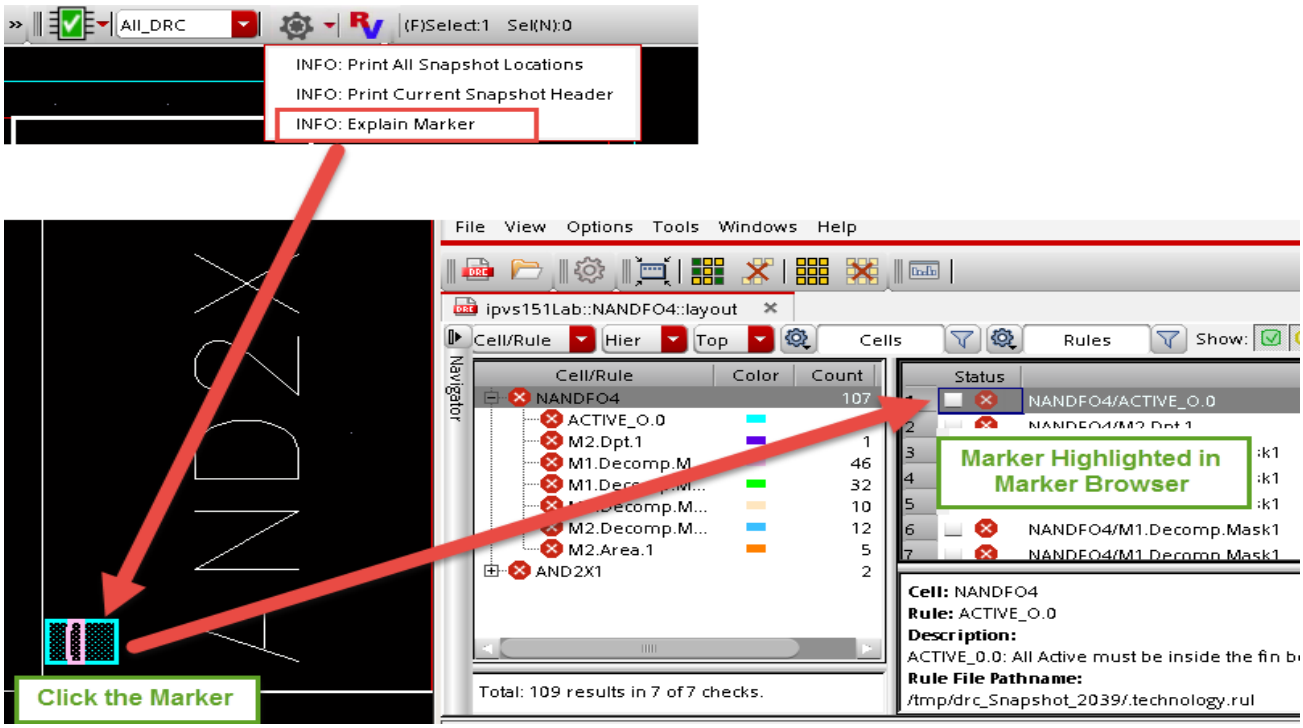
Header Info	Description
`Snapshot Name`	■ Current active snapshot
`Snapshot version`	■ Snapshot version issued by Pegasus ■ Version number is updated when Pegasus rule-deck to snapshot compiler engine supports new Pegasus command.
`The bloat value`	■ Area expansion threshold in X and Y directions for Pegasus to perform final area of checking. ■ It is the maximum value determined from selected checks stored in snapshot or custom bloat value defined in snapshot creation form.
`Licenses Features`	■ License required to run Pegasus Interactive for a given selected checks stored in the snapshot. ■ License issuing is dependent on the content of selected rule decks.
`Pegasus Version`	■ Pegasus binary used to create this snapshot
`Source Rule Deck`	■ The source rule deck(s) this snapshot is generated from
`Date of creation`	■ Time stamp when snapshot is first created

You can print all available snapshots' header information and locations where they are stored into by invoking *INFO: Print All Snapshot Locations*. Pegasus Interactive composes a

snapshot lookup map and populates this information in CIW. It is grouped by snapshot directory. Following figure shows lookup map outline.



The `INFO: Explain Marker` option allows you to select a marker in the layout viewer and that marker will then be highlighted in the Marker Browser window.



Note: To use this option, you must select Pegasus Results Viewer as browser.

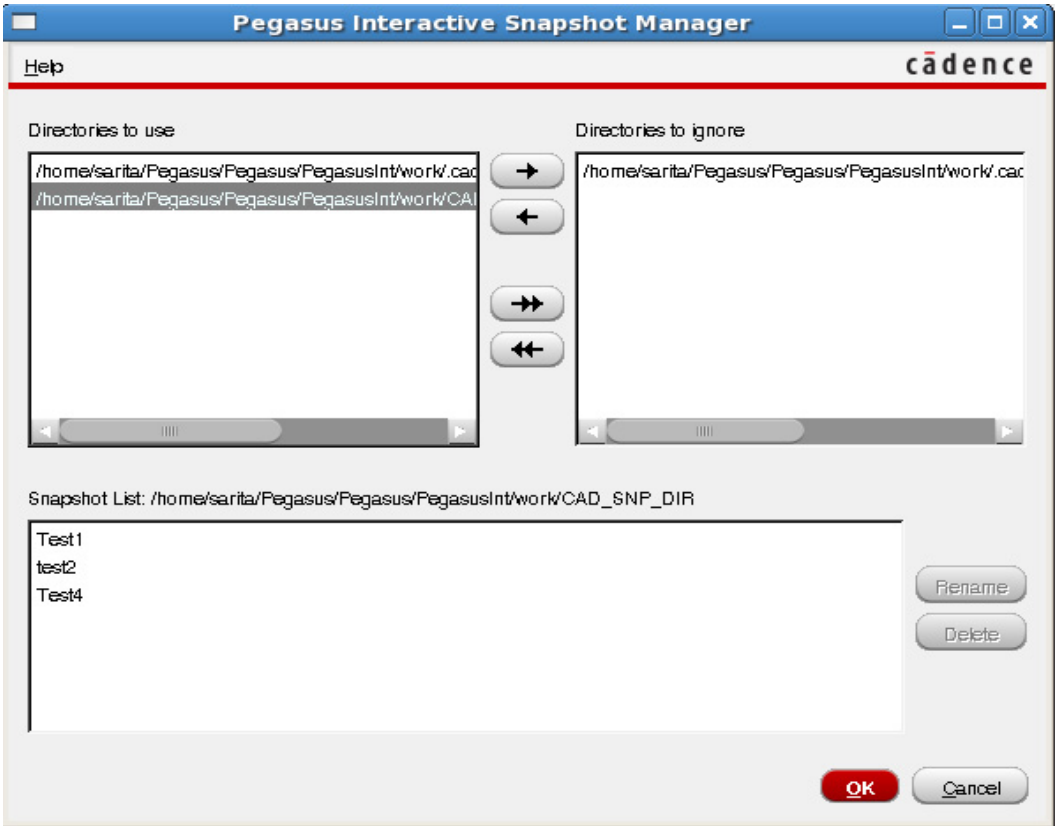
The Snapshot inquiry utility can be de-activated through shell variable `PegasusInt_Info no`.

Managing Snapshots

The snapshot accessibility becomes a challenge when multiple snapshot directories are available in their working environment. Since total of snapshot display counts has a limit of 10, some snapshots are hidden inside snapshot combo field unless you scroll down to the drop-down list. In addition to this limitation, items in the snapshot drop-down list gives you no information that which snapshot directory they are stored under.

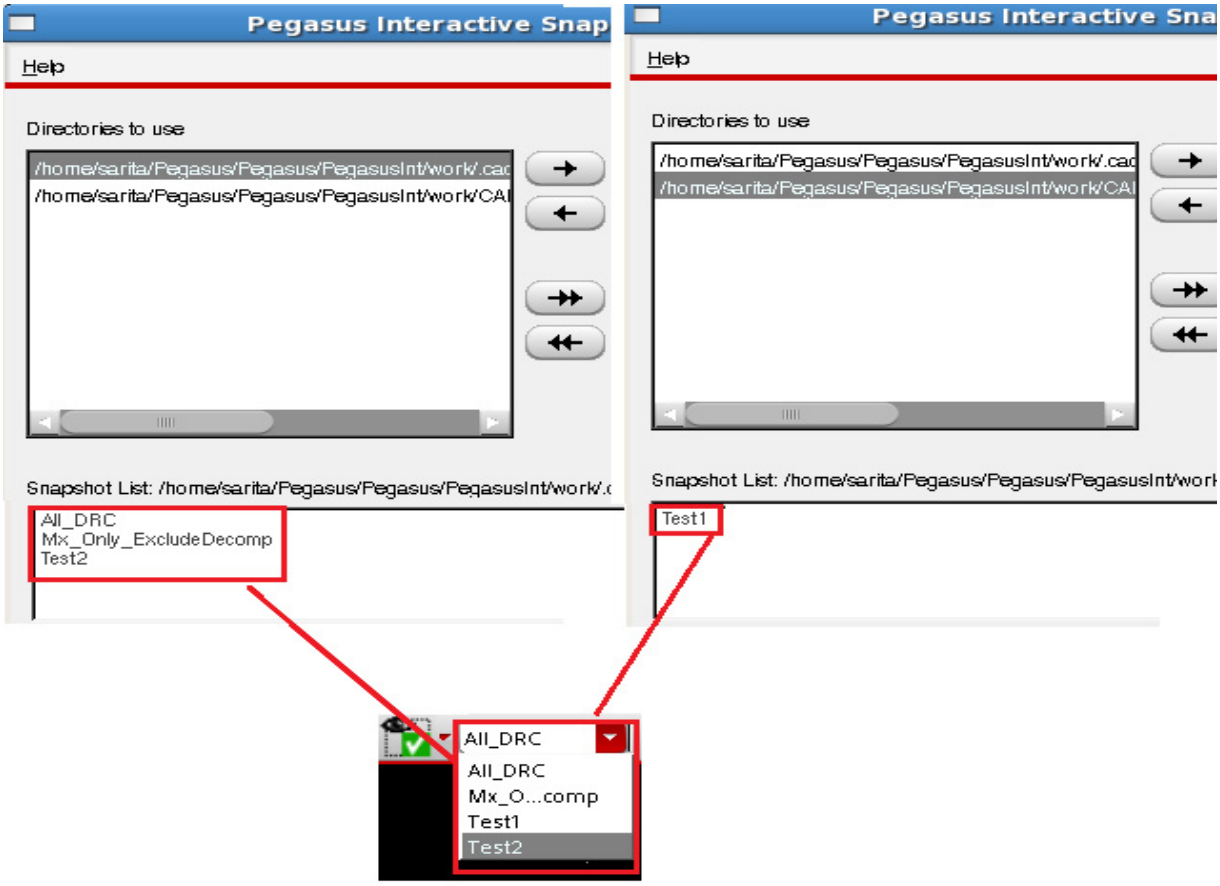
Thus, Pegasus Interactive allows you to selectively display snapshots based on snapshot directories in the snapshot drop-down list. It is through the *Manage* button under the *Snapshots* field in the *Pegasus Interactive Run Options.* You can enable it through `PegasusInt_EnableSNPManagement`.

Clicking the *Manage* button opens the *Pegasus Interactive Snapshot Manager* form:

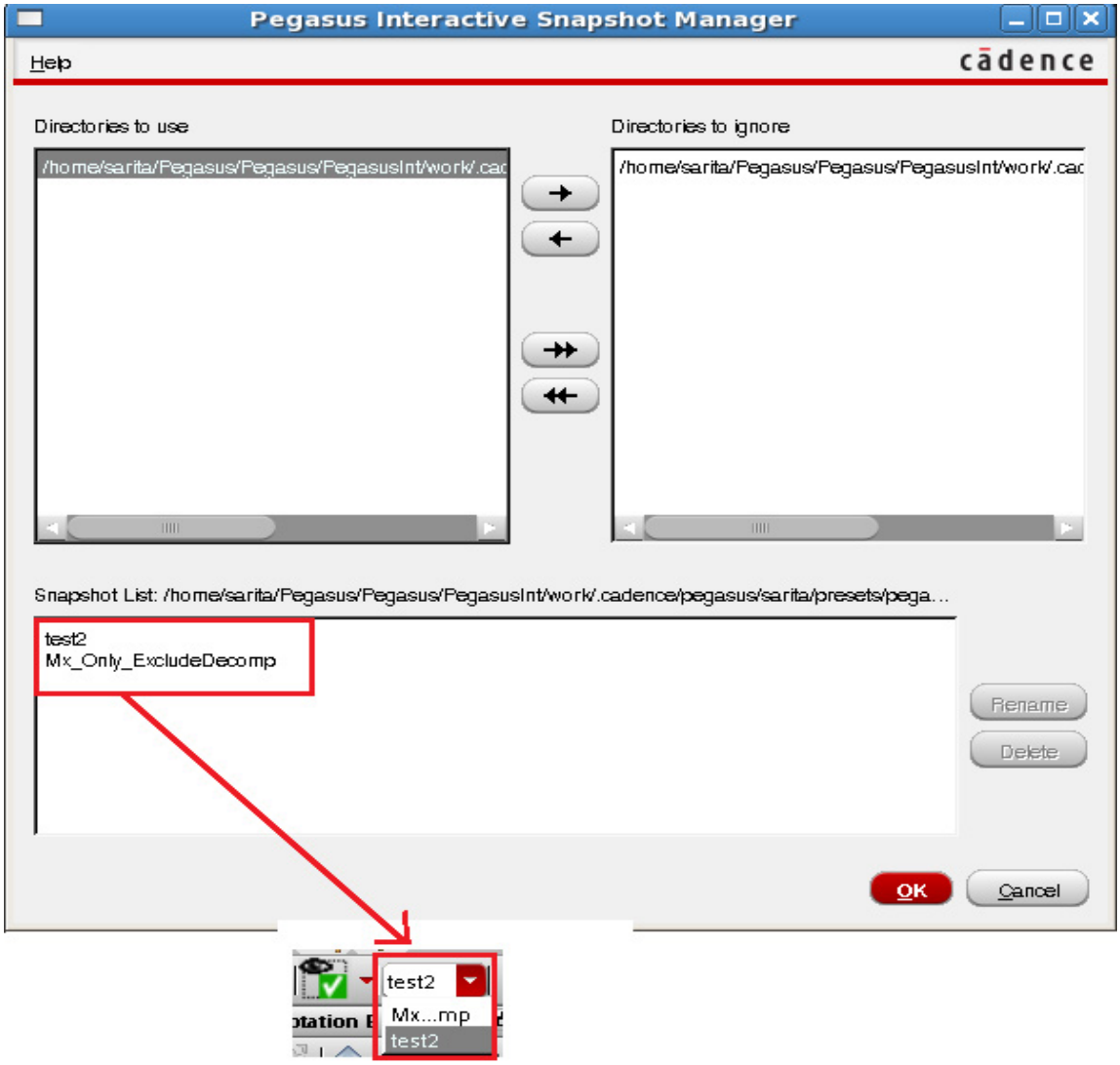


The form lists all snapshot directories defined in environment variable `PegasusInt_SnapshotsDirs`. If `PegasusInt_SnapshotsDirs` is undefined, then the form lists default snapshots directory `<$PWD>/.cadence/pegasus/<$userid>/presets/pegasusint/snapshot`.

By default, all fetched snapshot directories are listed in *Directories to use* block. This means that snapshots from these directories will be displayed in the toolbar.



If you do not want to display snapshots from specific directory in the toolbar, then move the directory to the *Directories to ignore* block using the arrow key:



You can rename the snapshot name by selecting the snapshot name and clicking the *Rename* button. Similarly, you can delete a snapshot by selecting a snapshot and clicking the *Delete* button.

Upgrading Snapshots

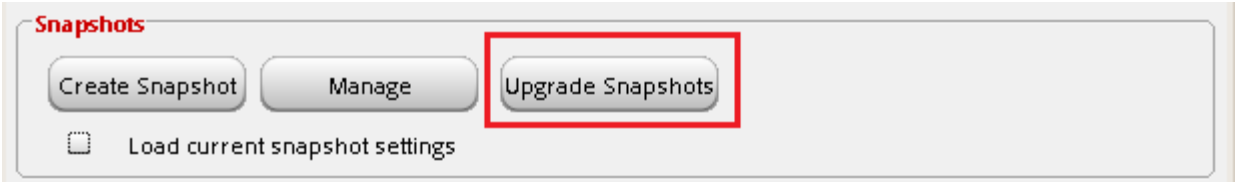
Pegasus guarantees snapshot forward compatibility for base release life cycle. In other words, engine can interpret a snapshot created by base release and its ISRs/HFs/EHFs. However, Pegasus Interactive still checks snapshot readability during layout opening. If the incompatibility is detected, it outputs following warning messages in CIW:

```
*WARNING* Pegasus Interactive: Following snapshots are generated using different
Pegasus that is incompatible with current Pegasus installation and will not be
populated under Snapshot list. Please re-generate the snapshots with current
Pegasus version.
```

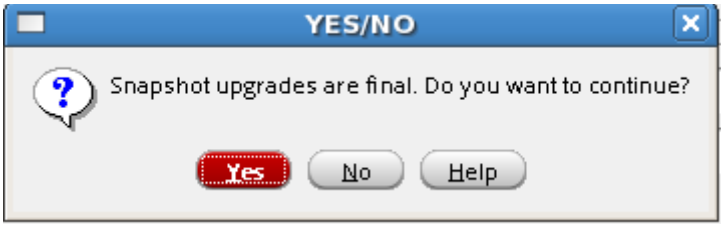
Then, these version incompatible snapshots will not be available to choose from the snapshot drop-down list.

Note: Pegasus rule parser may be changed in ISR or hot fix to achieve better results. In this case snapshot version will be changed. New Pegasus Interactive version will read old snapshots generated by previous Pegasus Interactive versions of the same base release. However, changes in rule parser will not be visible. To take advantage of new Pegasus rule parser, snapshot need to be upgraded.

You can invoke *Upgrade Snapshot* utility that runs a batch upgrade on all incompatible snapshots. The *Upgrade Snapshot* button in the *Snapshots* section of the *Pegasus Interactive Run Options* form is active if there are incompatible snapshots in snapshots directory. You can make it available through `PegasusInt_Maintenance`.

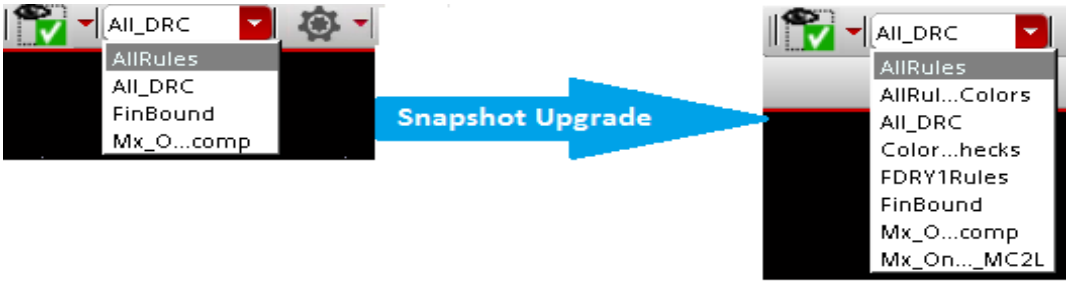


Once you click *Upgrade Snapshot*, following pop-up window is shown.



Above pop-up window is to remind you that snapshot upgrade is final and you cannot revert the changes once the commitment is made. Once you make the upgrade commitment, Pegasus Interactive precedes with snapshot upgrade and upgrade status is printed out in CIW. During the time of snapshot batch upgrade, Virtuoso is not accessible.

Once the snapshot upgrade is complete, all snapshots are populated in the snapshot drop-down list in the toolbar.



Portability of Snapshots

Pegasus Interactive snapshots are portable between different locations. It means that if rule deck contains the `include` statements of additional rule deck files, then in case you create snapshots and copy it to different location from which old paths are not visible, Pegasus Interactive will generate results for given snapshot.

However, if rule deck contains reference to an additional data like GOLDEN GDSII/OASIS files, you will need to use environment variable `PegasusInt_Data` to make snapshot portable between different location. This is because Pegasus Interactive snapshot stores absolute paths which may be not visible from other locations.

Example:

```
> setenv Pegasus_DK /usr/dk/Pegasus_data
```

Rule deck contains line:

```
layout_path $Pegasus_DK/GOLDEN_GDS/golden1.gds GOLDEN;
layout_path $Pegasus_DK/GOLDEN_GDS/golden2.gds GOLDEN;
```

In the Pegasus Interactive snapshot these lines will be stored as:

```
layout_path /usr/dk/Pegasus_data/GOLDEN_GDS/golden1.gds
layout_path /usr/dk/Pegasus_data/GOLDEN_GDS/golden2.gds
```

Thus, if path `/usr/dk/Pegasus_data/GOLDEN_GDS/` is not visible from end-user location, Pegasus Interactive will issue an error and will not generate any results.

To make snapshot portable, perform the following steps:

1. User, who generates snapshot should set:
   a. `setenv PegasusInt_Data /usr/`**`dk`**`/Pegasus_data`
   b. Generate snapshot. Paths to golden GDS files will be stored as:
      `${PegasusInt_Data}/GOLDEN_GDS/golden1.gds.`
      `${PegasusInt_Data}/GOLDEN_GDS/golden2.gds.`
2. End-user:
   a. Install design kit in local directory. For example: `/usr/`**`user`**`/Pegasus_data`
   b. `setenv PegasusInt_Data /usr/`**`user`**`/Pegasus_data`
   c. Run Pegasus Interactive.

