

Virtuoso IPVS User Guide

**Product Version 20.11
March 2021**

© 2012–2021 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

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

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence’s trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor

Contents

<u>Preface</u>	7
<u>How This Manual is Organized</u>	7
<u>Software Requirements</u>	8
<u>Licenses Requirements and Behaviors</u>	8
<u>Requirements</u>	9
<u>License Handling while Running VIPVS</u>	10
<u>Related Documents for Virtuoso IPVS</u>	11
<u>Installation, Environment, and Infrastructure</u>	11
<u>Technology Information</u>	11
<u>Virtuoso Tools</u>	12
<u>Additional Learning Resources</u>	12
<u>Typographic and Syntax Conventions</u>	12
<u>1</u>	
<u>Getting Started</u>	15
<u>Prerequisites</u>	16
<u>Using Virtuoso IPVS for the First Time</u>	17
<u>2</u>	
<u>Virtuoso IPVS Toolbar</u>	23
<u>Overview</u>	24
<u>Toolbar Icons</u>	26
<u>Verify-Design</u>	26
<u>Default Active Snapshot</u>	26
<u>Run Option</u>	27
<u>Error Browsing Utility</u>	28
<u>Virtuoso IPVS Run Options</u>	29
<u>Run Settings for Verify Design</u>	30
<u>Run Settings for SignOff Fill</u>	49
<u>Run Settings for Density Analysis</u>	53

Virtuoso IPVS User Guide

<u>Buttons at the Bottom of the Form</u>	56
--	----

3

<u>All about Design Rules: Snapshot</u>	59
<u>Snapshot Definition and Files</u>	60
<u>Snapshot Directory Types and Customizations</u>	62
<u>Quick Start: Create Snapshot</u>	63
<u>First Time</u>	64
<u>From Preset</u>	66
<u>From Current Active Snapshot Setting</u>	67
<u>Virtuoso IPVS Snapshot Creator Form</u>	69
<u>Snapshot Content Customization Field</u>	69
<u>Snapshot Output Field</u>	86
<u>Manage Snapshots</u>	88
<u>Snapshots Inquiry</u>	89
<u>Snapshots Management</u>	93
<u>Snapshots Upgrade</u>	96
<u>Portability of Snapshots</u>	99
<u>Checking that Snapshot is Generated by Up-to-Date DK Version</u>	100

4

<u>Invoke Design Rule Checking</u>	103
<u>Overview: Run Behavior</u>	104
<u>Run Results</u>	104
<u>Violation Markers Handling</u>	104
<u>Verify Design Mode</u>	105
<u>Current CellView</u>	105
<u>Visible Area</u>	106
<u>Changed Area</u>	107

5

<u>SignOff Fill: Snapshot</u>	111
<u>Virtuoso IPVS SignOff Fill Overview</u>	112
<u>Traditional Fill Flow</u>	112

Virtuoso IPVS User Guide

<u>Virtuoso IPVS SignOff Fill Use Model</u>	113
<u>Virtuoso IPVS SignOff Fill features</u>	113
<u>Technology Setup</u>	114
<u>Creating PVS Fill Snapshot</u>	115

6

<u>Running SignOff Fill</u>	119
<u>Running VIPVS SignOff Fill</u>	120
<u>Running VIPVS SignOff Fill with Blockages</u>	122
<u>Running VIPVS SignOff Fill in Area</u>	124

7

<u>Running Density Analysis</u>	131
<u>Introduction to VIPVS Density Analysis</u>	132
<u>Running Density Analysis</u>	133
<u>Running Density Analysis on Visible Layers</u>	136
<u>Merging Density Analysis Results</u>	140

8

<u>Error Viewing</u>	145
<u>DRC Design Environment</u>	146
<u>Quick Start: PV Color Scheme Setup in Virtuoso</u>	146
<u>Customization</u>	146
<u>Annotation Browser</u>	147
<u>PVS Results Browser</u>	148

9

<u>Batch Commands</u>	149
<u>IpvGenSnapshot</u>	149

10

Environment Variables 153

VIPVS Shell Environmental Variables 154

Virtuoso CDS Environmental Variables 158

A

Appendix A - Preset to Snapshot Batch Migration 161

Preface

This document describes how to verify a layout against physical design rules using the Virtuoso® Integrated Physical Verification System Option for Virtuoso Layout Suite (Virtuoso IPVS). Using Virtuoso IPVS, you can capture and validate design intent and ensure design convergence by performing in-design DRC checks on the layout immediately after editing the layout. These checks are performed using the PVS rule deck, which is a signoff rule deck. It allows you to analyze density for your design and generate metal fill in OA database.

The intent of this manual is to provide:

- In depth knowledge of setting Virtuoso IPVS
- Details of its verification capability
- Information of customize its rule sets

How This Manual is Organized

The chapters are organized as following:

- Getting Started: Specifies prerequisites for using VIPVS including design rules preparations and error browser setup.
- Virtuoso IPVS 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 Virtuoso IPVS to perform DRC on a layout in both of operate modes: Post-Edit and Verify-Design.
- Error Viewing: Describes how to use different browser utility provided by Virtuoso Layout Suite and PVS.
- Batch Commands: Describes batch commands used for snapshot creation
- Environment Variables: Describes Virtuoso IPVS environment variables.

Software Requirements

In order to run Virtuoso IPVS, both PVS and Virtuoso Layout Suite software need to be available. Its availability is based on the following versions of PVS and Virtuoso:

- Virtuoso L/XL/GXL:
 - IC616 ISR12 or newer
 - IC617 or newer
 - ICADV121 ISR14 or newer
 - ICADV122 or newer
- PVS:
 - 15.20 or newer.
- Snapshots:
 - Version 13.

Note: VIPVS 15.22 supports both snapshots versions 7 and 8. If your snapshot is older than v7 then it will not be shown in the VIPVS toolbar. To make it available you have to upgrade it. If snapshot is generated with PVS15.17 or older then to make it work properly snapshot needs to be re-generated with current PVS version. See [Snapshots Upgrade](#) on page 96 for details.

Note: When running Virtuoso IPVS on the Redhat Linux platform, RHEL 5.x or newer version is required. Virtuoso IPVS is not supported on earlier versions of Redhat Linux, such as RHEL 4.x.

Licenses Requirements and Behaviors

This section describes license requirements as well as license checkout and release behaviors.

Virtuoso IPVS User Guide

Preface

Requirements

Basic Licenses:

Name	Number	Description
Virtuoso_DRC_Opt	96400	Virtuoso Integrated Physical Verification System Option for Virtuoso Layout Suite license.
Phys_Ver_Sys_Results_Mgr	96240	Cadence® Physical Verification System Results Manager.

Advanced Licenses:

Name	Number	Description
Virtuoso_IPVS_Adv_Anal_Opt	96310	Virtuoso® Integrated Physical Verification System Advanced Analysis Option.
Phys_Ver_Sys_Results_Viewer	96360	Cadence® Physical Verification System Results Viewer

To run VIPVS, you must first have a snapshot. A snapshot is a collection of reference files that VIPVS refers during DRC. For details of these files, refer to [All about Design Rules: Snapshot](#). Licenses requirements are documented within the snapshot. VIPVS uses this information to check out required licenses.

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 Virtuoso IPVS only determinates license requirement based on the content of rule deck. However, the license validity check for Virtuoso_DRC_Opt is executed.

Following figure illustrates `Mx_Only_ExcludeDecomp` snapshot header information. It indicates that Virtuoso_DRC_Opt and Virtuoso_IPVS_Adv_Anal_Opt licenses are required for Virtuoso IPVS to run successfully.

Virtuoso IPVS User Guide

Preface

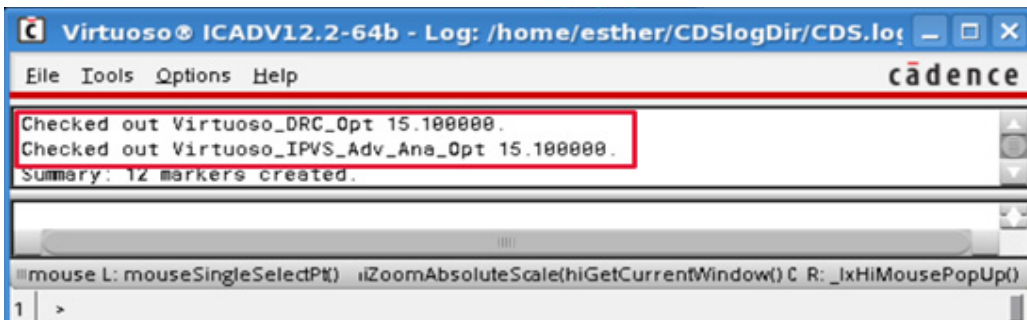
Figure: Snapshot Header

```
"Snapshot name: Mx_Only_ExcludeDecomp"  
- version: 4"  
- bloat: 0.07"  
- license features: (\\"Virtuoso_DRC_Opt\\" \\"Virtuoso_IPVS_Adv_Ana_Opt\\")"  
- PVS version: PVS 15.14.000 s005 64bit 10-25-2015"  
- ruledeck(s): (\\"/grid/cic/catenard_t3_001/esther/RAK/VIPVS_RAK1514/TECHDIR/PVS/PVL/cdn20.config.pv1\\")"  
- created: Mon Oct 26 16:54:11 2015"
```