Checking that Snapshot is Generated by Up-to-Date DK Version

If DK team wants to make sure that their users use snapshot generated by latest DK versions then they can create a SKILL code which checks validity of snapshot based on `PegasusIntValidateSnapshot` SKILL function and then define how to proceed further: either allow to use with warning or hide snapshot.

General assumption is that the customer implements procedure of fixed name and list and order of arguments.

1. If `PegasusIntValidateSnapshot(snp_name snp_dir snp_comment)` returns:
   a. `t`
      i. Snapshot name will show up on Pegasus Interactive Toolbar
      ii. There will be no message in CIW
   b. `nil`
      i. Snapshot name will not show up on Pegasus Interactive Toolbar
      ii. Following warning message will be shown in CIW:

      ```
      Pegasus Interactive: snapshot <snp_name> located in <snp_dir> didn't pass validation against comment <snp_comment>
      ```
   c. Non empty `error_message (string)`
      i. Snapshot name will not show up on Pegasus Interactive Toolbar
      ii. Following warning message will be shown in CIW:

      ```
      "Pegasus Interactive: <error_message>"
      ```
   d. Empty `error_message (string)`
      i. Snapshot name will not show up on Pegasus Interactive Toolbar
      ii. There will be no message in CIW
   e. Anything else (list, symbol, and so on …) will be treated as nil like point # b above
2. This procedure must be loaded before first layout cell is opened (this is when the SKILL context is loaded and snapshot validation is run on init).

   Examples (called from `.cdsinit`):

   Allowing snapshots located only in a specific directories:

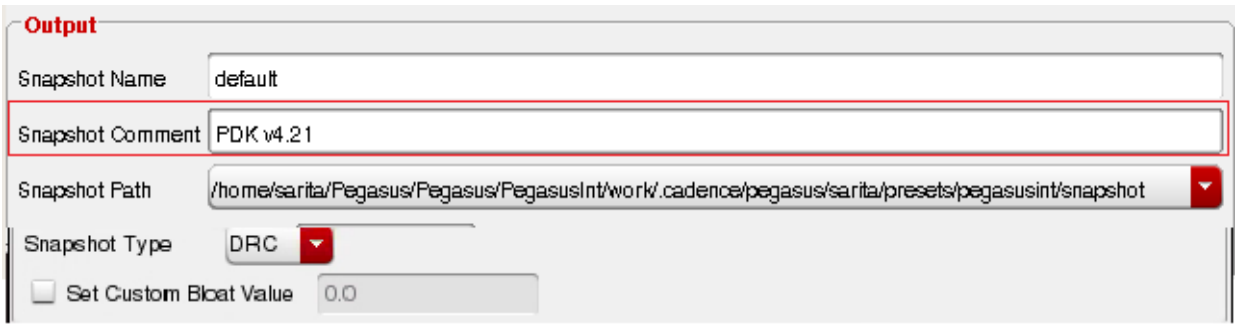
```
procedure( PegasusIntValidateSnapshot( snp_name snp_dir snp_comment)
    let((project_dir valid)
     project_dir = getShellEnvVar("Pegasus_N7_DIR")
     if( project_dir && simplifyFilename(project_dir) == simplifyFilename(snp_dir)
     then
        valid = t
     else
        valid = sprintf(nil "Refusing snapshot %s from %s. Only snapshots from %s
        directory are enabled." snp_name snp_dir project_dir)
        );if
     valid
    );let
);procedure
```

Allowing snapshots based on processing its comment line:

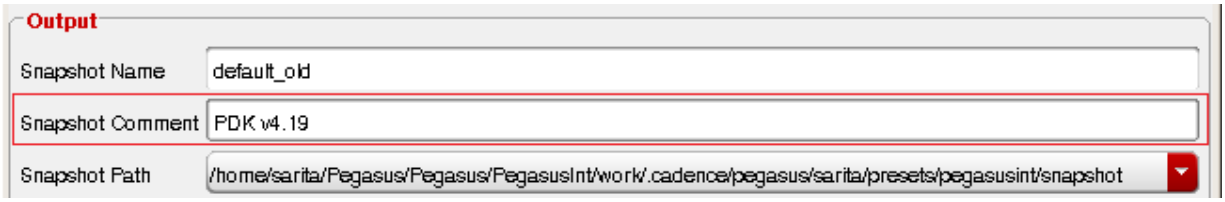
```
procedure( PegasusIntValidateSnapshot( snp_name snp_dir snp_comment)
    let(()
        rexMatchp( " N[5-7]" snp_comment)
    );let
);procedure
```

Example

1. Create snapshot and add comment:



2. Create another snapshot and add comment:



3. Create SKILL and load it at Virtuoso start:

```
;; if SNP comment does not match PDK v.4.21 then snp will not be shown in the list,
;;warning will be issued.
procedure( PegasusIntValidateSnapshot( snp_name snp_dir snp_comment)
    let(()
        rexMatchp( "PDK v.4.21" snp_comment)
       );let
);procedure
```

4. Open layout window. Just default snapshot is available. default_old is hidden. Warning is issued to CIW.

4

Invoke Design Rule Checking

This chapter provides a detailed description of Verify Design mode and methods of customizing its behavior.

The content of this chapter is organized as follows:

Overview: Run Behavior

Related Topics

- Run Results on page 104
- Violation Markers Handling on page 104

Run Results

The run results are stored under `$PWD/PEGASUS_INT_RUNDIR_VD`. Within this run directory, run results are stored by cell under `<cellName>_<viewName>_<libraryName>` directory. Within this directory, file run results are stored in `<cellName>_<viewName>_<libraryName>.drc.db` file. Unlike Pegasus sign-off run, Pegasus Interactive only outputs following commands defined in the rule deck:

- `results_db` with `-ascii`
- `density` with `-rdb`

Any commands asking Pegasus to output data found in the snapshot are ignored during DRC for optimum performance.

All run logs can be found by default under `$PWD/PEGASUS_INT_RUNDIR_VD/<cellName>_<viewName>_<libraryName>/PegasusInt.log`.

You can change the default Pegasus Interactive run directories through `PegasusInt_Dir`.

Violation Markers Handling

Since Pegasus Interactive is an area specific run, DRC violations populated on the layout is per run base. In other words, any violations created by the previous Pegasus Interactive run are removed before loading the latest run results on the layout.

Also, Pegasus Interactive run does not modify the layout. Initially, violations are stored in an ASCII file format. Violation generation and display is different depending on whether you use Pegasus RV or Annotation Browser. If it is Annotation Browser, OA markers are generated based on ASCII results and populated on the layout. These data are stored in a virtual memory at first, but you can choose to save violations within the cellview by clicking *Save*. If not, once Layout is closed, these OA markers are lost. On the contrary, Pegasus RV can populate the latest run results on demand. As long as ASCII results are accessible by

Pegasus RV, you can invoke Pegasus RV on the toolbar and violation highlights are populated in the layout.

Verify Design Mode

This Pegasus Interactive mode is an area specific and on-demand DRC. You can specify initial area of checking ranged from the whole cellview to one changed portion of the layout. There are three initial area of checking modes available:

- Current CellView
- Changed Area
- Visible Area

Both *Current CellView* and *Visible Area* are static DRC while *Changed Area* is a dynamic DRC. By definition, a static DRC is run on polygons within the area of interest as it-is. On the contrary, dynamic DRC checks only on the modified portion of layout, which is tracked by Pegasus Interactive. Any editing done from the moment layout is opened prior to DRC is considered as the modified portion of the layout. This layout editing tracking is stored temporarily in the virtual memory and is reset after Pegasus Interactive completes DRC.

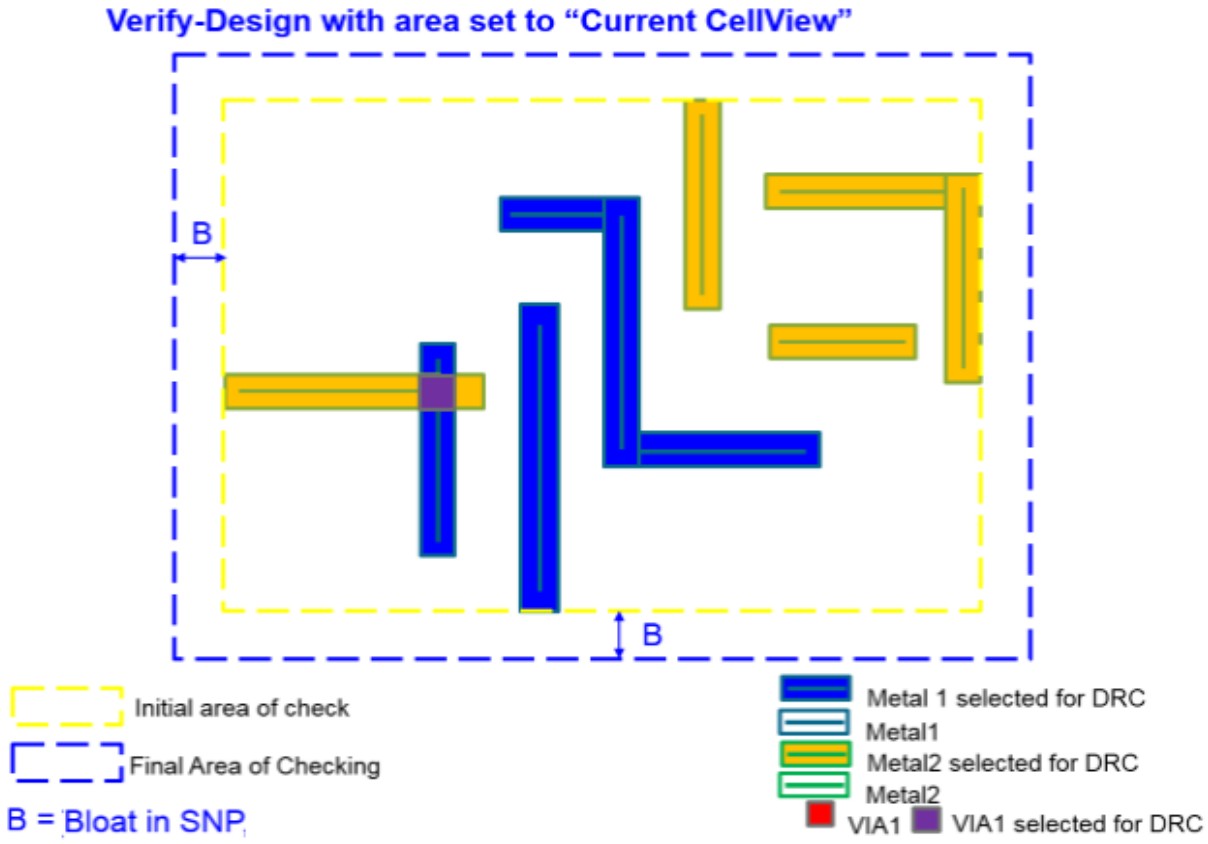
Regardless the hierarchy depth displayed in the layout, Pegasus Interactive checks on all polygons found within the area of checking.

In general, you can interactively set area checking mode for a given window and this is a Virtuoso session based setting. Furthermore, you can control the default *Verify Design* mode area setting through `PegasusInt_InitArea`.

Current CellView

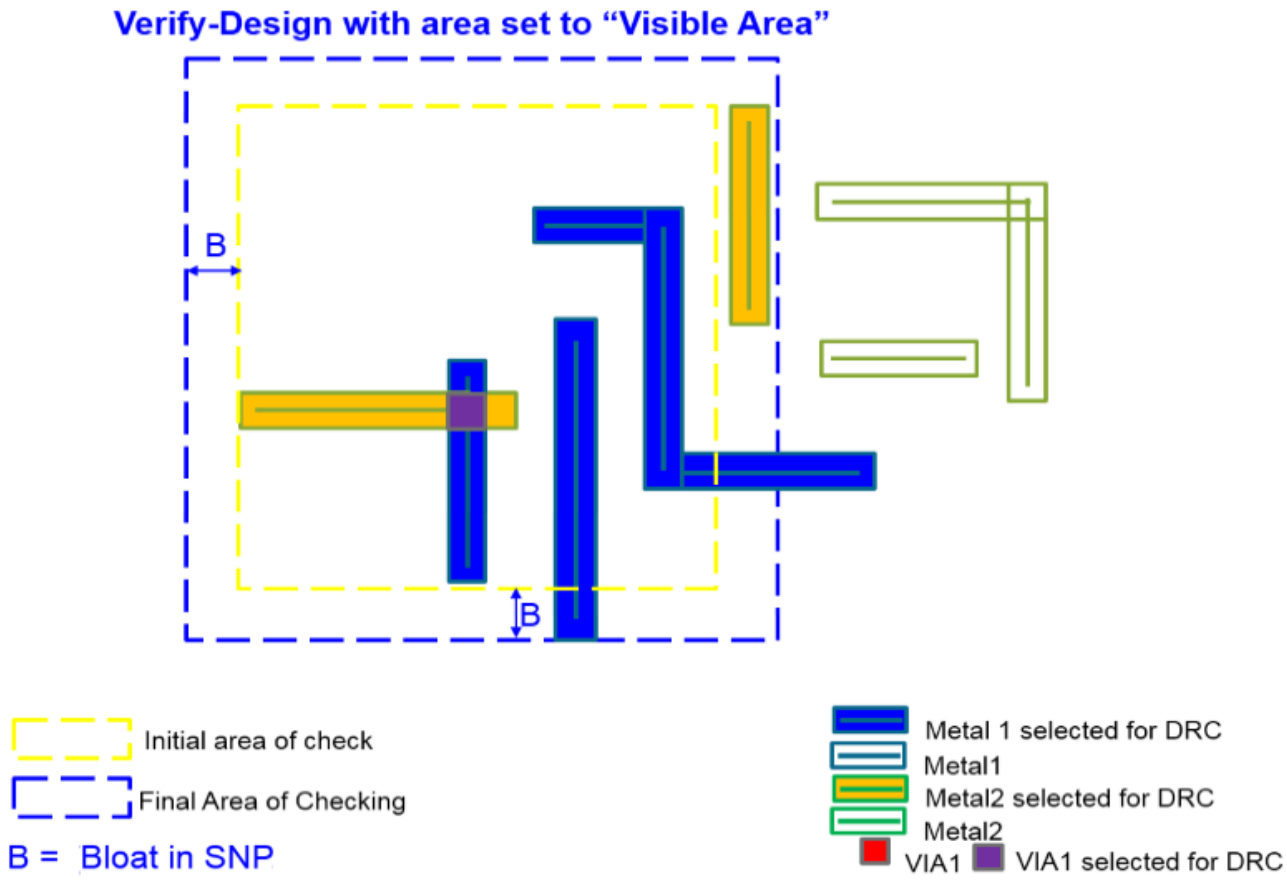
Verify-Design mode area checking is default to *Current CellView*. Regardless of current viewing area of the layout, Pegasus Interactive always checks the entire cellview when this

area mode is on. Even though the whole cellview is checked, the final area of checking depends on the bloat value.



Visible Area

This mode is a *What-You-See-Is-What-You-Check* mode. The current viewing area is first taken in as an initial area of checking. Pegasus Interactive then computes final area of checking by factoring bloat value defined in the current active snapshot.

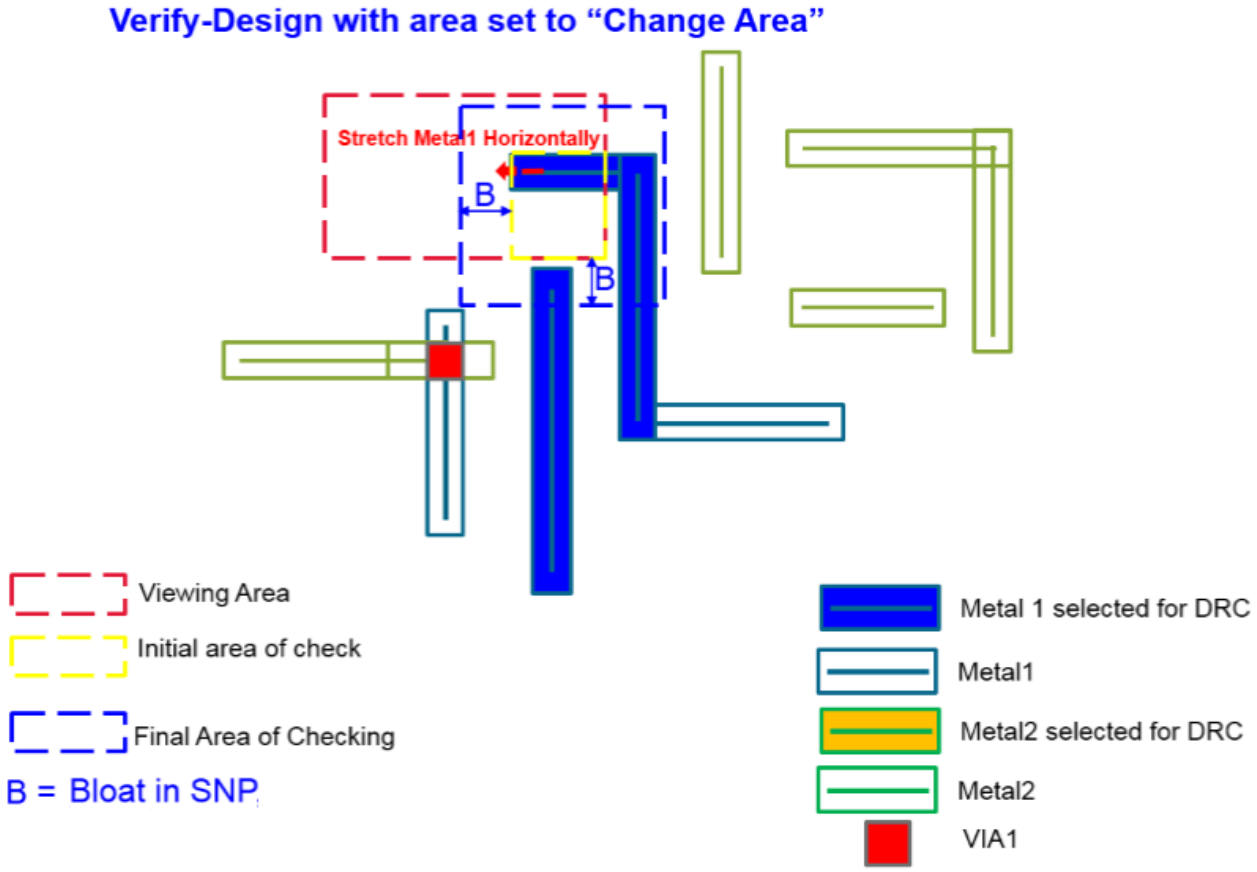


Changed Area

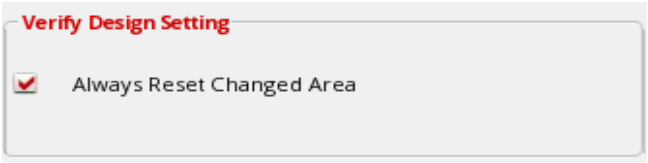
This mode allows you to conduct multiple layout editing and tracks these modification between each DRC. Once you are satisfied with the editing, you can invoke Pegasus Interactive to check on the final changed area. Its final area is computed based on the final layout modifications plus bloat value defined in the current active snapshot.

Pegasus Interactive starts to track modified polygon after *Changed Area* is enabled. The tracking is only between each Verify Design run with *Changed Area* enabled. Once DRC run

is complete, the changed area tracking is reset. Also, any change made prior to *Changed Area* enablement is not tracked by Pegasus Interactive.



To disable automatic reset of Changed Area tracking deselect *Always Reset Changed Area* check box in the *Run Options* form.



Once deselected the tracking of changed area is kept between each Verify Design runs. You can reset tracking using toolbar button *Clear Changed Area*.

Note that *Clear Changed Area* button is available only when selected mode is *Changed Area* and *Always Reset Changed Area* settings of the *Run Options* form is deselected.

5

SignOff Fill: Snapshot

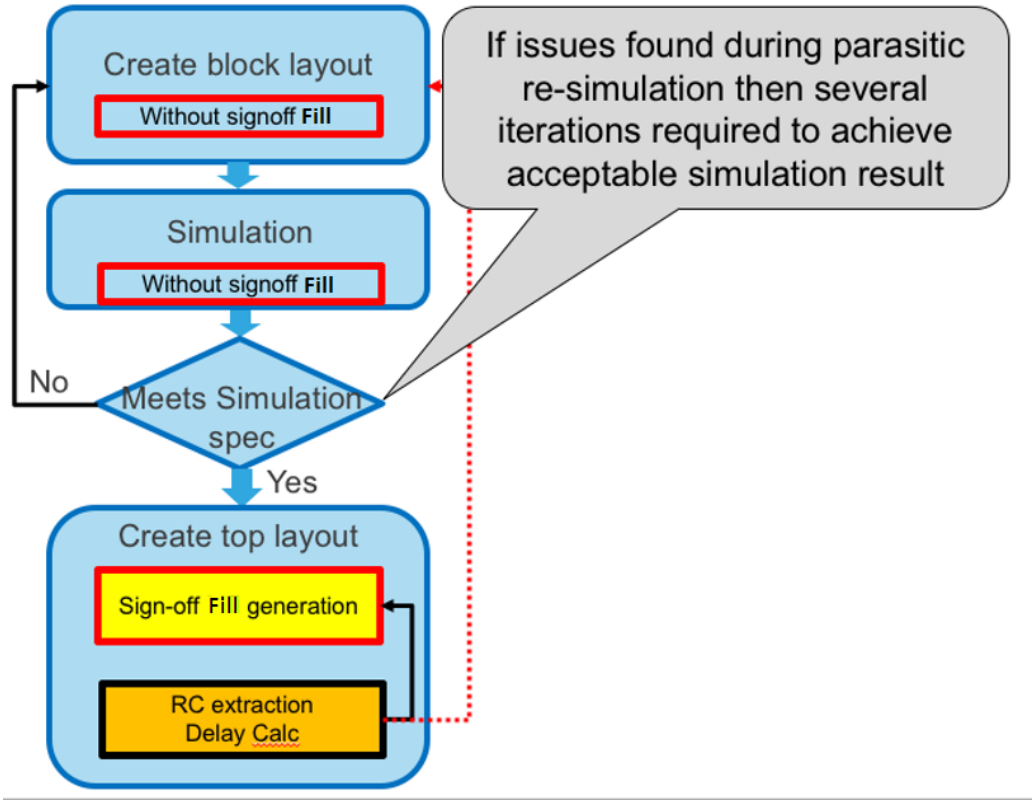
This chapter introduces you to the concept of SignOff fill and how to set up the form to run signoff fill. For the basics of Snapshot creation, see All About Design Rules: Snapshot on page 57.

The content of this chapter is organized as follows:

Pegasus Interactive SignOff Fill Overview

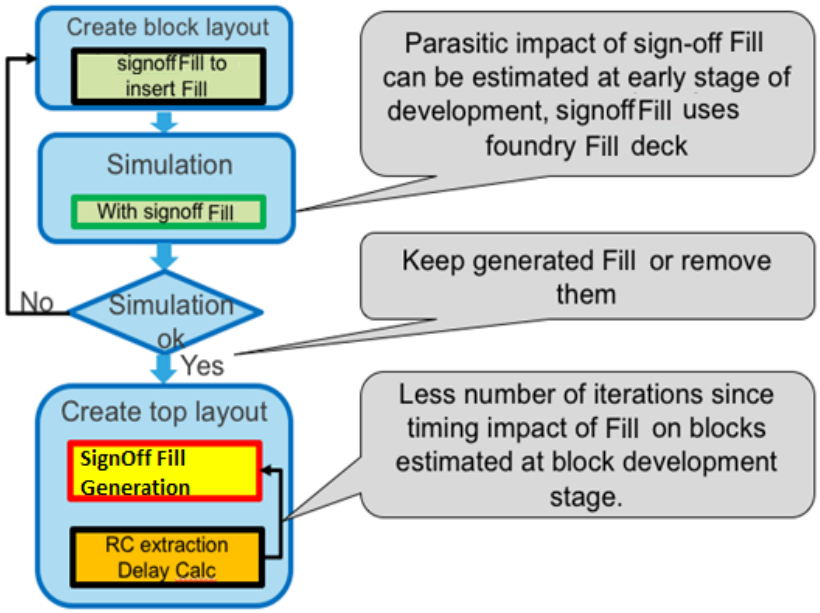
Pegasus Interactive SignOff Fill allows you to generate dummy fills using foundry fill decks and automatically map generated dummy fills into Virtuoso OA database.

Traditional Fill Flow



Pegasus Interactive SignOff Fill Use Model

To reduce the number of iterations during cell/block development you can use the SignOff Fill feature to generate dummy fills inside block and therefore estimate timing impact of dummy fills at early development.



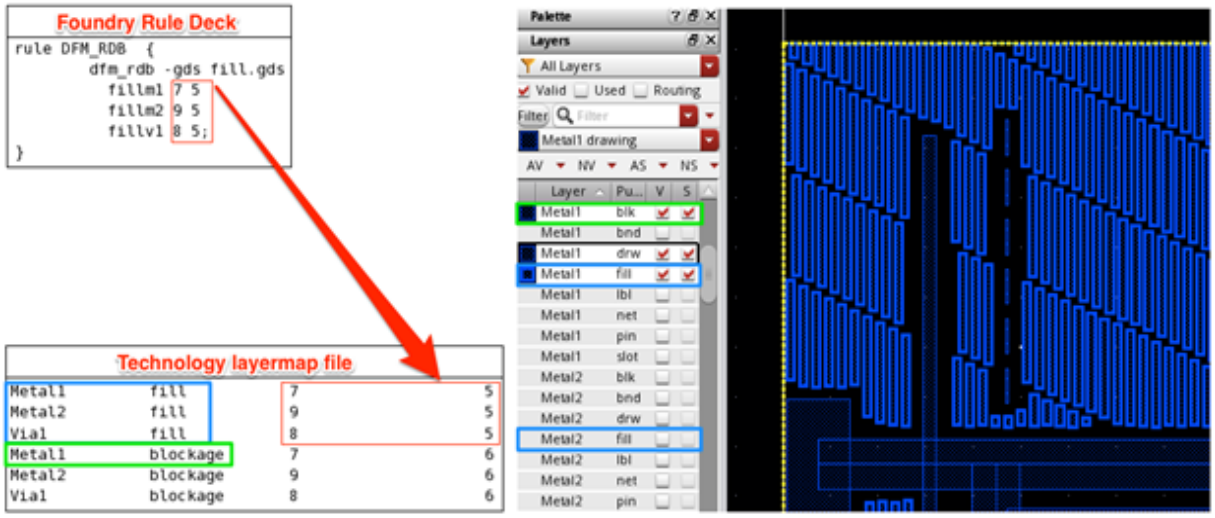
Pegasus Interactive SignOff Fill Features

- Support for qualified foundry fill deck.
- Generated fills immediately included into the block lib/cell/view. No need to manually merge generated fills with the design.
- Tight integration into Virtuoso Pegasus Interactive flow.
  - Ability to select layer(s) to fill from the Virtuoso layer palette (active or visible)
  - Complete snapshot flow support: create snapshot, select snapshot from the Pegasus Interactive toolbar, and run fill.
  - Less run time on advanced nodes due to elimination of the rule parser stage.
- Fill in area support.
- Incremental fill: fill design, modify design, and re-fill in the modified area.
- Blockages support:

- Draw blockages on layout using the blockage purpose layer and run fill.

Technology Setup

To backport the generated dummy fills into the OA technology layermap file, you must define the mapping between the fill layer and layer-purpose pair:



Blockage layer number/datatype must be defined in the layer_def/layermap section of the foundry fill deck

There must be proper mapping between the generated dummy fills in the foundry rule deck and the Virtuoso technology library. If there is no proper mapping, then the generated dummy fills will not be shown in Virtuoso.

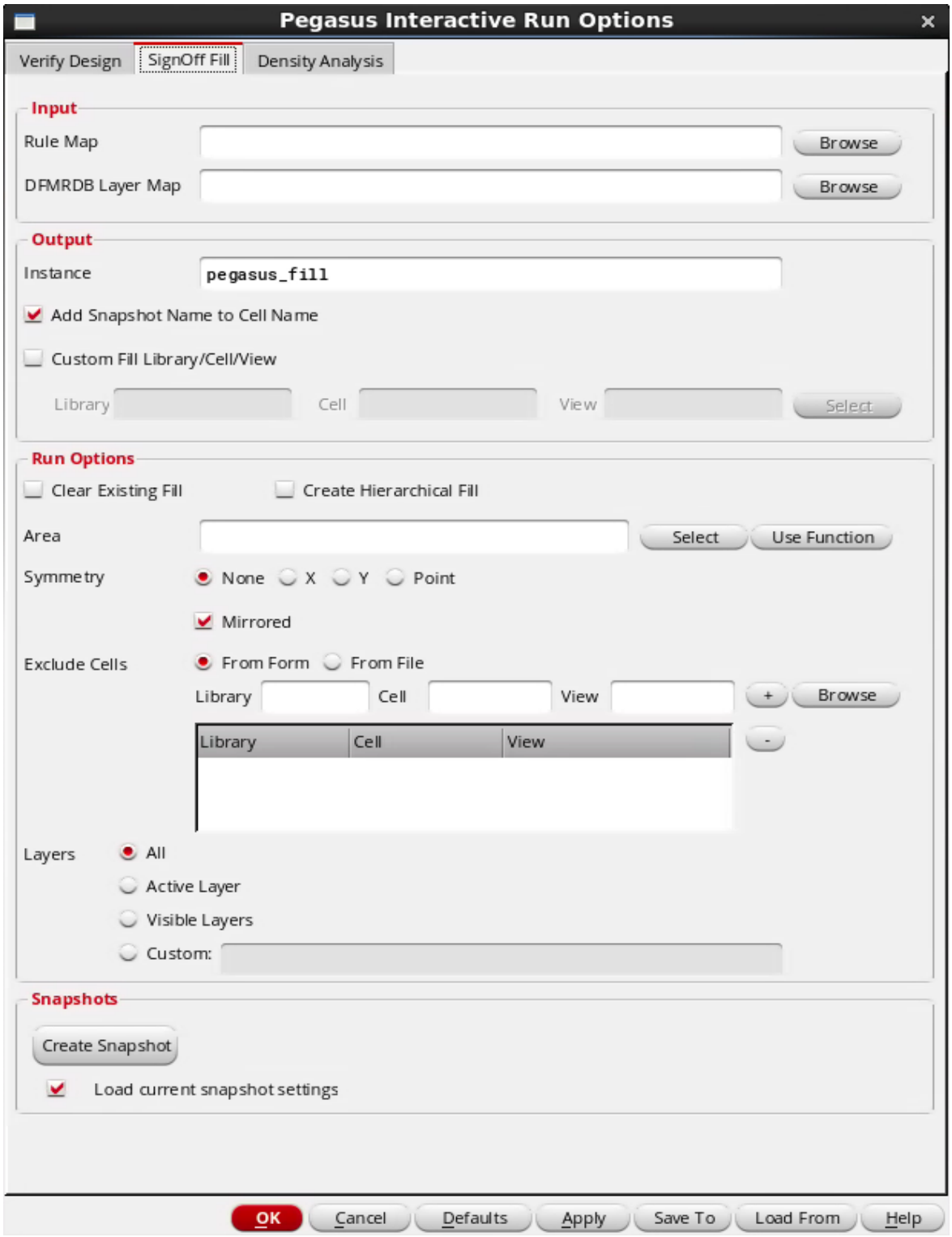
Creating Pegasus Interactive Fill Snapshot

Pegasus SignOff Fill feature is snapshot-based, same as Pegasus Interactive Verify Design. For detailed description of fields in the Pegasus Interactive Snapshot Creator form, see All About Design Rules: Snapshot on page 57.

Perform the following steps to create Pegasus fill snapshot:

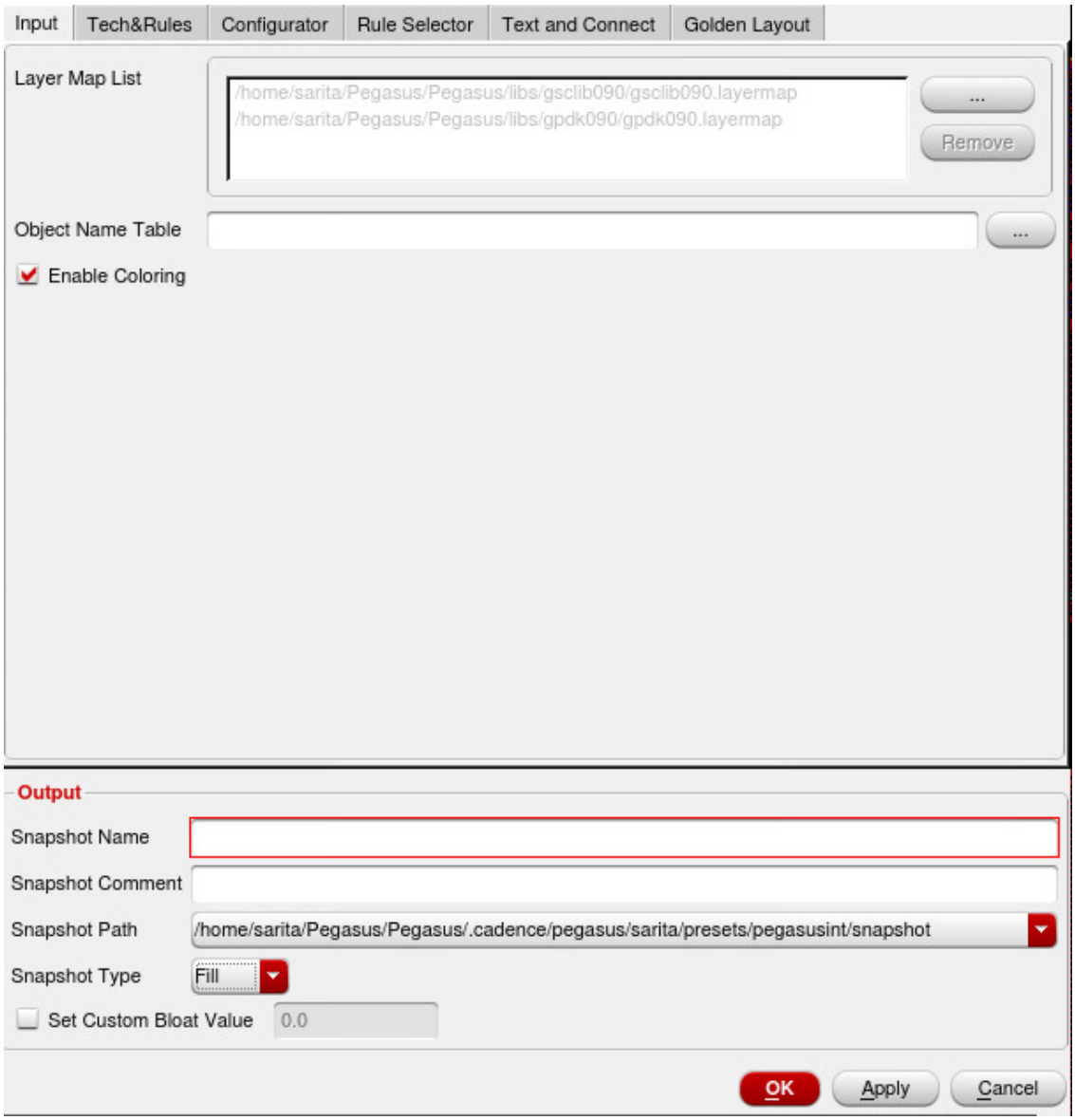
1. Click the *Pegasus Interactive Run Options* icon on the *Pegasus Interactive* toolbar. The *Pegasus Interactive Run Options* form opens.

2. Select the *SignOff Fill* tab.



3. Click *Create Snapshot*.

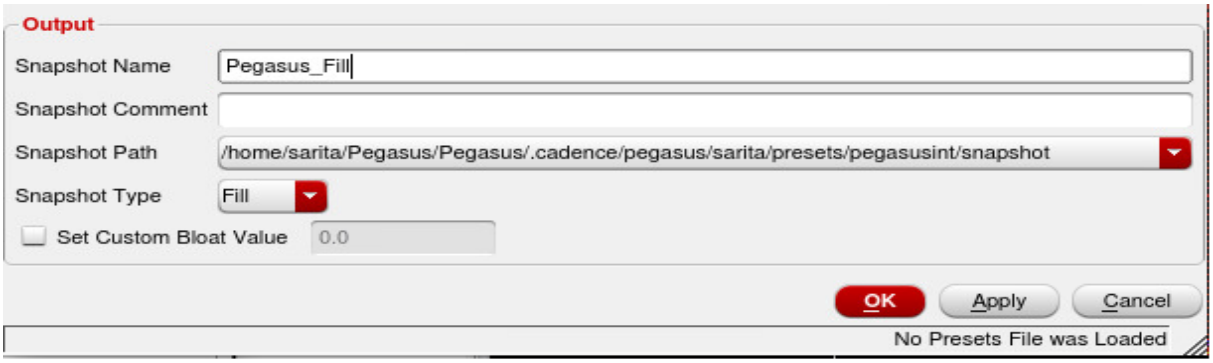
   The *Pegasus Interactive Snapshot Creator* form opens.



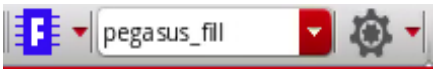
   For detailed description of the *Pegasus Interactive Snapshot Creator* form fields, see Pegasus Interactive Snapshot Creator Form on page 68.

4. Enter the required details in the form. The mandatory fields are rule file, layermap file and snapshot name.

5. In the *Output* section, enter the snapshot name.



6. Click *OK* to generate a snapshot and close the form. Then close the *Pegasus Interactive Run Options* form. The *Pegasus Interactive* toolbar shows `pegasus_fill` snapshot.



The *Run Pegasus Interactive SignOff Fill* icon on the left of snapshot name reflects that selected snapshot is type of *Fill*.

6

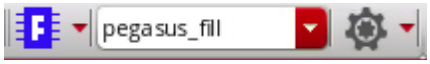
Running SignOff Fill

This chapter includes the process of running Pegasus Interactive SignOff Fill from Virtuoso and generating fills in the OA database.

The content of this chapter is organized as follows:

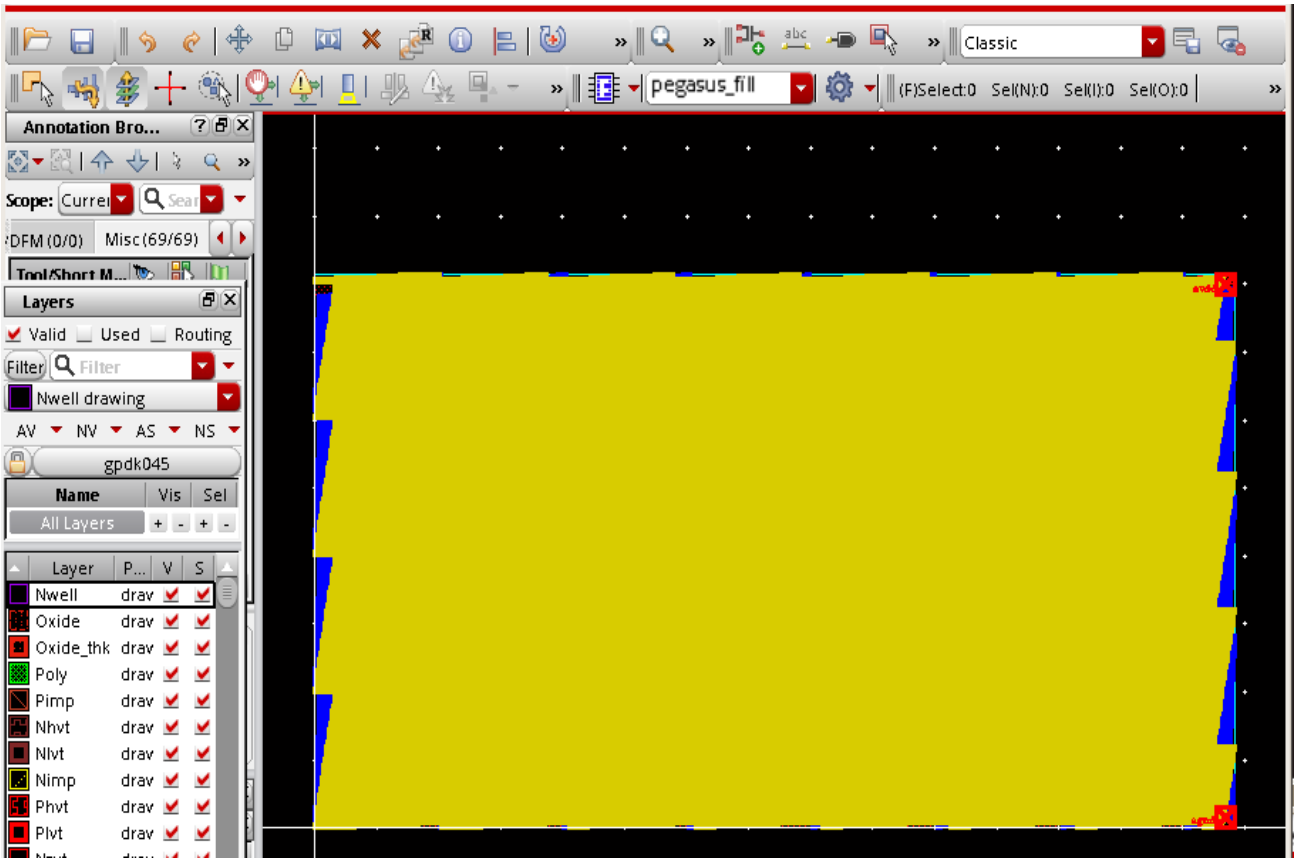
Running Pegasus Interactive SignOff Fill

After completing the settings in the *Snapshot Creator* form, you run signoff fill. The *Pegasus Interactive* toolbar shows the `pegasus_fill` snapshot. The *Run Pegasus Interactive SignOff Fill* icon on the left of the snapshot name reflects that the selected snapshot is of the type *Fill*.



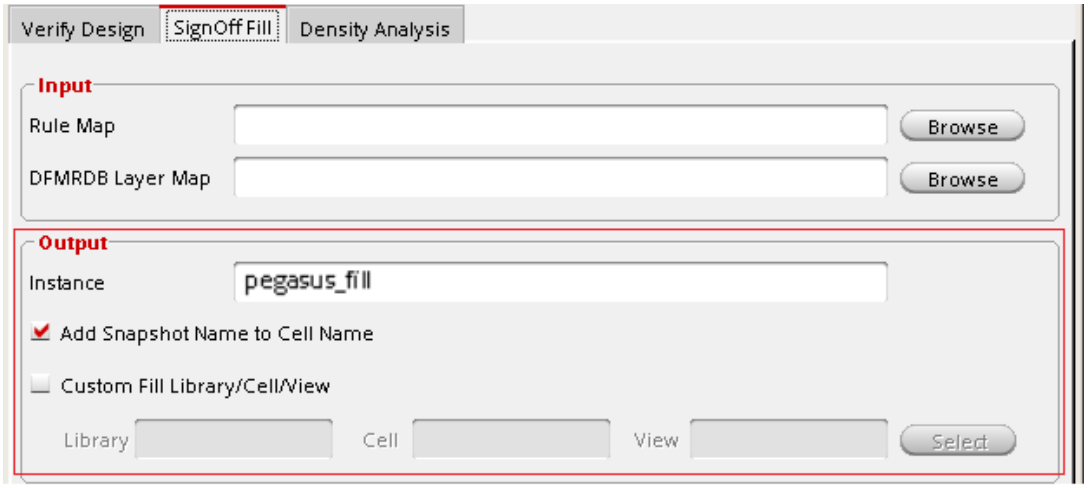
Perform the following steps to run Pegasus Interactive signoff fill:

1. Click the *Run Pegasus Interactive SignOff Fill* icon. Virtuoso CIW prints the message: *SignOff Fill is running….* Once the fill run is completed, Virtuoso CIW prints the message: *SignOff Fill is finished successfully* and dummy fills are shown on the layout canvas.