License Handling while Running VIPVS

Virtuoso IPVS licenses checkout is one time process when running VIPVS for the first time. License checkout is printed in CIW:

Figure: License checkout for the first time

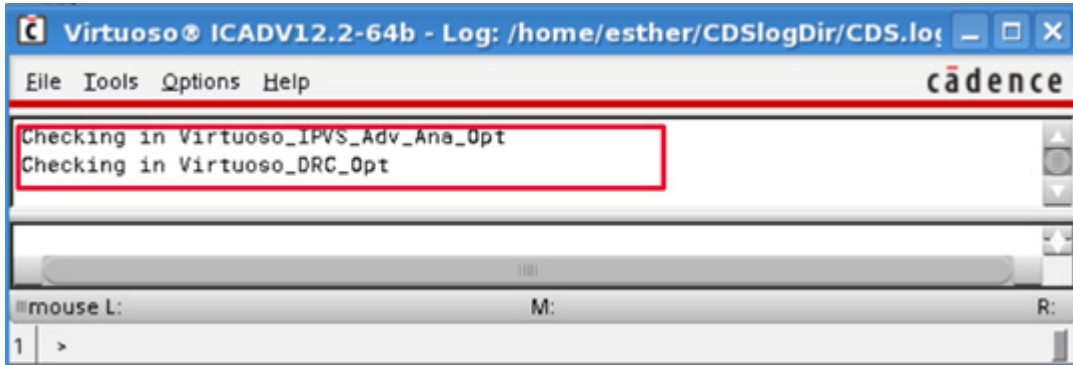


Unlike its sign-off DRC counterpart, license{s} is continuing checkout until any of following termination condition is met:

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

Similarly, license release status is printed in CIW.

Figure: License release status



Related Documents for Virtuoso IPVS

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 Virtuoso IPVS 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 Virtuoso IPVS RAK.

Typographic and Syntax Conventions

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

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

Virtuoso IPVS User Guide

Preface

text

Indicates names of manuals, menu commands, form buttons, and form fields.

Virtuoso IPVS User Guide

Preface

Getting Started

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

The content of this chapter is organized as follows:

- [Prerequisites](#) on page 16
- [Using Virtuoso IPVS for the First Time](#) on page 17

Prerequisites

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

- Snapshot: pre-compiled DRC deck that is loadable by PVS
- Color scheme setup for PVS DRC Browser

When both PVS and Virtuoso are available, VIPVS 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 VIPVS 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](#)

PVS DRC Browser is the default browser for both Virtuoso IPVS and PVS. It has its own set of PVS specific display packages. Thus, it is required for you to integrate this set of display package within Virtuoso for proper error highlight capability. For the quick PVS DRC Browser color scheme integration, refer to:

- [Quick Start: PV Color Scheme Setup in Virtuoso](#)