The generated fills are inserted as instance.

2. The tool by default adds primary prefix `pegasus_fill` and secondary prefix `snapshot_name` to the name of the cell and puts it to the same library as the parent cell. You can change this behavior from the *Pegasus Interactive Run Options* form.



In the steps above, the SignOff Fill was run for all layers. On similar lines, you can run this for active layers, visible layers, custom layers and for the specified area by making appropriate selections in the *Pegasus Interactive Run Options* form. For more details see, Run Settings for SignOff Fill on page 46.

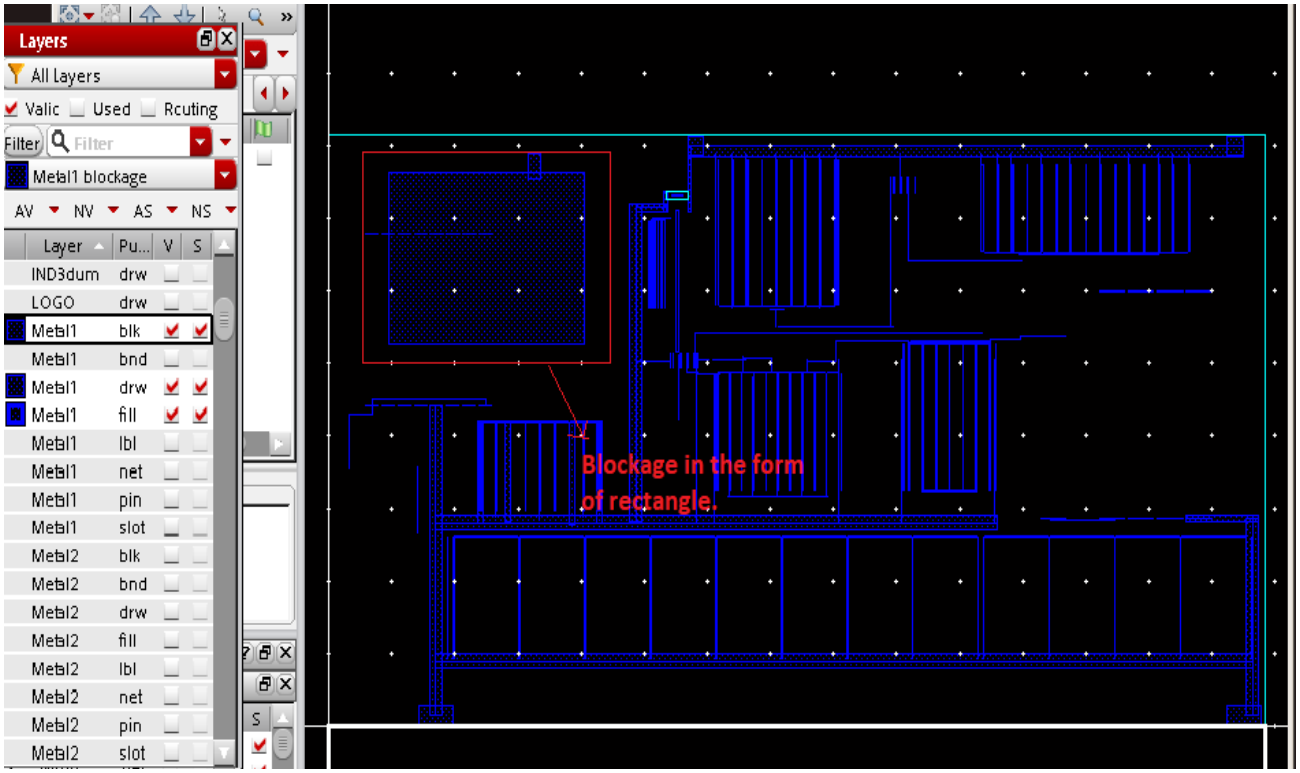
Running Pegasus Interactive SignOff Fill with Blockages

In the Pegasus Interactive SignOff Fill flow, you can create blockages to avoid dummy fills generation under blockages.

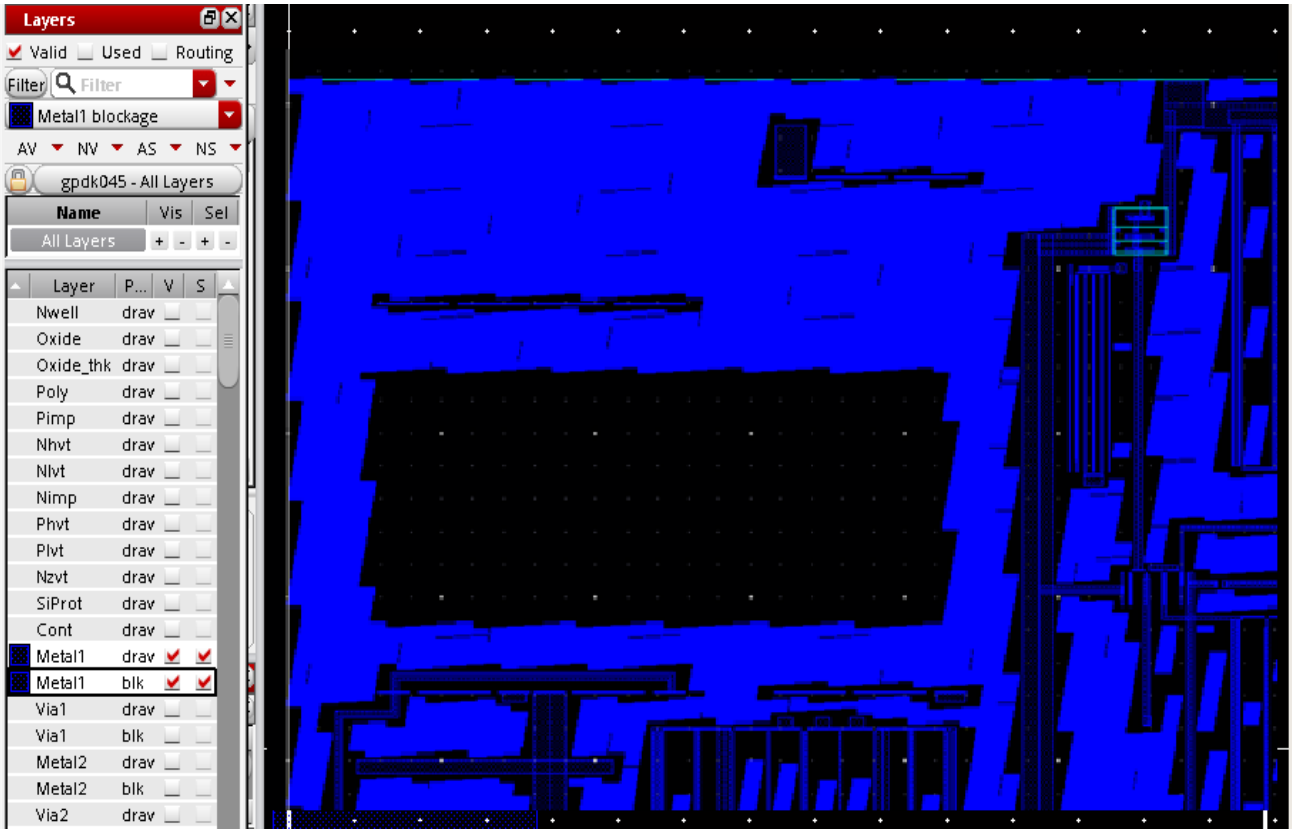
Note: The fill rule deck must support blockage layers and there must be proper mapping between the rule deck layer numbers and the blockage layer/purpose in Virtuoso.

Perform the following steps to create dummy fills under blockages:

1. Open the *Pegasus Interactive Run Options* form and go to the *SignOff Fill* tab.
2. Select the appropriate run options and click *OK*.
3. Select the blockage layer on Virtuoso Layer Palette, and create a blockage by drawing a rectangle on the layer that is part of your SignOff Fill run.



4. Run Pegasus Interactive SignOff Fill. Dummy fills are generated except for the area where blockage was created.

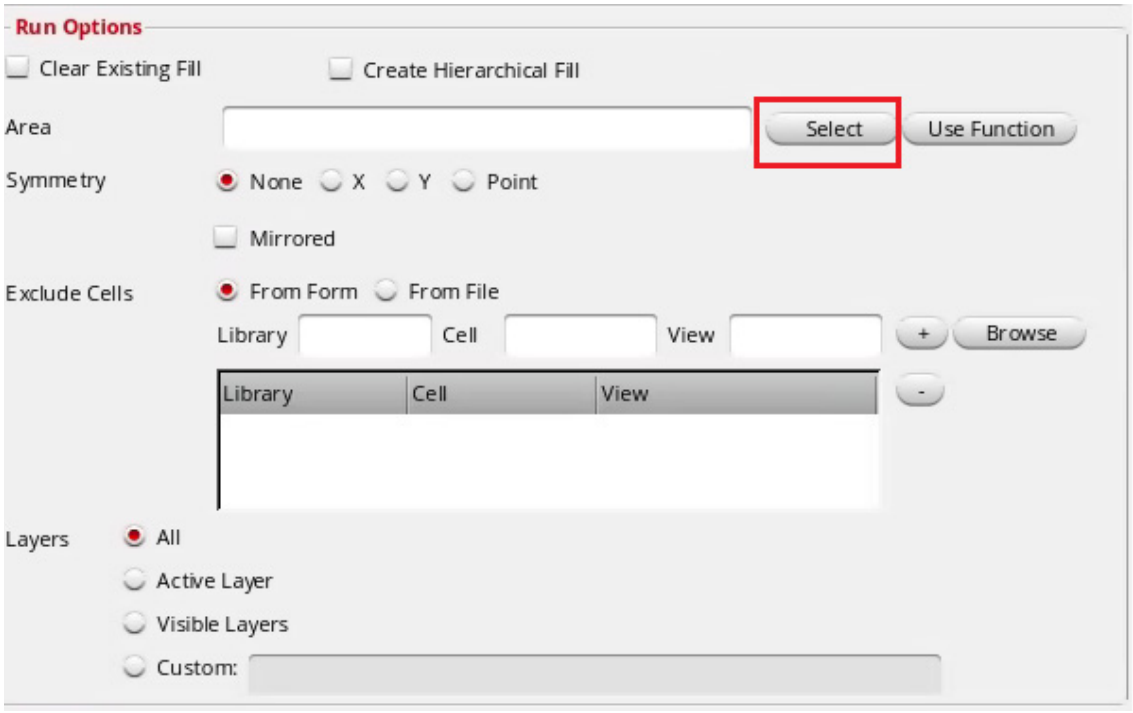


Running Pegasus Interactive SignOff Fill in Area

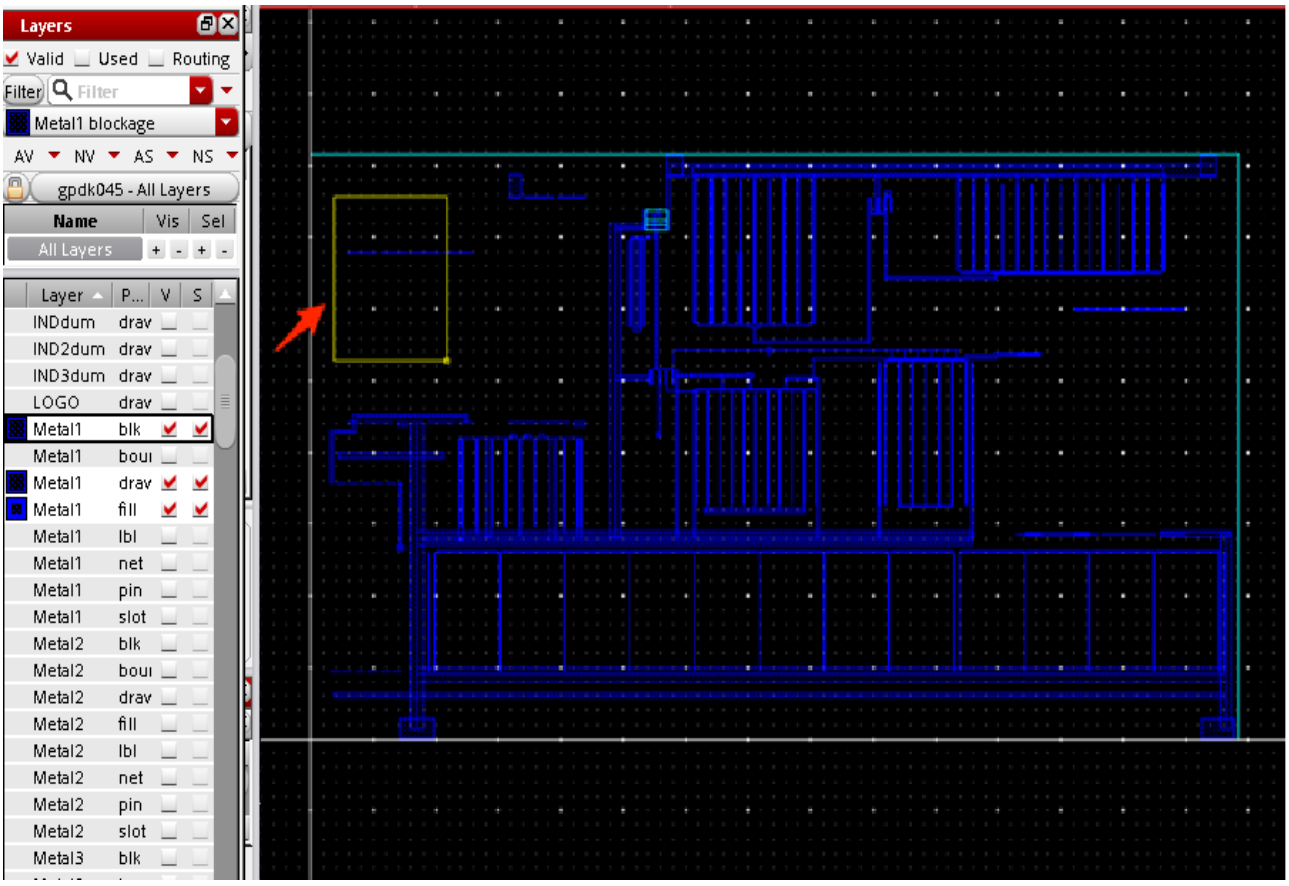
You can generate fills in a specific area and then add more fills incrementally in another area using Pegasus Interactive SignOff Fill.

Perform the following steps to generate fills incrementally:

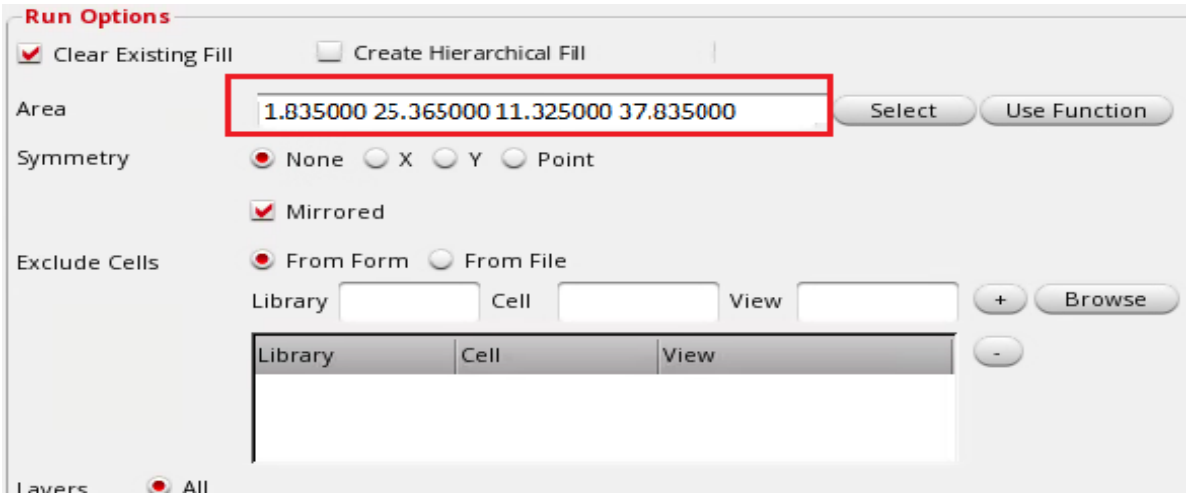
1. Open the *Pegasus Interactive Run Options* form and go to the *SignOff Fill* tab.
2. Make appropriate selection on layer visibility in the *Run Options* block. In the *Area* field, you can select the area by either clicking the *Select* button or the *Use Function* button.
    a. To make selection using the *Select* button, the *Select* button of the *Area* field.



b. On the layout window, create a boundary box as shown below.



The *Area* field is populated with coordinates of the boundary box.

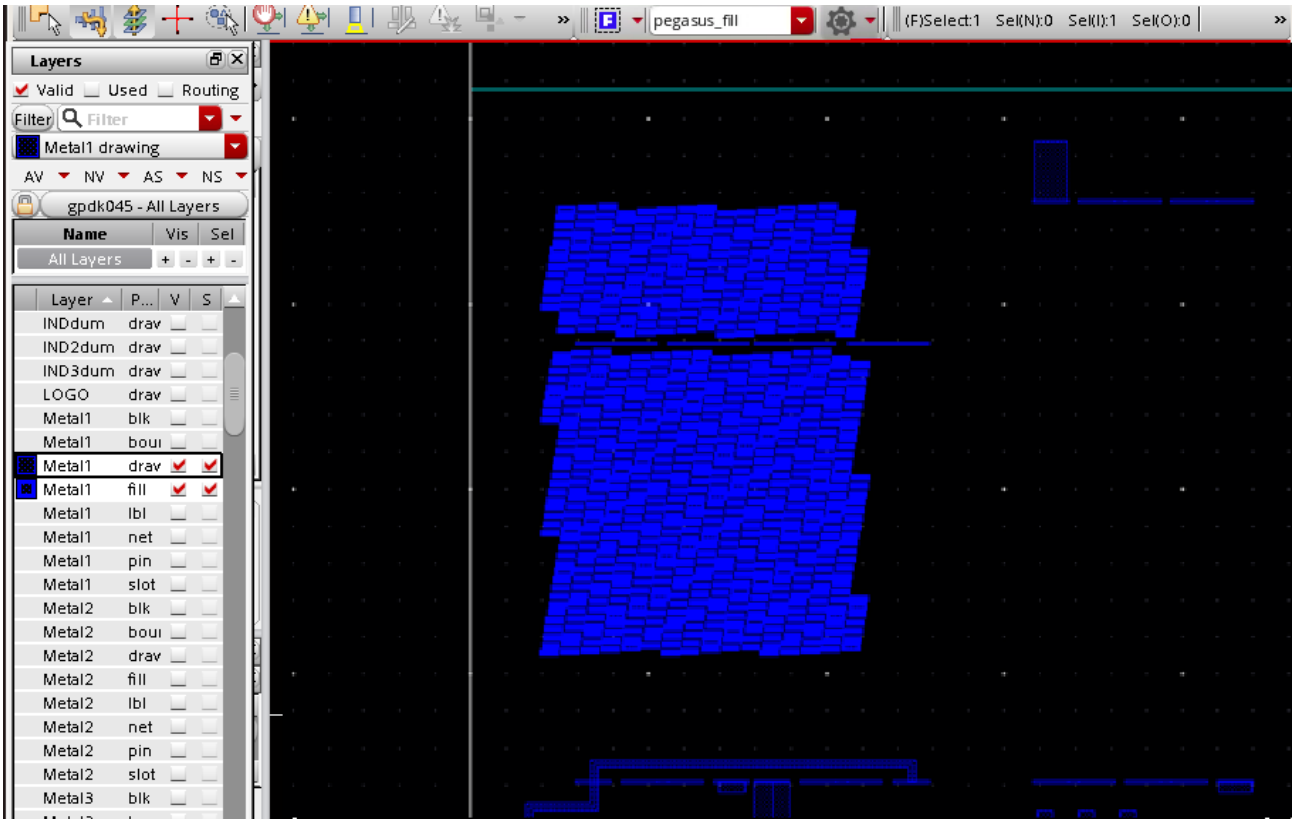


3. Similarly, you can use, the *Use Function* button to fill the *Area* field. You can define `pgssUserSelectedFillArea()` function to select area to fill. If you have defined this function then clicking on the *Use Function* button fills the area field with already defined coordinates. If you have not defined the function then an error message is displayed.

   Note: To view example of defining `pgssUserSelectedFillArea()` function, see Example of pgssUserSelectedFillArea Procedure on page 134.