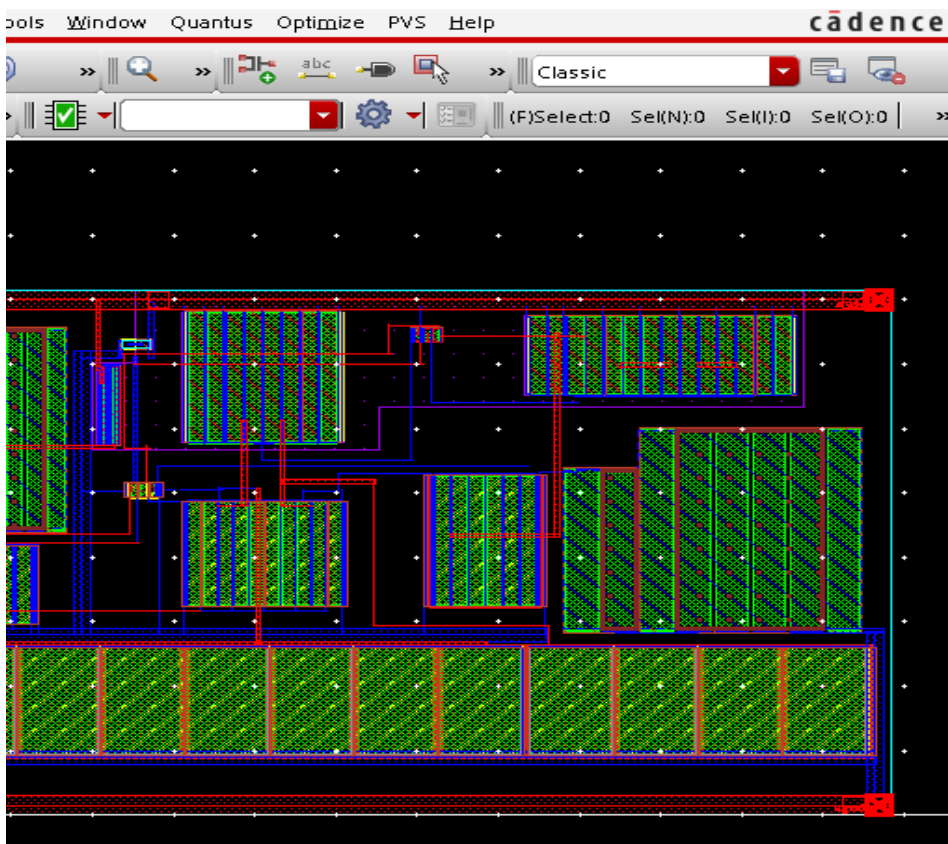
Note: PVS DRC Browser can be either DRCDE or DRC Results Viewer.

Using Virtuoso IPVS for the First Time

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

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

Figure 1-1 Open the Layout



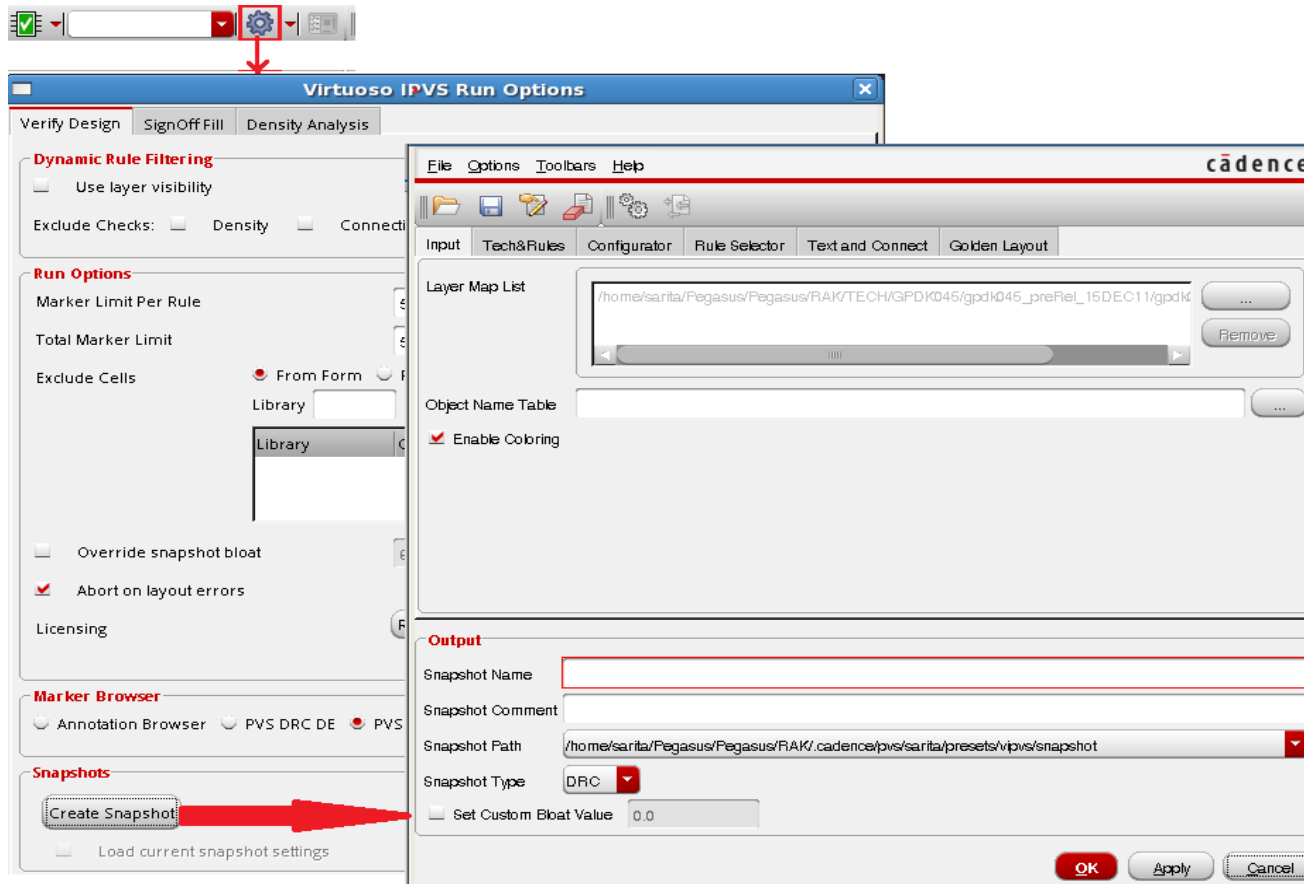
At this point, only *Density_Analysis* snapshot is available. The *Open PVS DRC Browser* option is disabled at this stage.