4. Click *OK* to close the form.

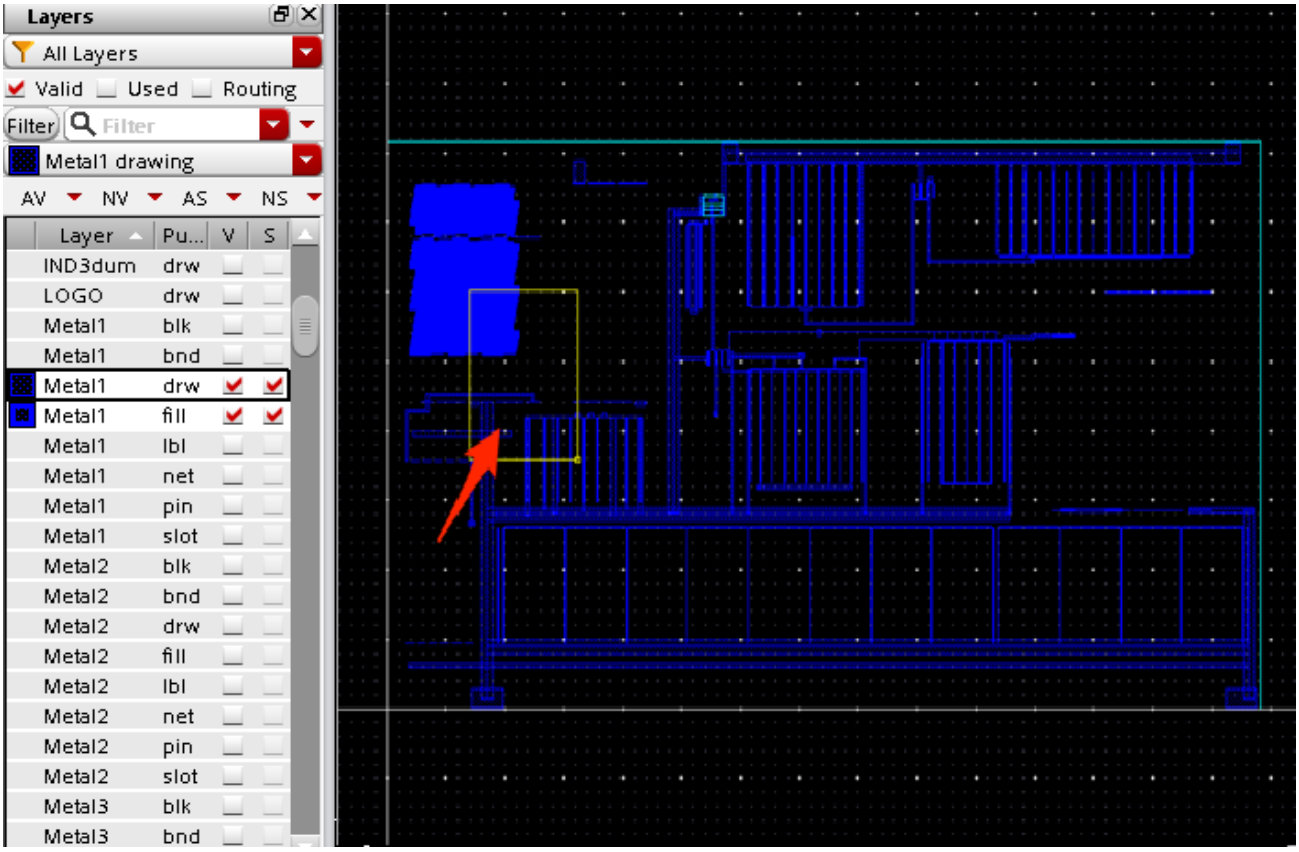
5. Run Pegasus Interactive SignOff Fill. The *Run Pegasus Interactive SignOff Fill* icon changes to show that you have selected *Area* mode.



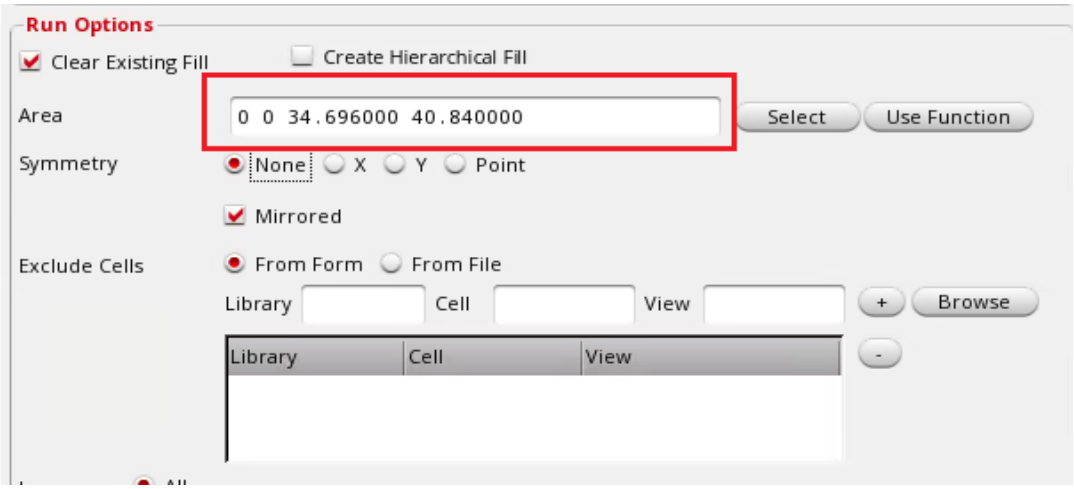
   Dummy fills are generated for the specified area. Dummy fills outside the specified area are not generated.

6. Open the *Pegasus Interactive Run Options* form and go to the *SignOff Fill* tab.

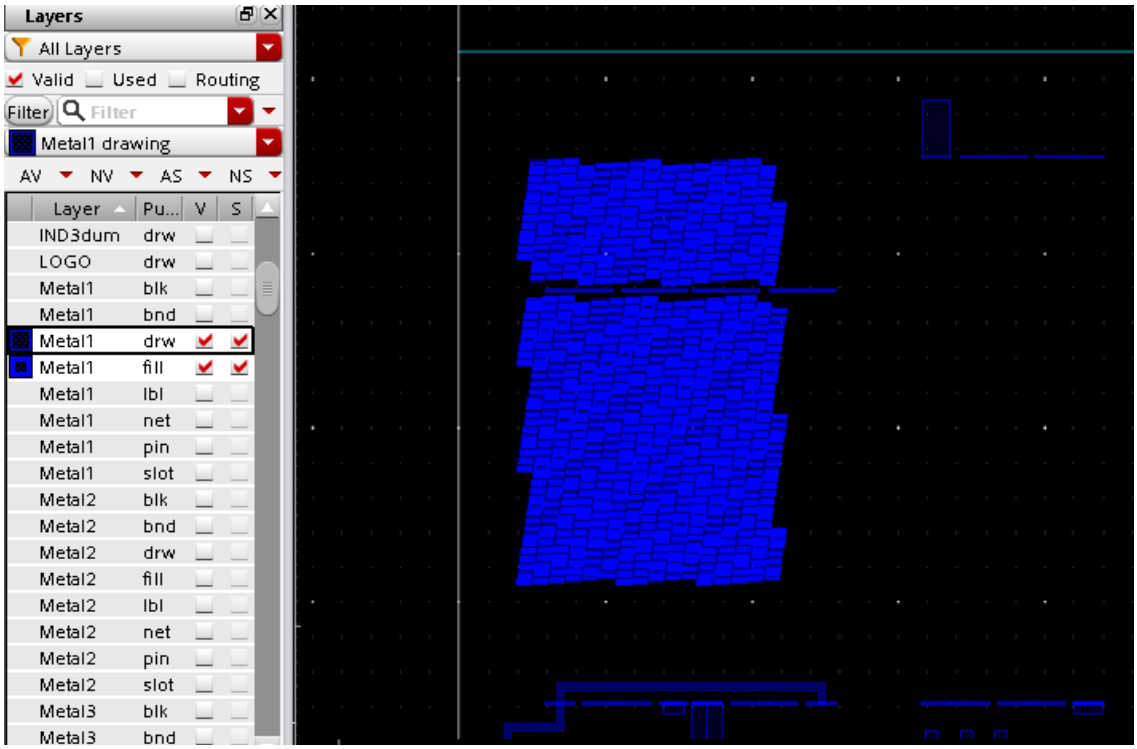
7. On the layout window create a boundary box as shown below. Note that the new box should partially overlap the existing fills.



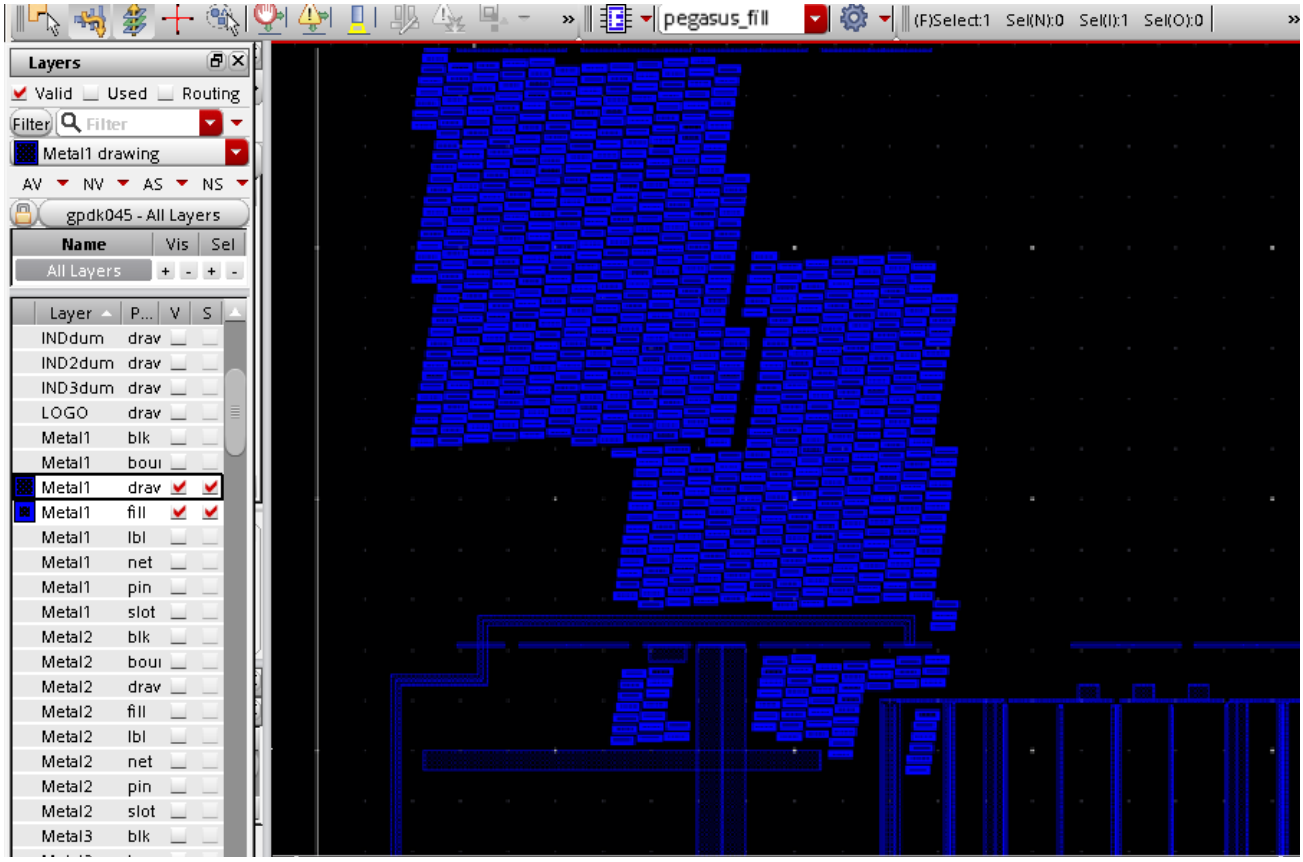
8. Once you do a second click to define the second point of the rectangle, the *Area* field will be automatically populated with coordinates of the boundary box:



9. Click *OK* to close the form and run Pegasus Interactive Signoff Fill.



Dummy fills are generated for the specified area incrementally to the already generated fills.



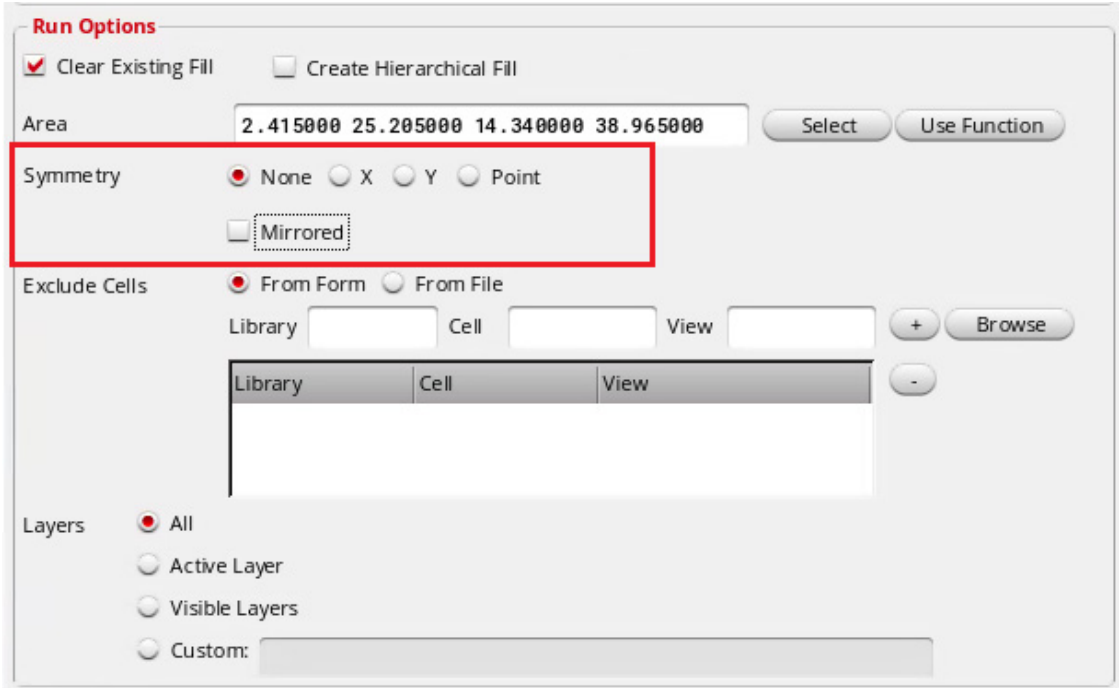
Related Topics

- Generating Fill Shapes Using Symmetry Value

Generating Fill Shapes Using Symmetry Value

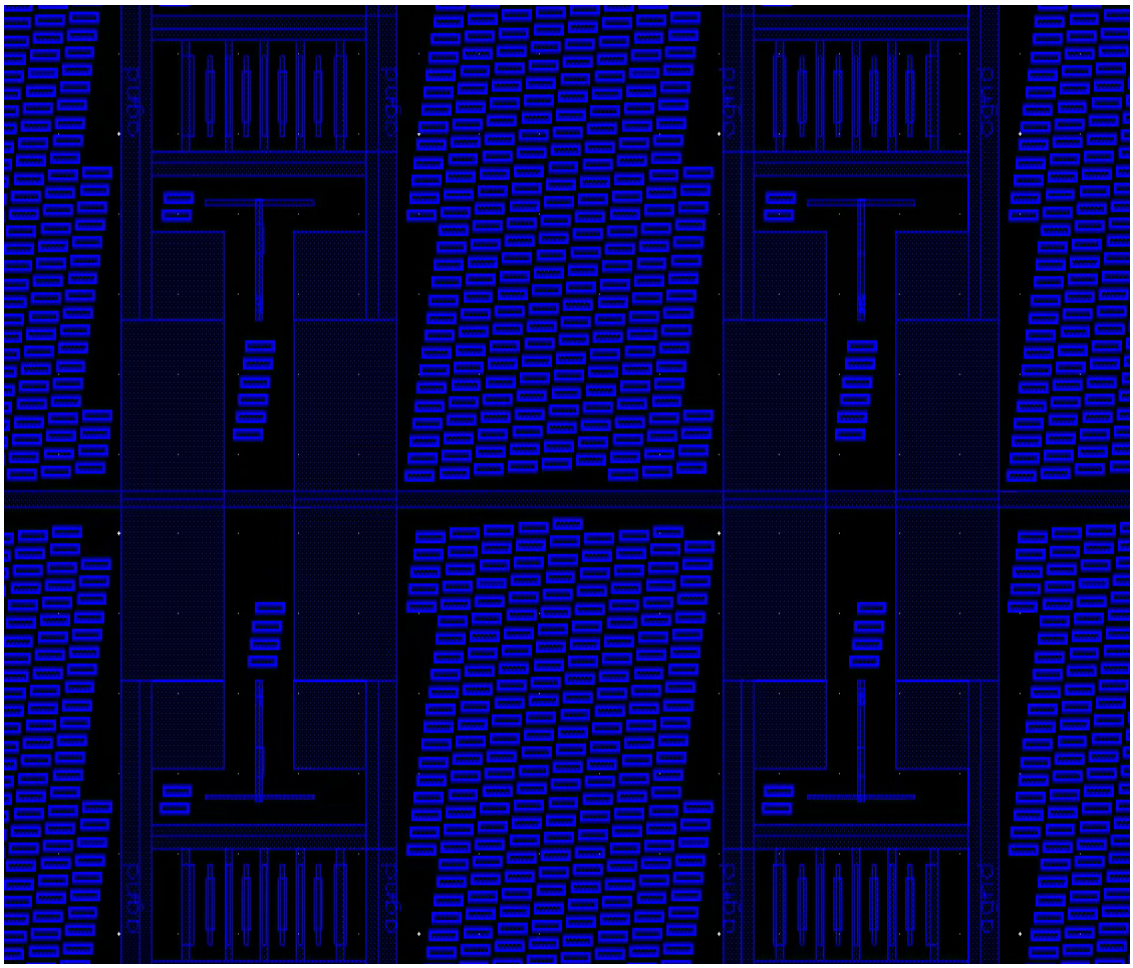
You can generate fill shapes in a user specified area based on specified symmetry value. You can specify the Symmetry options from the *Run Options* sections.

Note: For symmetry fill area to fill must be selected carefully around symmetrical structures. If you run symmetry fill on non-symmetrical structure, then there may be chances of presence of DRC violations or metal-fill overlap.



Note: To use any option other than *None*, the *Area* option must be selected. If the *Area* option is not selected and area is not defined, an error message is displayed.

When you select the *None* option, the results are displayed as a normal run:



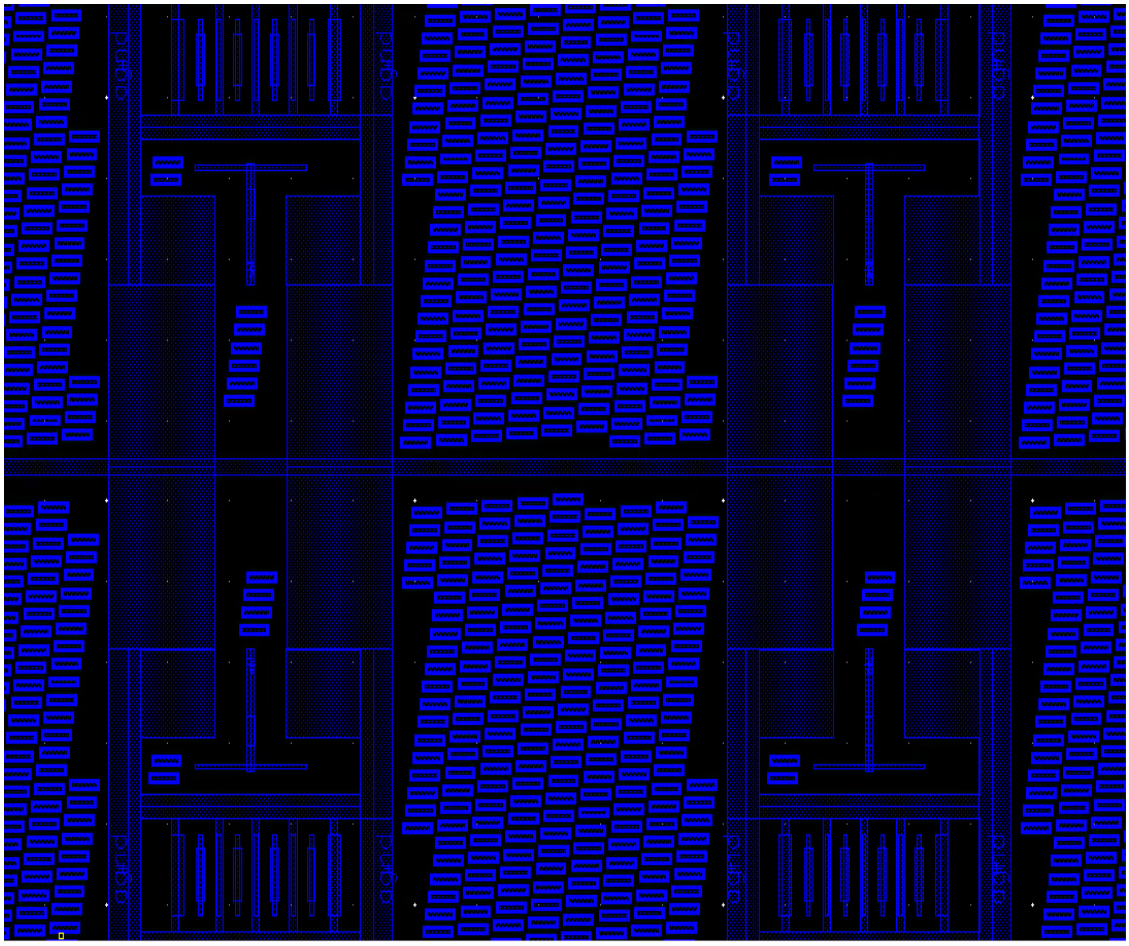
There can be following combinations:

- Option X, Non-mirrored
- Option X, mirrored
- Option Y, Non-mirrored
- Option Y, Mirrored
- Option Point, Non-mirrored
- Option Point, Mirrored

Examples of different symmetry fill options

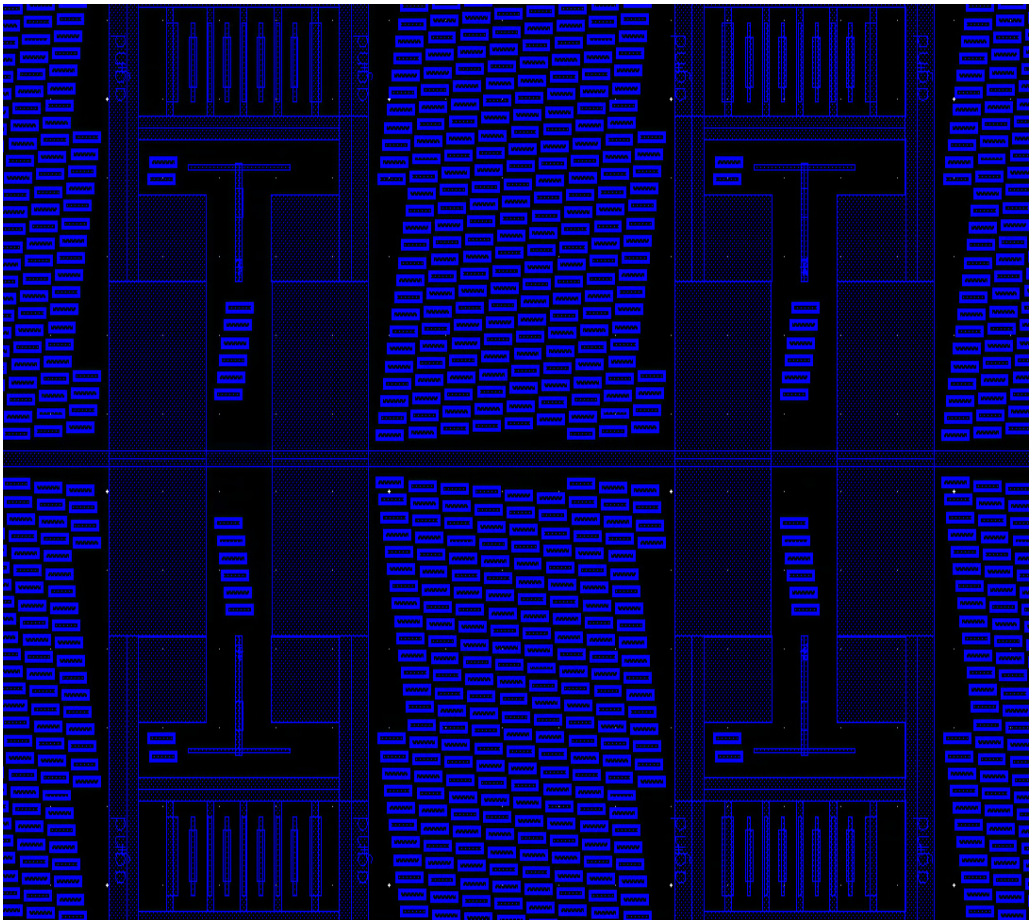
Example 1

You have selected the *X* radio button but unchecked the *Mirrored* check box. The result is shown as follows:



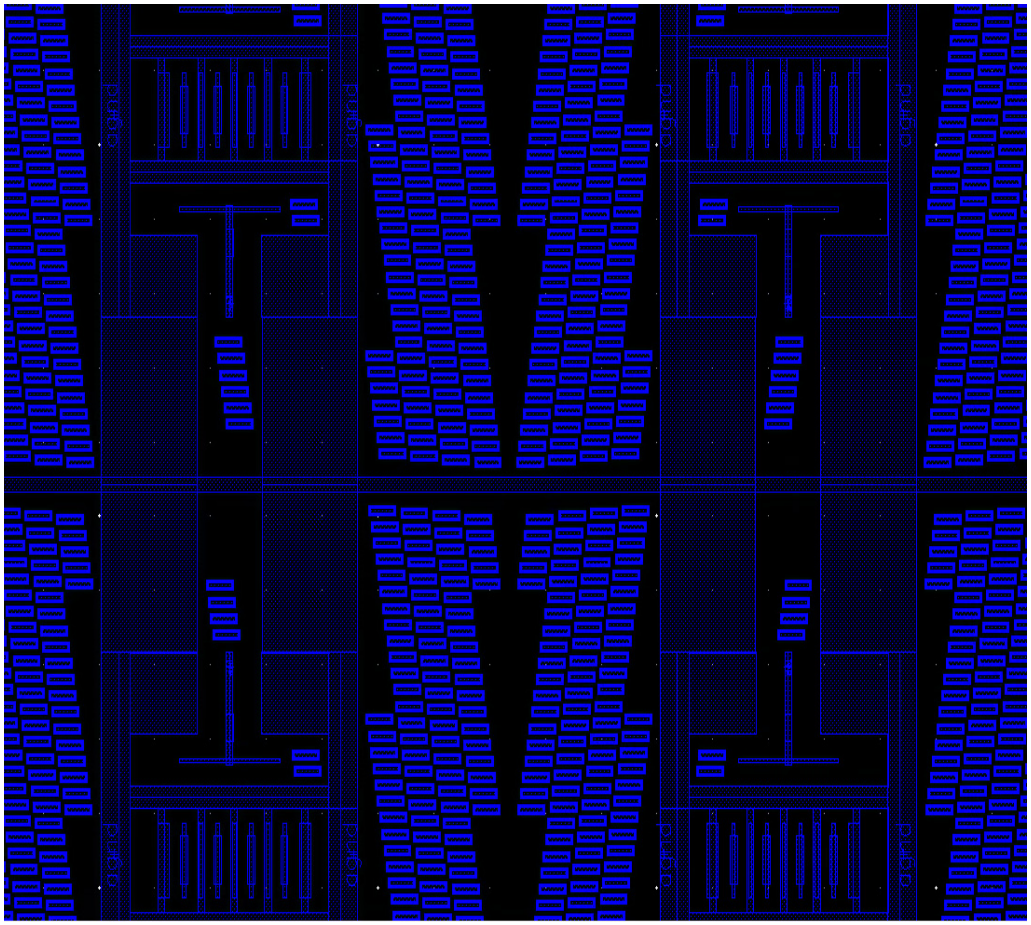
Example 2

You have selected *X* radio button and the *Mirrored* check box. The result is shown as follows:



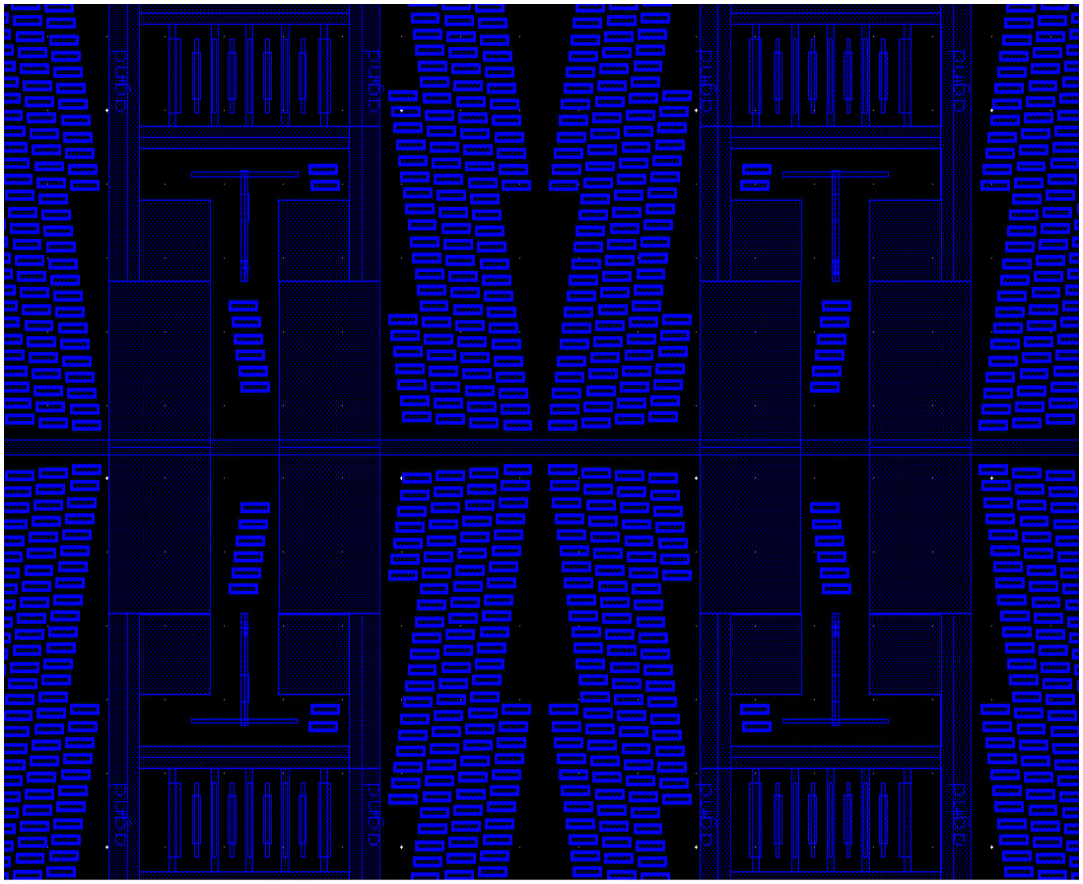
Example 3

You have selected *Y* radio button and the *Mirrored* check box. The result is shown as follows:



Example 4

You have selected *Point* radio button and the *Mirrored* check box. The result is shown as follows:



Example of pgssUserSelectedFillArea Procedure

An example of SKILL which creates area around selected instances:

```
procedure( pgssUserSelectedFillArea()
    let( (obj xmin ymin xmax ymax selectedList hSet)
        selectedList = geGetSelectedSet()

        if( selectedList then
            obj = car(selectedList)
            xmin = xCoord(car(obj~>bBox))
            ymin = yCoord(car(obj~>bBox))
            xmax = xCoord(cadr(obj~>bBox))
            ymax = yCoord(cadr(obj~>bBox))

            foreach(obj cdr(selectedList)
                xmin = min(xmin xCoord(car(obj~>bBox)))
                ymin = min(ymin yCoord(car(obj~>bBox)))
                xmax = max(xmax xCoord(cadr(obj~>bBox)))
                ymax = max(ymax yCoord(cadr(obj~>bBox)))
            )

            hSet = geCreateHilightSet(geGetEditCellView() list("y0" "drawing"))
            geAddHilightRectangle(hSet list(xmin:ymin xmax:ymax))
            geSetHilightSetHaloParameters(hSet "under" "fadeout" "thin" 25 nil)
            hSet~>enable = t
       )

       if( xmin then
           list(list(xmin ymin) list(xmax ymax))
       )
    )
)
```

7

Running Density Analysis

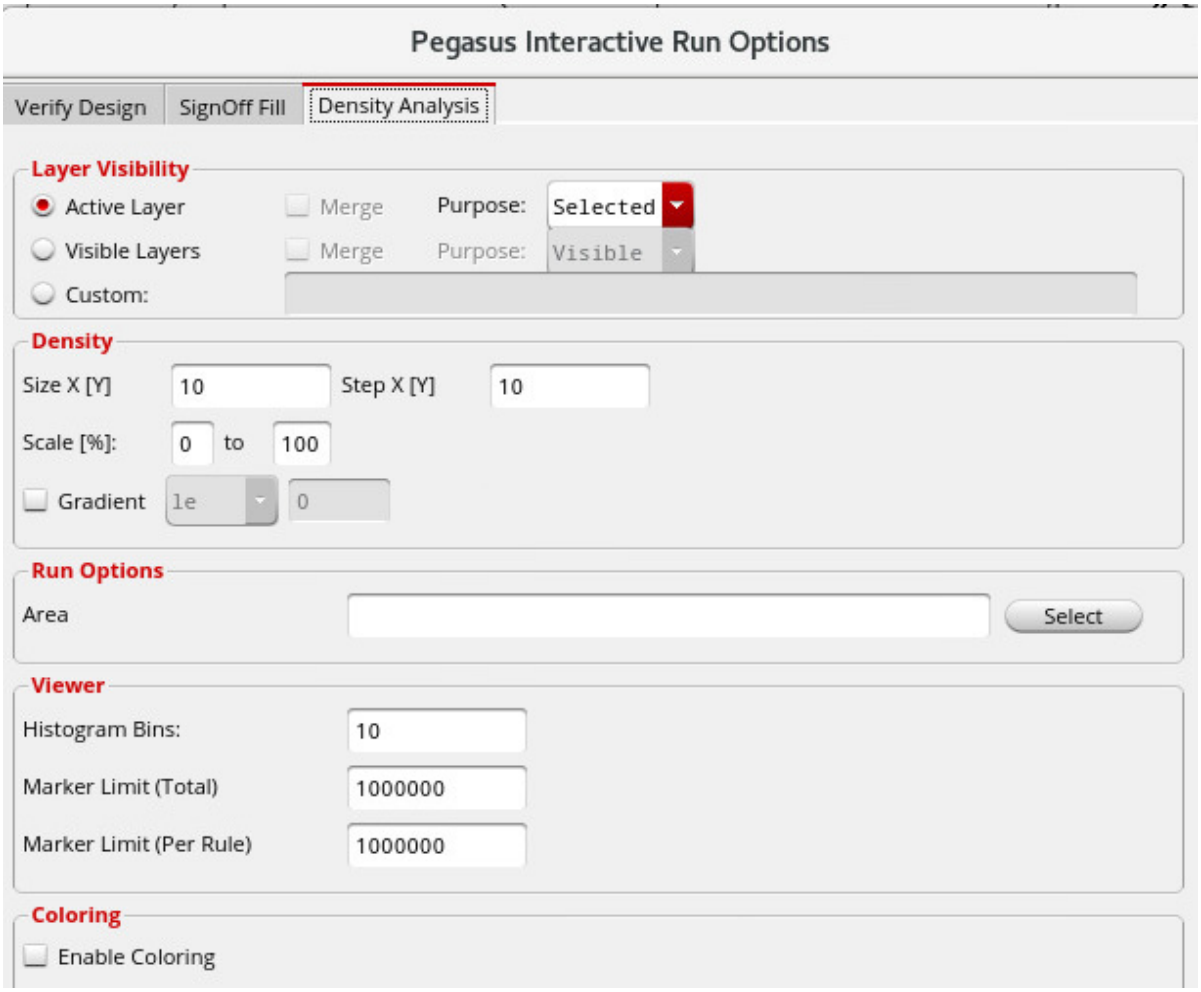
This chapter includes the process of running Density Analysis from Virtuoso and generating Histograms and Density heatmaps.

The content of this chapter is organized as follows:

Introduction to Pegasus Interactive Density Analysis

Density analysis generates density results from the *Pegasus Interactive* toolbar for the required layers. Rule deck is not required. Results are shown in the layout viewer as density heatmap and in Pegasus Results Viewer as histogram.

You can customize the results per your requirements using the options available in the *Density Analysis* tab of the *Pegasus Interactive Run Options* form:

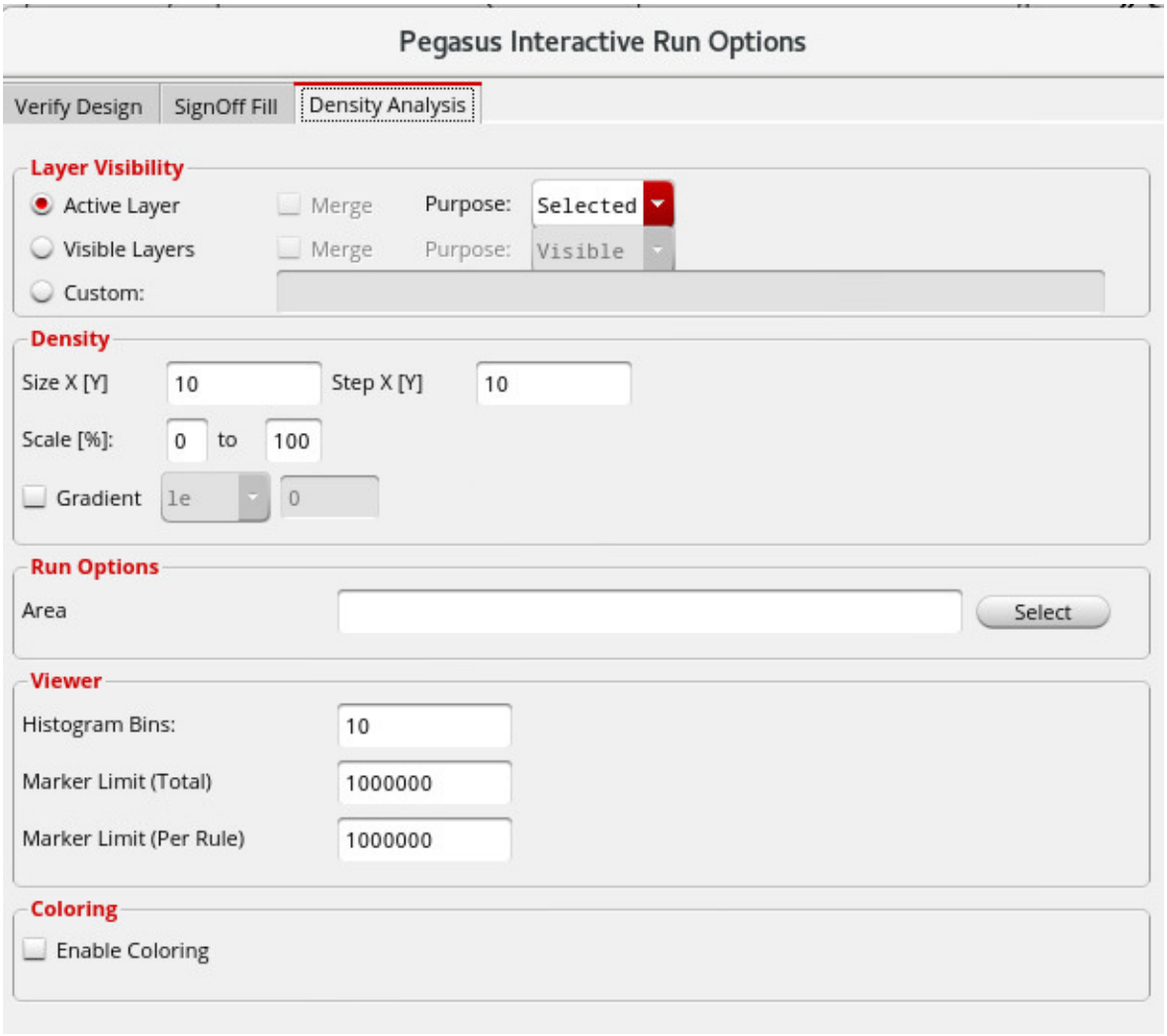


For detailed description of the fields of the *Density Analysis* tab, see Run Settings for Density Analysis on page 51.

Running Density Analysis

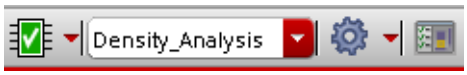
Perform the following steps to do density analysis:

1. Click the *Pegasus Interactive Run Options* icon on the *Pegasus Interactive* toolbar. The *Pegasus Interactive Run Options* form opens.
2. Select the *Density Analysis* tab.
3. Make the required selections in the *Density Analysis* form as per your requirement, such as for which layers you want to run density analysis, specifications for the histogram to be generated, and density setup.