To create a snapshot, click *VIPVS Run Options* icon, click *Create Snapshot* button on *Virtuoso IPVS Run Options* form and create snapshot using the *Virtuoso IPVS Snapshot Creator* form.

Virtuoso IPVS User Guide

Getting Started

Figure 1-2 Invoke Virtuoso IPVS Snapshot Creator



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 *Open PVS Results Viewer* icon. *Open PVS DRC Browser* icon is inactivated due to no run results available for *Open PVS DRC Browser* to load.

Figure 1-3 Virtuoso IPVS toolbar ready to use



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

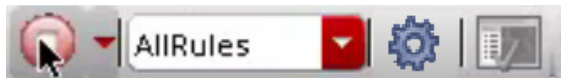
Once VIPVS is running DRC against the layout, the *VIPVS Verify-Design* icon changes to a *Stop* symbol. At this moment, you are free to do any of following:

Virtuoso IPVS User Guide

Getting Started

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

Figure 1-4 Virtuoso IPVS run in progress



Once the run is complete, *PVS DRC Browser* is automatically triggered and has error violations populated in the browser. Meanwhile, these errors are highlighted in the layout as well.

By default, markers are displayed on all cells found in the layout. You can ask VIPVS to generate and display markers by cell master through `drdPVSErrorDisplayMaster`. Once this CDS environmental variable is set, *PVS DRC Browser* populates markers by cell name and first cell markers are highlighted by default.

Note: The `drdPVSErrorDisplayMaster` environment variable is available only in ICADV122 or IC617.