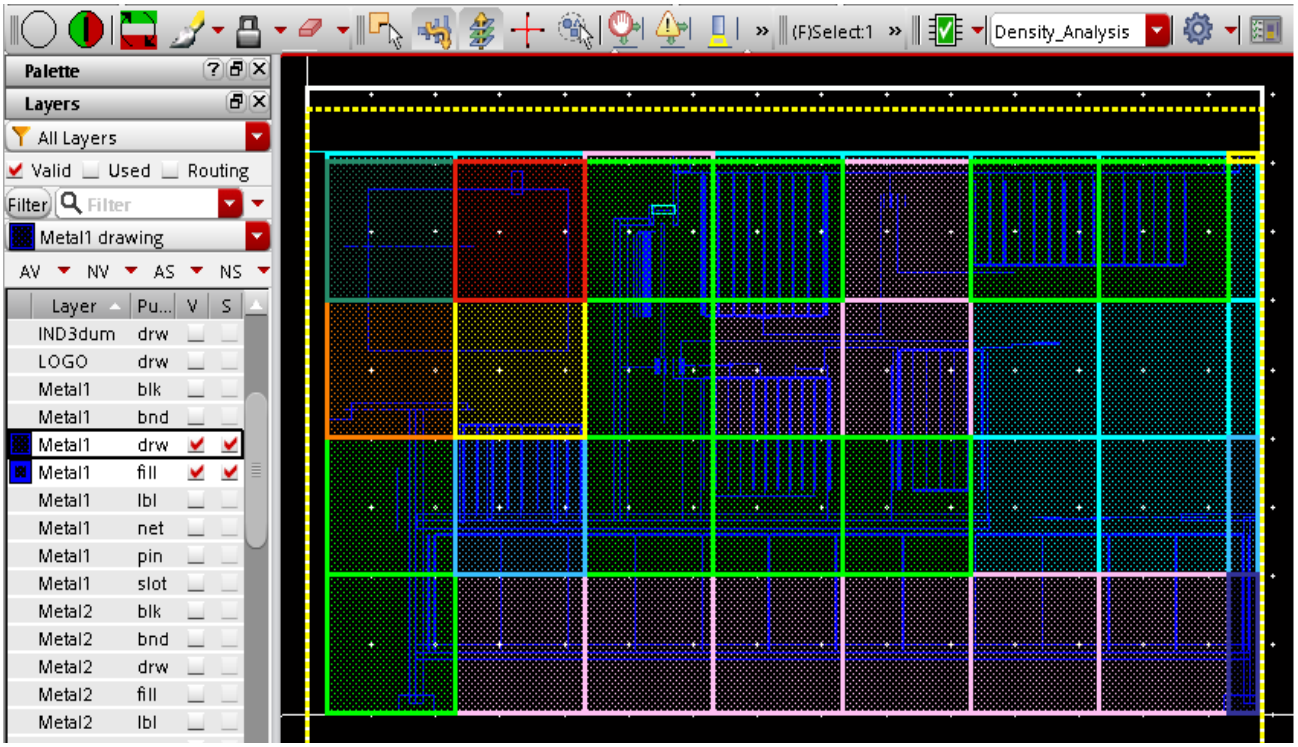


4. Click *OK* to close the *Pegasus Interactive Run Options* form.

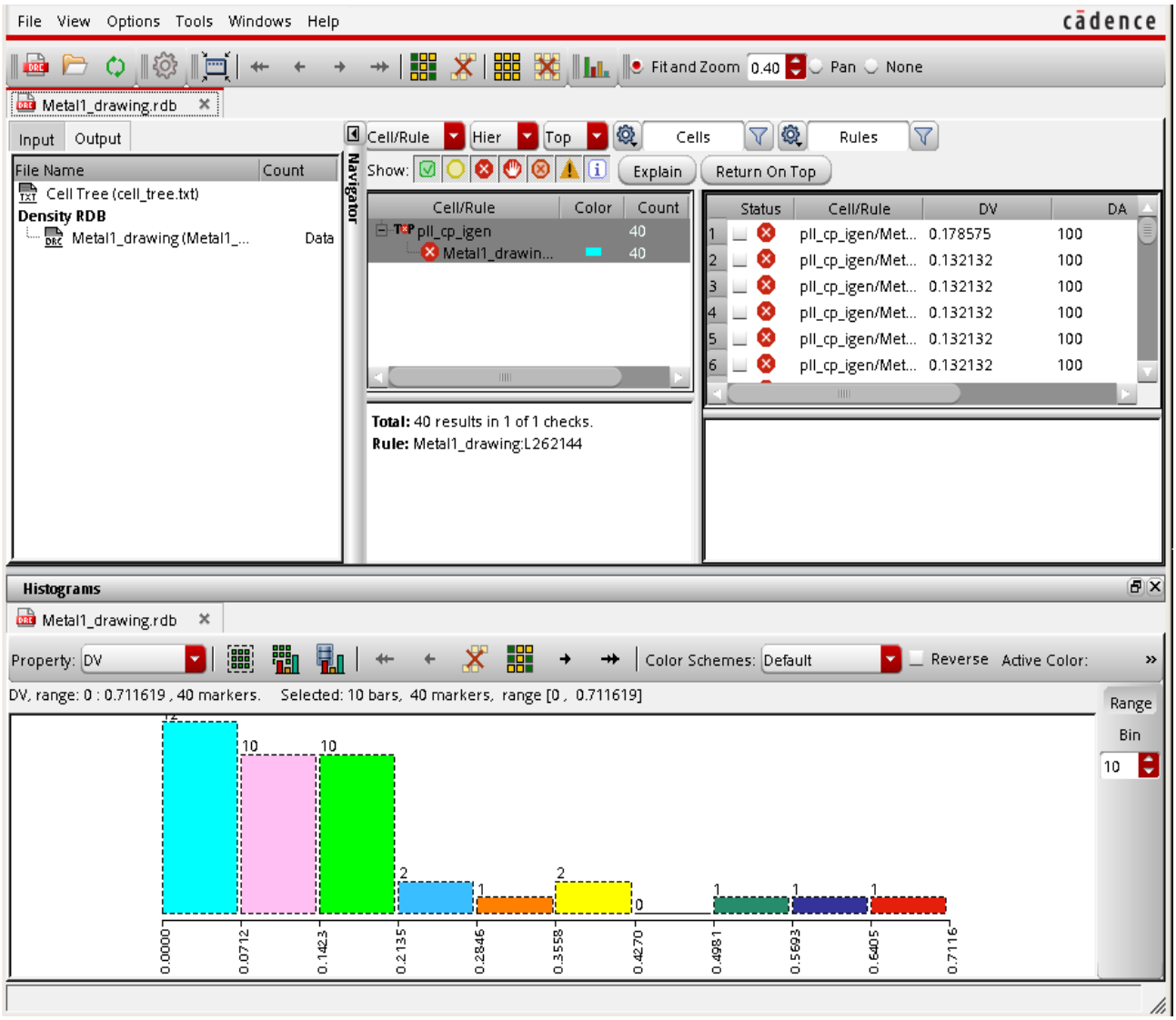
5. Select *Density Analysis* from the *Pegasus Interactive* toolbar.



6. Click the *Run Pegasus Interactive Verify Design* icon. On completion of the run, the density heatmap is shown on the layout canvas:



The Pegasus Results Viewer also opens showing the generated histogram:

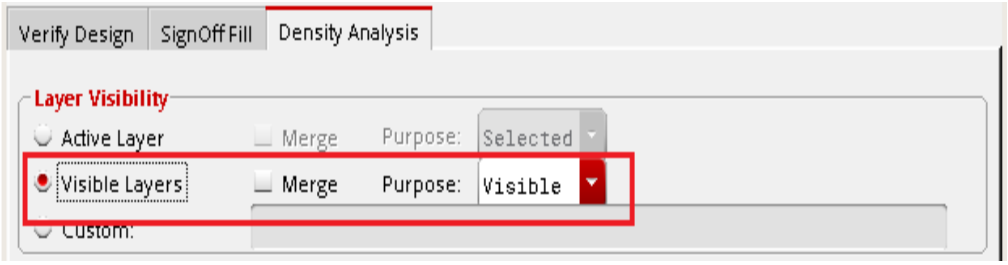


In the above steps, density analysis was run on the active layer. Similarly, you can run density analysis on the visible and custom-defined layers.

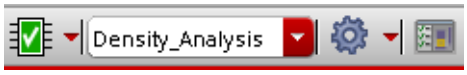
Running Density Analysis on Visible Layers

Perform the following steps to run density analysis for the visible layers:

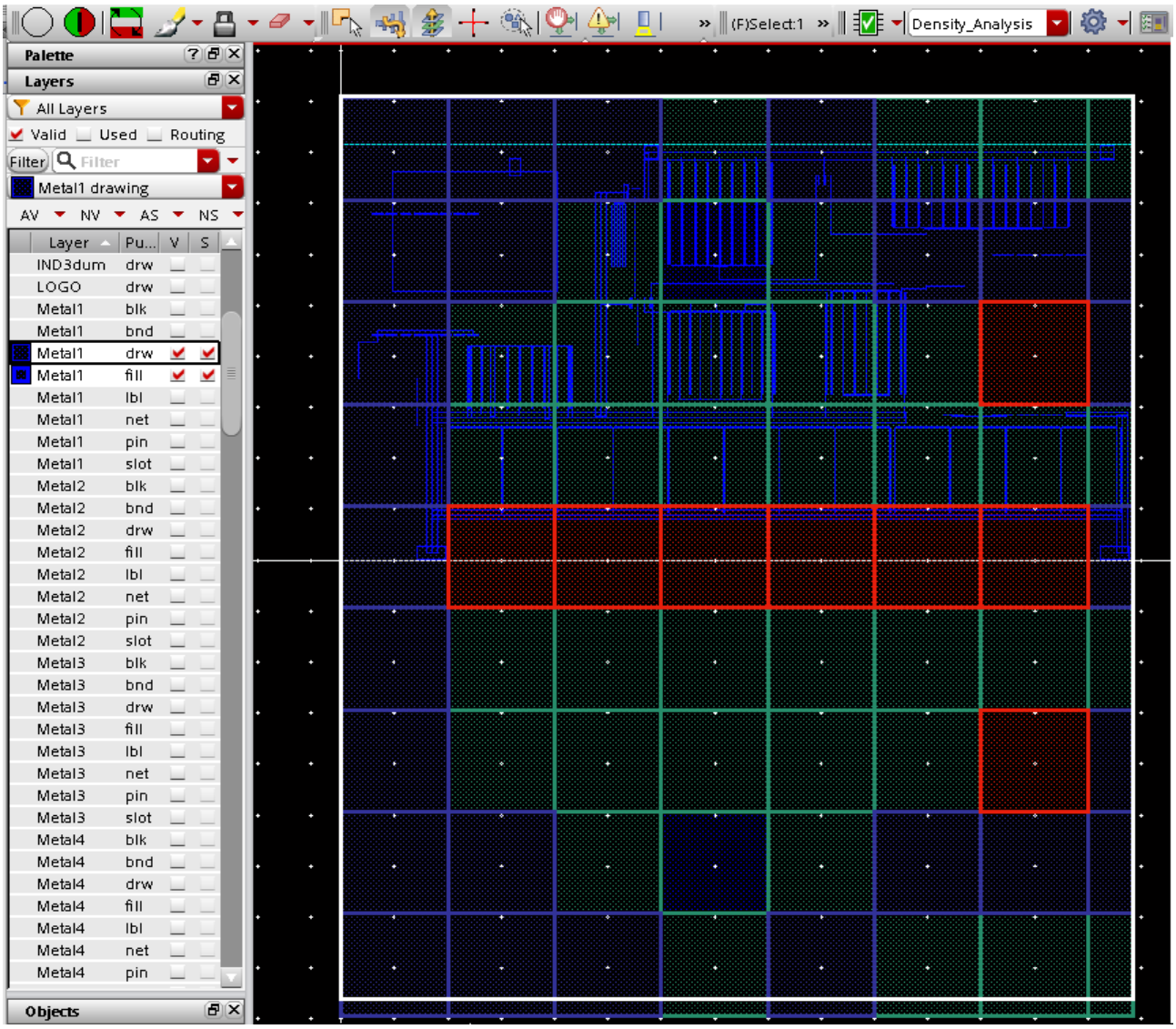
1. Set the visibility of the layers for which you want to run density analysis. In this example, two layers `metal1_drawing` and `metal1_fill` are set to visible.
2. Click the *Pegasus Interactive Run Options* icon on the *Pegasus Interactive* toolbar. The *Pegasus Interactive Run Options* form opens.
3. Select the *Density Analysis* tab.
4. Select the *Visible Layers* radio button in the *Layer Visibility* section.



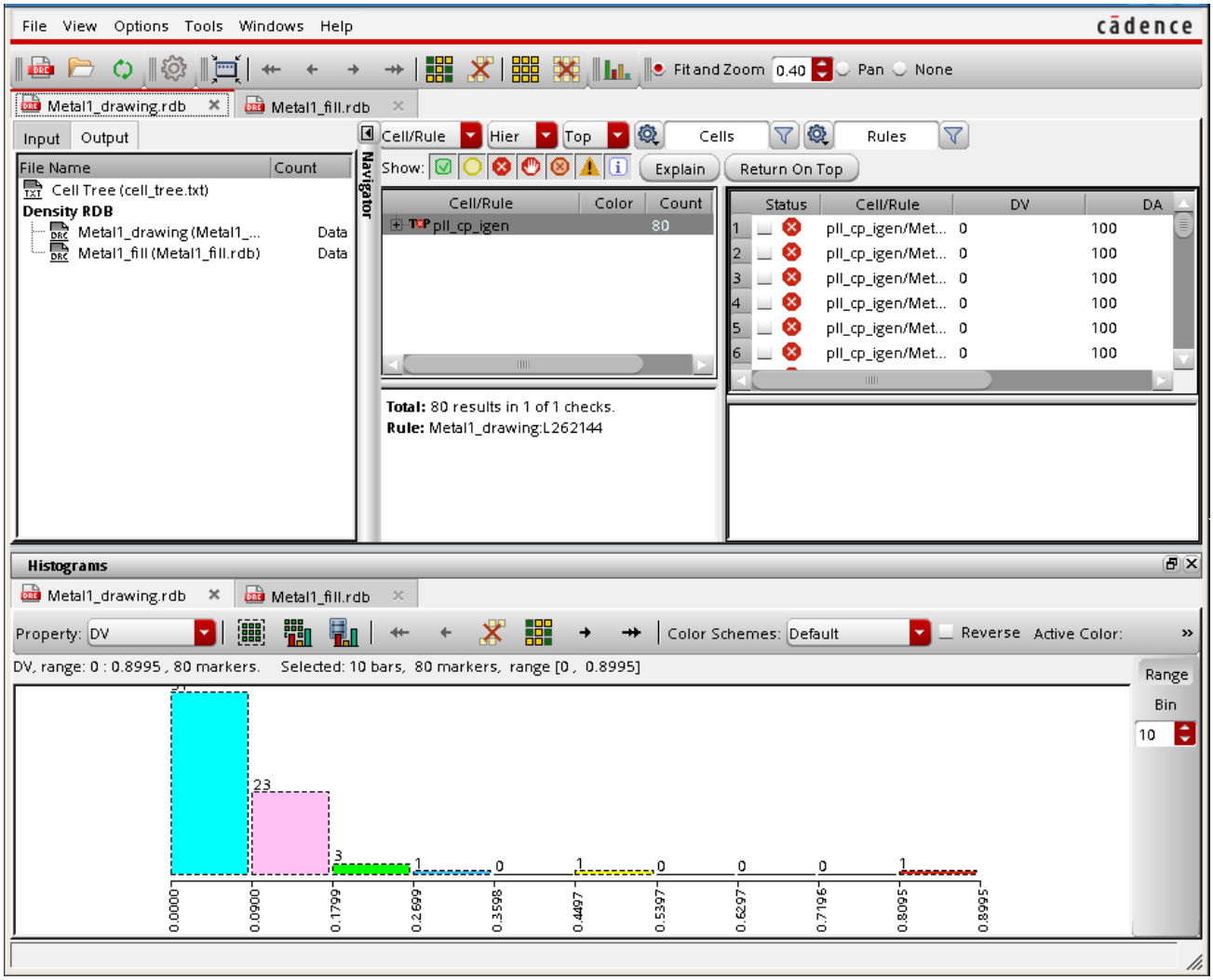
5. Click OK to close the *Pegasus Interactive Run Options* form.
6. Select *Density Analysis* from the *Pegasus Interactive* toolbar.

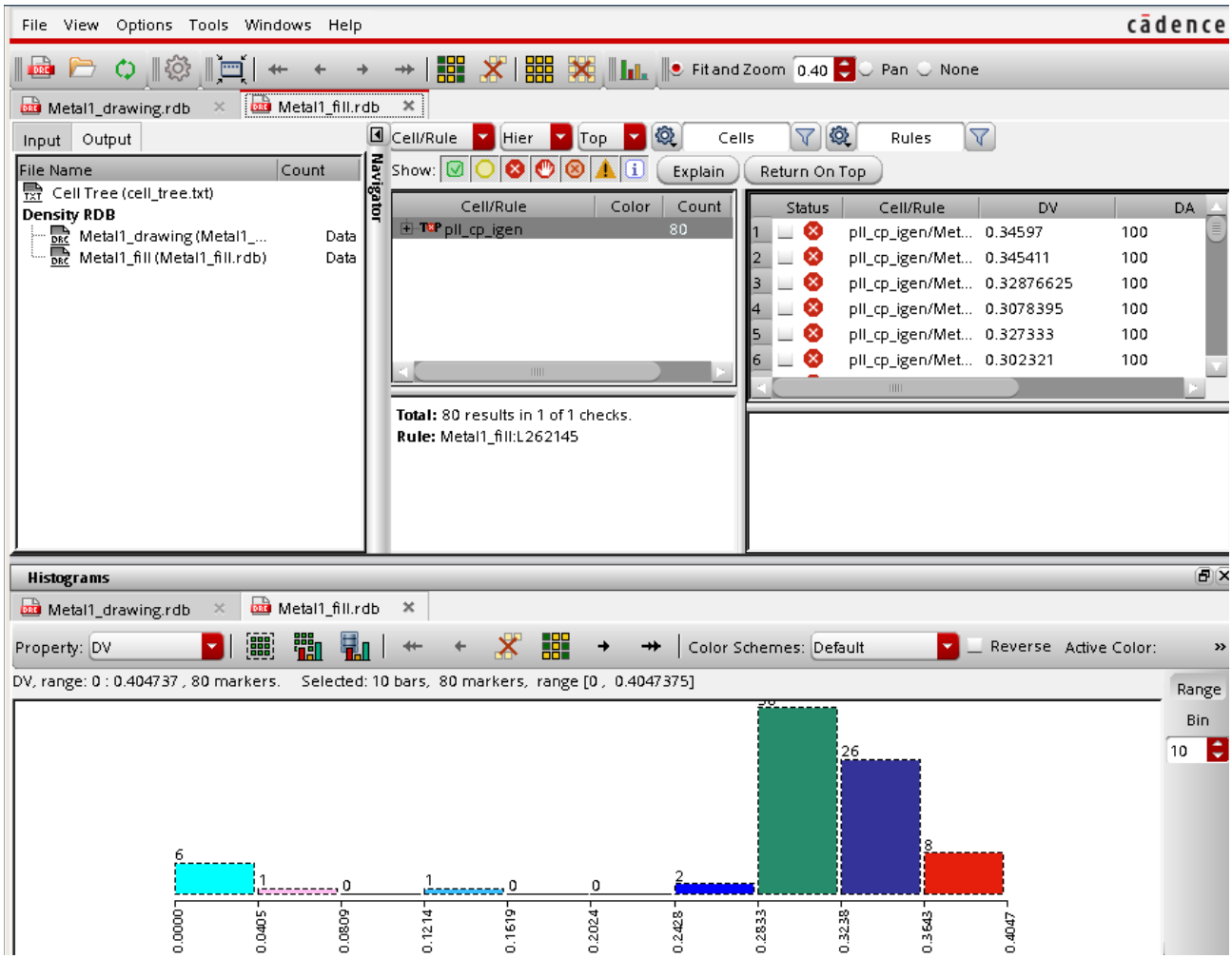


7. Click the *Run Pegasus Interactive Verify Design* icon. On completion of the run, the density heatmap is shown on the layout canvas:



The Pegasus Results Viewer also opens showing the generated histograms. The Pegasus Results Viewer shows two tabs for each layer showing two histograms:

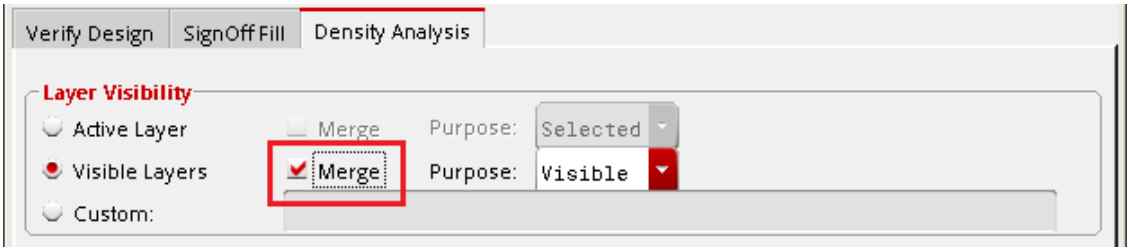




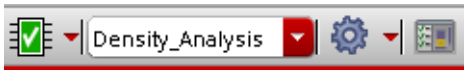
Merging Density Analysis Results

You can merge the run results of multiple layers using the *Merge* option in the *Density Analysis* tab. In this case, the tabs are merged into one with Pegasus Results Viewer and histogram show result under one tab only.

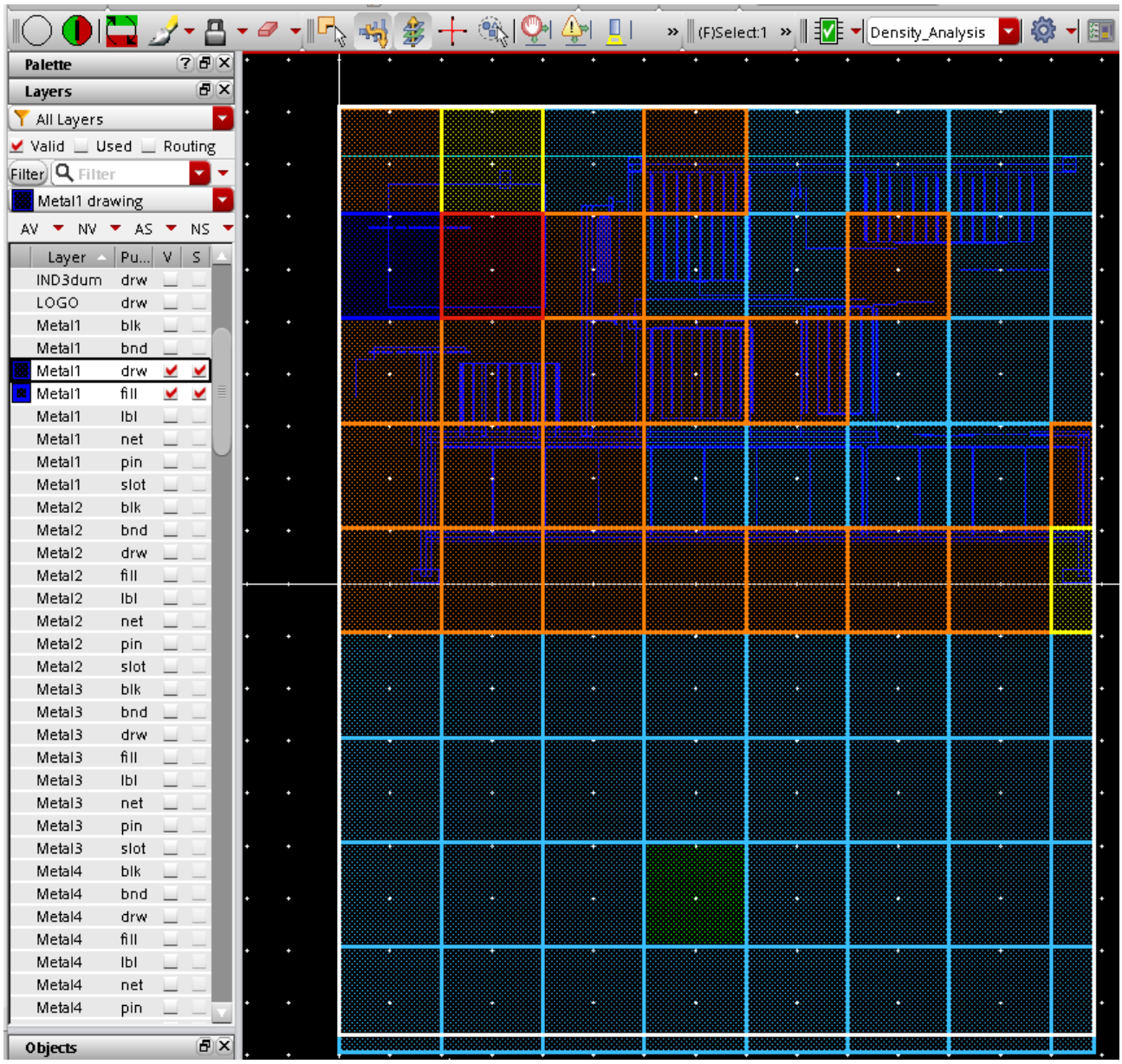
1. Set the visibility of the layers for which you want to run density analysis. In this example, two layers `metal1_drawing` and `metal1_fill` are set to visible.
2. Click the *Pegasus Interactive Run Options* icon on the *Pegasus Interactive* toolbar. The *Pegasus Interactive Run Options* form opens.
3. Select the *Density Analysis* tab.
4. Select the *Visible Layers* radio button in the *Layer Visibility* section and also select the *Merge* check box.



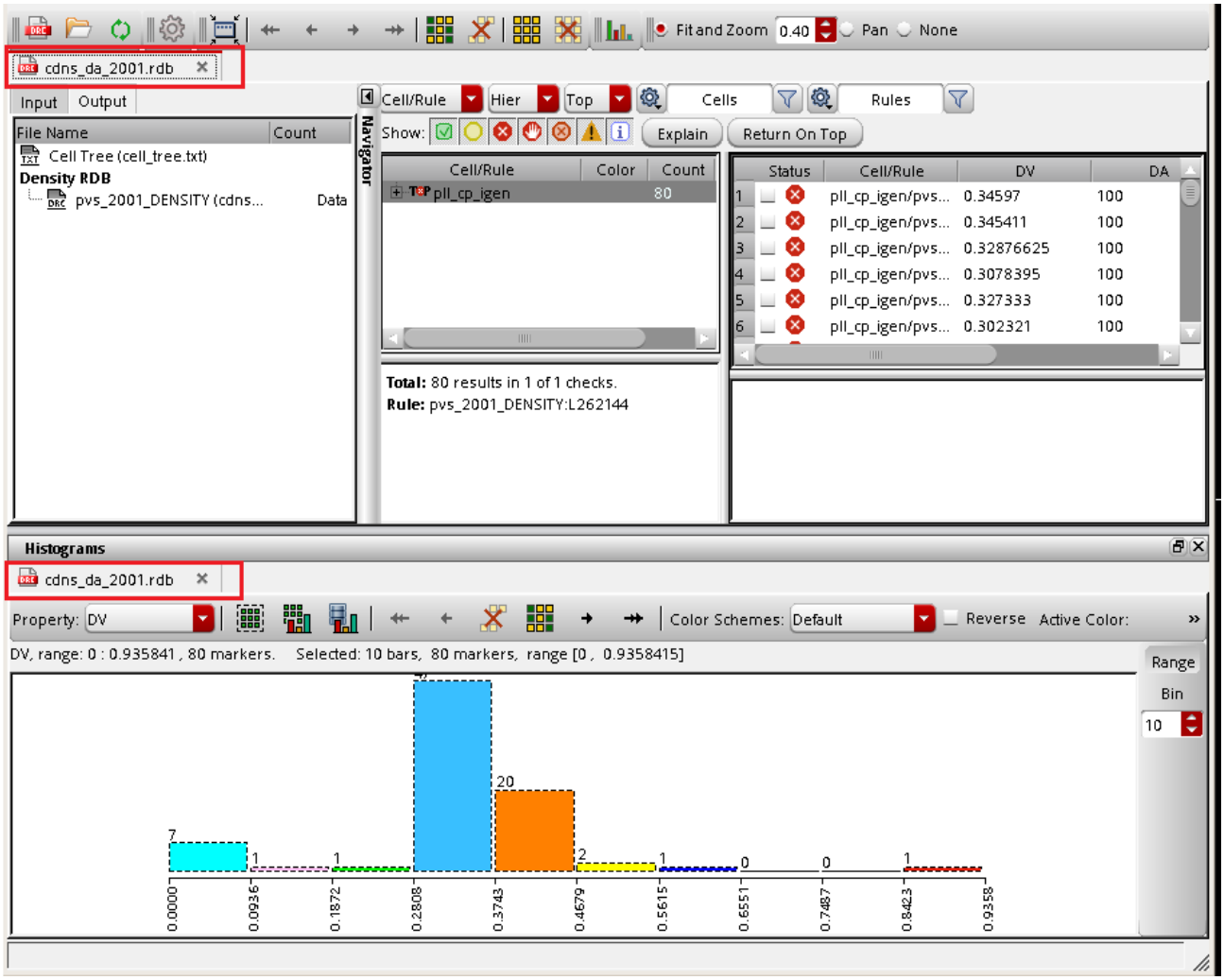
5. Click OK to close the *Pegasus Interactive Run Options* form.
6. Select *Density Analysis* from the *Pegasus Interactive* toolbar.



7. Click the *Run Pegasus Interactive Verify Design* icon. On completion of the run, the density heatmap is shown on the layout canvas:



The Pegasus Results Viewer opens showing the generated merged histogram:



8

Error Viewing

Pegasus Interactive supports two type of browsers: one is the Pegasus Results Viewer and second is Virtuoso native supported Annotation Browser. Pegasus Results Viewer is the default browser for Pegasus Interactive. This chapter describes the basic setup and customization of the browsers. For details of each browser usage, refer to:

- Pegasus Results Viewer: Cadence Pegasus Verification System User Guide
- Annotation Browser: Virtuoso Layout Suite XL User Guide

The content of this chapter is organized as follows:

Pegasus Results Browser

The Pegasus Results Browser allows you to view and analyze DRC, Antenna, Density, and ERC violations. For more details on Pegasus Results Browser usage and customization, refer to *Cadence Pegasus Verification System User Guide*.

Annotation Browser

Annotation Browser is the native Virtuoso browser for all Virtuoso tools reporting markers. Using this browser, you can run DRC on editable and read-only cellveiw. However, browser will not be invoked like Pegasus RV after each run. You have to invoke it through *Window > Assistants > Annotation Browser* in the layout. Once it is invoked, Pegasus Interactive violations are populated under the *DRC/DFM* tab.

Similar to Pegasus RV, you can customize its setting. Once the customization is complete, you can save the customization through RMB on the *Assistant > Settings > Save*. By default, it is saved under `$PWD/.cadence/dfII/ab/ab.init`.

For more details on Annotation Browser usage and customization, refer to *Virtuoso Layout Suite XL User Guide*.

9

Batch Commands

This chapter provides the syntax, descriptions, and examples for the batch utility that can be used in conjunction with the Pegasus™ Interactive editing feature.

pegasusGenSnapshot

Allows you to generate snapshot files based on inputs of snapshot name, directory path, layermap and rule deck. In addition, it allows you to define rule selections via a control file. The syntax within the control file is understood by Pegasus and can be used directly for Pegasus sign-off technology enablement. By default, this utility generates a preset file that can be loaded into *Pegasus Interactive Snapshot Creator*.

Usage:

```
pegasusGenSnapshot [-h | -help | -v | -version ]
```

To generate a snapshot via the command line:

```
pegasusGenSnapshot -name snapshotName -dir dirFilePath [output_arguments]
     -layerMap FilePath [-objectMap FilePath] [-control FilePath]
     [ -comment string ] rulefile1 [rulefile2 ... rulefileN]
```

To convert existing GUI preset file to snapshot via the command line:

```
pegasusGenSnapshot -name snapshotName -dir dirFilePath [output_arguments] -
     preset FilePath
```

Arguments

`-name` *`snapshotName`*	Mandatory keyword specifies snapshot name.
`-dir` *`dirFilePath`*	Mandatory keyword specifies the snapshot directory path (relative or absolute).
`-layerMap` *`FilePath`*	Mandatory keyword to specify absolute layermap file path that Pegasus Interactive is referenced to. If `PegasusInt_DefaultSNPLayerMap` is set, this option is not required. However, explicit declaration has priority over this shell variable. Ignored if `-preset` argument is specified.
`-objectMap` *`FilePath`*	Optional keyword to specify absolute object mapping file path that Pegasus Interactive is referenced to. Ignored if `-preset` argument is specified.
`-control` *`FilePath`*	Optional keyword to specify the control file containing Pegasus readable syntax for rule selection (`select_check` or `unselect_check`) or switches (`#define` or `#undefine`). Wildcard (* | ?) is supported for `select_check` / `unselect_check` commands. Ignored if `-preset` argument is specified.
`-preset` *`FileName`*	Input preset file.
`-comment` *`string`*	Optional keyword to specify a description comment for the snapshot. You can specify either single word or several words. In second case, double quotes are needed, such as `"description comment"`. For single word double quotes are optional.
`rule_file1` `[` *`… rule_fileN`*`]`	Mandatory keyword specifies absolute rule deck path. If `PegasusInt_DefaultSNPRuleDeck` is set, this option is not required. However, explicit declaration has priority over this shell variable. Ignored if `-preset` argument is specified.

Pegasus Interactive Snapshot Creator automatically detects layermap in the attached techlib and uses it during creating snapshot. In this case preset file, which is part of snapshot, has empty `layerMap` field.

```
LayerMapFile        ""
```

If you use preset file as input of `pegasusGenSnapshot` batch utility, you will notice that `pegasusGenSnapshot` fails with following error:

```
[ERROR] Pegasus UI: Run Data->Layer Map List [LayerMapFile]: Layer map is not
defined and not found in technology library.
```

To avoid this error you have to specify path to layer map file:

```
LayerMapFile       "<path to layer map file>"
```

Output Arguments

`-noGenPreset`	Optional keyword specifies that preset file for *Pegasus Interactive Snapshot Creator* should not be generated. Ignored if `-preset` argument is specified.
`-log` *`FilePath`*	Optional keyword prints run log to a designated file. If it is not specified, run log is stored in `default.snpGen.log` at the current working directory.
`-h,--help`	Prints help message and exits.
`-v, -V, -version, --version`	Prints Pegasus version and exits.

```
setenv PegasusInt_SnapshotsDirs $PWD/CAD_SNP_BatchDir
setenv PegasusInt_DefaultSNPRuleDeck $RULEFILE
setenv PegasusInt_DefaultSNPLayerMap $LAYERMAP

#pegasus interactive -- Basic without control file
pegasusGenSnapshot -name CDN20_Default -dir $PegasusInt_SnapshotsDirs -log ./cdn20Default.snpGen.log

#pegasus interactive -- switch on and off vias control file
pegasusGenSnapshot -name CDN20_Colorless -dir $PegasusInt_SnapshotsDirs -control .../TECHDIR/Pegasus/batchSnpGenSetup/Colorless.ctl  -log ./CDN20_Colorless.snpGen.log

#pegasus interactive --- select | unselect rules via wildcard through  control file
pegasusGenSnapshot -name ColorDenOnly -dir $PegasusInt_SnapshotsDirs -control .../TECHDIR/Pegasus/batchSnpGenSetup/CDNOnly.ctl -log ./CDNOnly.snpGen.log
```

CDNonly.ctl

```
text_depth -primary;
virtual_connect -colon no;
virtual_connect -semicolon yes;
virtual_connect -noname;
virtual_connect -report no;
virtual_connect -depth -primary;
df2key enableColoring yes;

unselect_check ?.Dpt.? ?.SMS.? G_FEOL; ?DM0*
unselect_check *Width* *Spacing* *Area?;
unselect_check G_ColorStateChk G_DPC_Rules;
```

Colorless.ctl

```
text_depth -primary;
virtual_connect -colon no;
virtual_connect -semicolon yes;
virtual_connect -noname;
virtual_connect -report no;
virtual_connect -depth -primary;
df2key enableColoring yes;
#UNDEFINE COLOR_STATE_CHECK;
#UNDEFINE FDRY1Rules;
#DEFINE BEOL_CHECK;
#DEFINE FEOL_CHECK;
```

10

Environment Variables

This chapter provides information on Pegasus Interactive shell and CDS environmental variables that allow you to customize tool's default options and behaviors.

The content of this chapter is organized as follows:

Pegasus Interactive Shell Environmental Variables

These shell variables are packaged within Pegasus installation. Their availability is based on the Pegasus version. Four types of shell environmental variables are used by Pegasus Interactive:

- Table 1 lists shell variables that allow you to customize the display and content of fields available in Pegasus Interactive Run Option.
- Table 2 lists shell variables that allow you to declare default technology setup for Pegasus Interactive Snapshot Creator GUI.
- Table 3 lists shell variables that allow you to customize the default Pegasus Interactive toolbar icons display.
- Table 4 lists shell variables that enable Pegasus Interactive in debug environment.

Table 1: Pegasus Interactive Run Option GUI Customization

Name	Description	Type	Default
`PegasusInt_MarkerBrowser`	Specifies the default error browser as `PegasusRV` or `AnnotationBrowser`.	string	`PegasusRV`
`PegasusInt_EnableSNPCreation`	When set to `YES`, the *Create Snapshot* button will be displayed under the *Snapshots* section.	boolean	`NO`
`PegasusInt_EnableSNPManagement`	When set to `YES`, the *Manage* button will be displayed under the *Snapshots* section.	boolean	`NO`
`PegasusInt_Maintenance`	When set to `YES`, the *Upgrade Snapshot* button is displayed under the *Snapshots* section.	boolean	`NO`

Name	Description	Type	Default
`PegasusInt_ExcludeCellFile`	Specifies the default path of exclude cell file.	string	`NIL`
`PegasusInt_Data`	Specifies global path to Pegasus Interactive data. Needed in case there are include statements in the rule deck and you want to make snapshot portable between different locations.	string	`NIL`

Table 2: Technology Enablement

Name	Description	Type	Default
`PegasusInt_SnapshotsDirs`	Specifies global snapshot directories as *`<path1>:<path2>:…`*. If declared directory does not exist, Pegasus Interactive creates the directory. If it is not specified, snapshot directory is pointed to `<$PWD>/.cadence/pegasus/<$userid>/presets/pegasusint/snapshot`. Path input can be relative path, absolute path or shell variables.	string	`NIL`

Name	Description	Type	Default
PegasusInt_DefaultSNPConfigFile	Specifies the default configurator file that is loaded under the *Configurator* tab in the *Pegasus Interactive Snapshot Creator* form.	string	NIL
PegasusInt_DefaultSNPLayerMap	Specifies the default layermap file that is pre-populated in *Layer Map List* in the *Input* tab of the *Pegasus Interactive Snapshot Creator* form.	string	NIL
PegasusInt_DefaultSNPObjectMap	Specifies the default objectmap file that is pre-populated in *Object Name Table* in the *Input* tab of the *Pegasus Interactive Snapshot Creator* form.	string	NIL
PegasusInt_DefaultSNPRuleDeck	Specifies the default rule deck{s} that is pre-loaded in *Rules* under the *Tech& Rules* tab in the *Pegasus Interactive Snapshot Creator* form.	string	NIL

Table 3: Pegasus Interactive Toolbar Customization

Name	Description	Type	Default
PegasusInt_Info	When set to NO, *Snapshot Info inquiry* triggers are not populated as *Pegasus Interactive Run Option* icon drop-down items.	boolean	YES
PegasusInt_InitSnapshot	Specifies the default active snapshot populated in the *Snapshot* combo field. If it is not found, first item from the available snapshot list is set to active snapshot.	string	NIL
PegasusInt_InitArea	Specifies the default Verify-Design Area checking as CELL (Whole CellView), VIEWPORT (Visible Area) or AREA (Changed Area).	string	CELL

Table 4: Pegasus Interactive Debug Utility

Name	Description	Type	Default
PegasusInt_Dir	Specifies final run results directory that should be stored to. If it is not specified, all the run results are stored under current working directory.	string	$PWD/ PEGASUS_INT_RUNDIR_VD

Virtuoso CDS Environmental Variables

Virtuoso environment variables, also called CDS environmental variables, are used in Pegasus Interactive. Their availability is Virtuoso version dependent. These CDS environmental variables are packaged as part of the "layout" tool.

- Table 5 lists of variables to control error markers counts
- Table 6 lists of variables to customize Pegasus Interactive verification setting

Table 5: Error Markers Filtering

Name	Description	Type	Default
`drdBatchVioLimit`	Specifies maximum total number of markers reported by Pegasus Interactive per run.	int	`5000`
`drdVioLimitPerRule`	Specifies maximum number of markers per rules reported by Pegasus Interactive.	int	`5000`

Table 6: Pegasus Interactive Verification Setting

Name	Description	Type	Default
`drdPegasusIntFunctionalPresets`	Specifies the default dynamic rule filtering to any of following: ■ *Default* ■ *Any Visible* ■ *Any Visible nodensity* ■ *Any Visible noconnect* ■ *Any Visible noconnect nodensity* ■ *Ignore Visible nodensity* ■ *Ignore Visible noconnect* ■ *Ignore Visible noconnect nodensity* **Note:** If you are using ICADV123/IC617 ISR21 or newer then `drdPegasusIntFunctionalPresets` environment variable should be used. If you are using ICADV123/IC617 ISR20 or older then `drdPVSFunctionalPresets` environment variable should be used.	string	`Default`

Name	Description	Type	Default
`drdPegasusIntErrorDisplayMaster`	By default, markers are generated and displayed per instances. In Pegasus RV, markers are grouped per cell name. When `drdPegasusIntErrorDisplayMaster` is set to `nil`, markers are displayed on all cells found in the layout. **Note:** If you are using ICADV123/IC617 ISR21 or newer then `drdPegasusIntErrorDisplayMaster` environment variable should be used. If you are using ICADV123/IC617 ISR20 or older then `drdPVSErrorDisplayMaster` environment variable should be used.	boolean	`T`

A

Appendix A - Preset to Snapshot Batch Migration

To convert several preset files generated by Pegasus Interactive in older version to its snapshot equivalents, you can run `pegasusGenSnapshot` with `-preset` argument.

Example:

```
pegasusGenSnapshot -name Default -dir . -log default.log -preset
Default.OldVersion.pegasusint.preset
```

`pegasusGenSnapshot` will perform conversion of batch preset file `Default.OldVersion.pegasusint.preset` to Default snapshot.

Since the Pegasus Interactive Snapshot Creator form can be invoked outside of Virtuoso, you as an alternative to `pegasusGenSnapshot` can invoke it either in graphic or non-graphic mode to perform snapshot creation.

To simply invoke the Pegasus Interactive Snapshot Creator form, run the following command in your terminal:

```
pegasusgui -pegasusintSnapshot
```

Or invoke this form with a specific preset file loaded:

```
pegasusgui-pegasusintSnapshot -preset <inputPresetFilePath>
```

To convert several old preset files generated by Pegasus Interactive to its snapshot equivalents, you can run `pegasusgui` command in non-graphic mode to perform a batch preset file to snapshot conversion. Before doing a batch preset to snapshot run, following steps needs to be performed:

- Declare *`SnapshotName`* inside of the preset file
- Declare final snapshot directory via `PegasusInt_SnapshotsDirs`

Then, you can run batch preset to snapshot conversion as following:

```
pegasusgui -pegasusintSnapshot -preset <PresetPath> -any_flow -nograph
```

Following example shows batch snapshot creation using three presets created using older snapshot versions:

Step1: Decalare SnapshotName in all of preset files:

```
% grep SnapshotName ./PRESETS/*preset
./PRESET/ColorlessRules.preset: SnapshotName         ColorlessRules"
./PRESET/FDRY1ColorChk_Mx.preset: SnapshotName       "FDRY1ColorChk_Mx_1"
./PRESET/Mx_DecompRules.preset: SnapshotName         "Mx_DecompRules"
```

Step2: Execute the run script contains:

- Destined snapshot directory declared through `PegasusInt_SnapshotsDirs`
- `pegasusgui -pegasusintSnapshot -nograph` execution commands

```
% cat runBatchPrep2Snp
setenv PegasusInt_SnapshotsDirs $PWD/CAD_PRS2SNP_BatchDir
pegasusgui -pegasusintSnapshot -preset ./PRESETS/ColorlessRules.preset -
any_flow -nograph
pegasusgui -pegasusintSnapshot -preset ./PRESETS/FDRY1ColorChk_Mx.preset -
any_flow -nograph
pegasusgui -pegasusintSnapshot -preset ./PRESETS/Mx_DecompRules.preset -
any_flow -nograph
```