Virtuoso IPVS User Guide

Getting Started

Figure 1-5 Display violations by cell in PVS DRC Browser

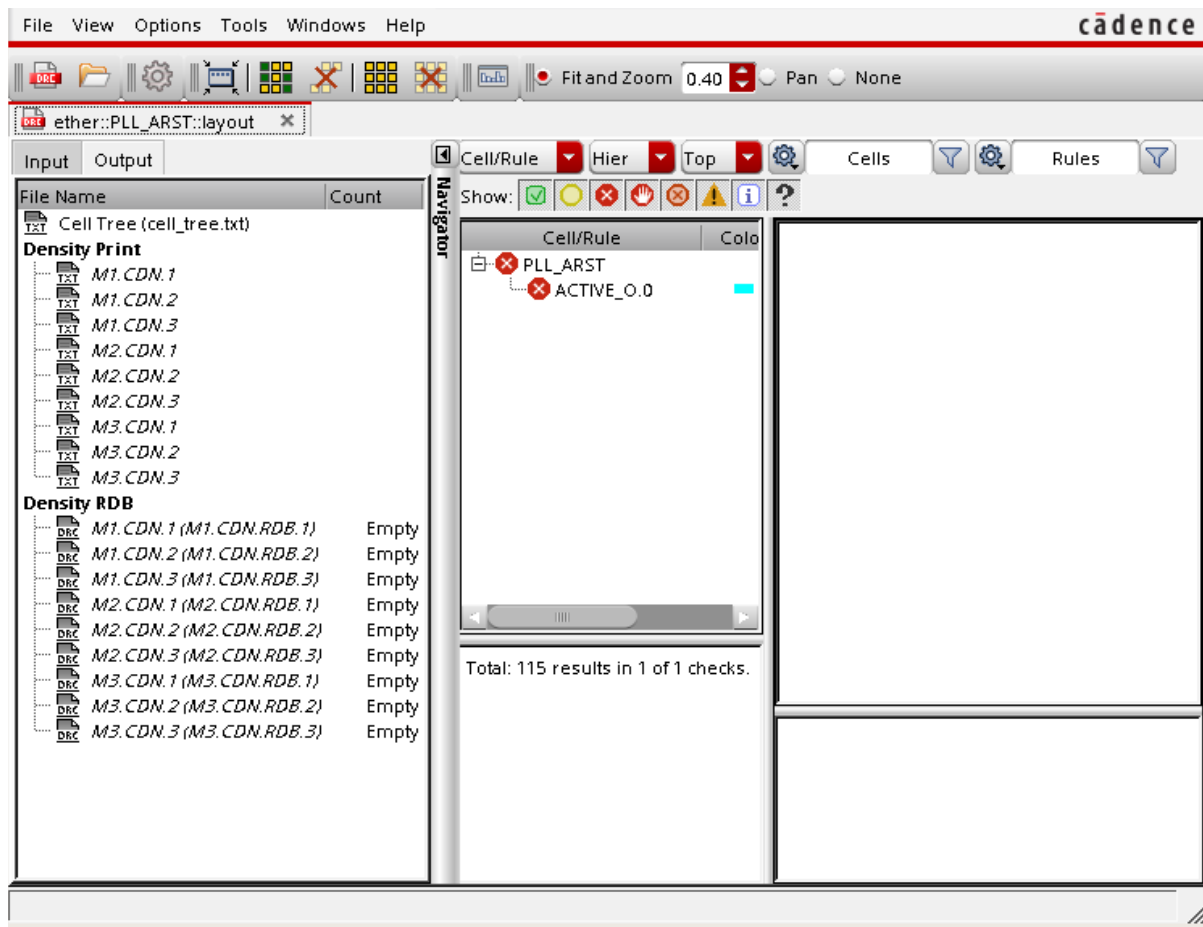
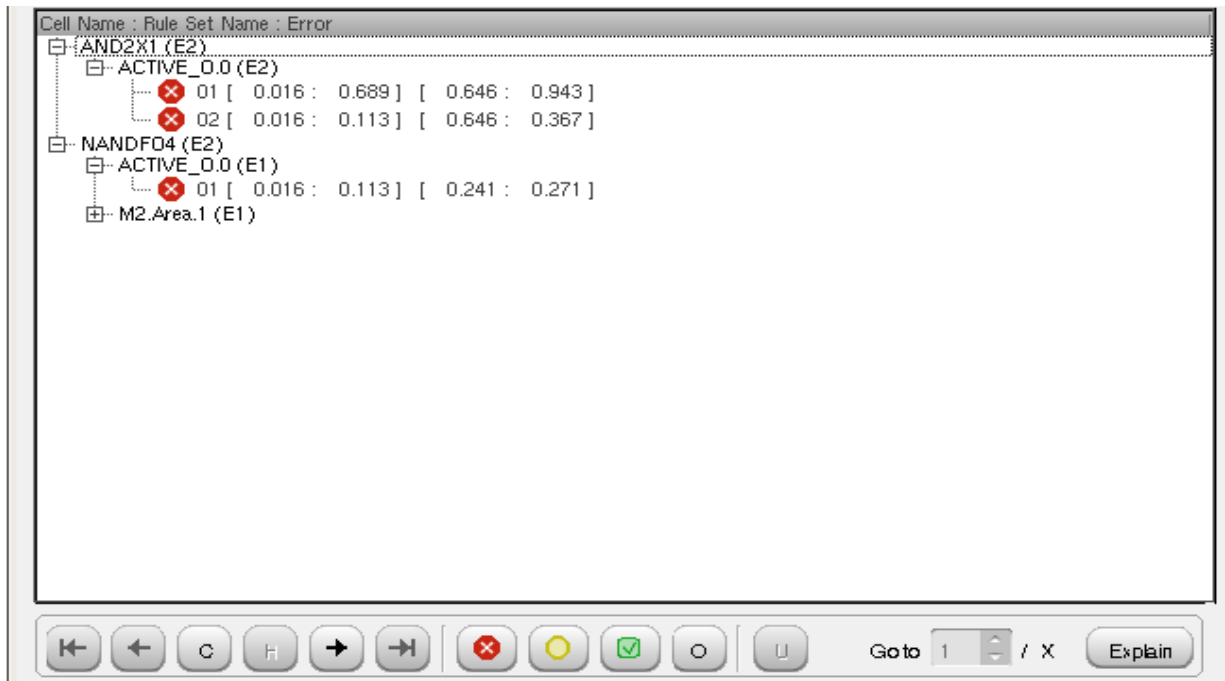
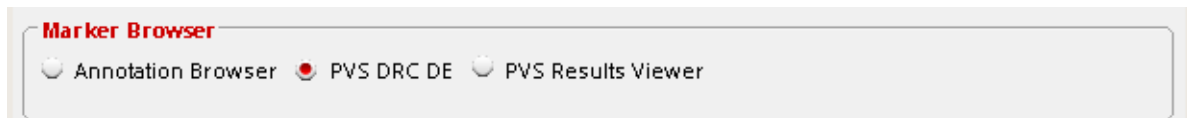


Figure 1-6 Display violations by cell in DRCDE Browser



Note: The browser you use depends on the selection in *Virtuoso IPVS Run Options* form:



Note: In the document, PVS Results Viewer will be used for screenshots. If you use DRCDE then select this option and refer to the *PVS User Guide* section, which describes this tool.

At this point, the *PVS DRC Browser* icon becomes active. You can invoke *PVS DRC Browser* to view the last run results as long as layout is opened regardless its editing status.

Figure 1-7 PVS Results Viewer icon is active



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, Virtuoso IPVS only selects check based on current visible layers in the Palette to run DRC.

Virtuoso IPVS User Guide

Getting Started

Virtuoso IPVS Toolbar

VIPVS 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](#) on page 24
- [Toolbar Icons](#) on page 26
 - [Verify-Design](#) on page 26
 - [Default Active Snapshot](#) on page 26
 - [Run Option](#) on page 27
 - [Error Browsing Utility](#) on page 28
- [Virtuoso IPVS Run Options](#) on page 29
 - [Run Settings for Verify Design](#) on page 30
 - [Run Settings for SignOff Fill](#) on page 49
 - [Run Settings for Density Analysis](#) on page 53

Overview

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

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

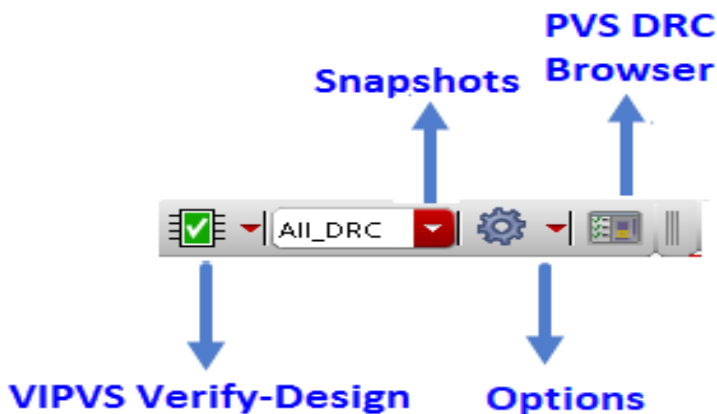
Thus, for a quick DRC during layout editing, VIPVS is more effective than the traditional batch DRC for in design verification solution.

VIPVS allows you to:

- Edit layout within Virtuoso when VIPVS is conducting DRC
- Run DRC on either *editable* or the *read-only* cellview
- Allows you to analyze density for your design
- Generate metal fill in OA database.

Following figure shows the VIPVS toolbar:

Figure 2-1 VIPVS Toolbar Icons



- *VIPVS Verify-Design* or *VIPVS Post-Edit*: DRC operate modes. 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](#).
- *VIPVS Run Options*: For more details on the usage of this toolbar refer to [Run Settings for Verify Design](#)

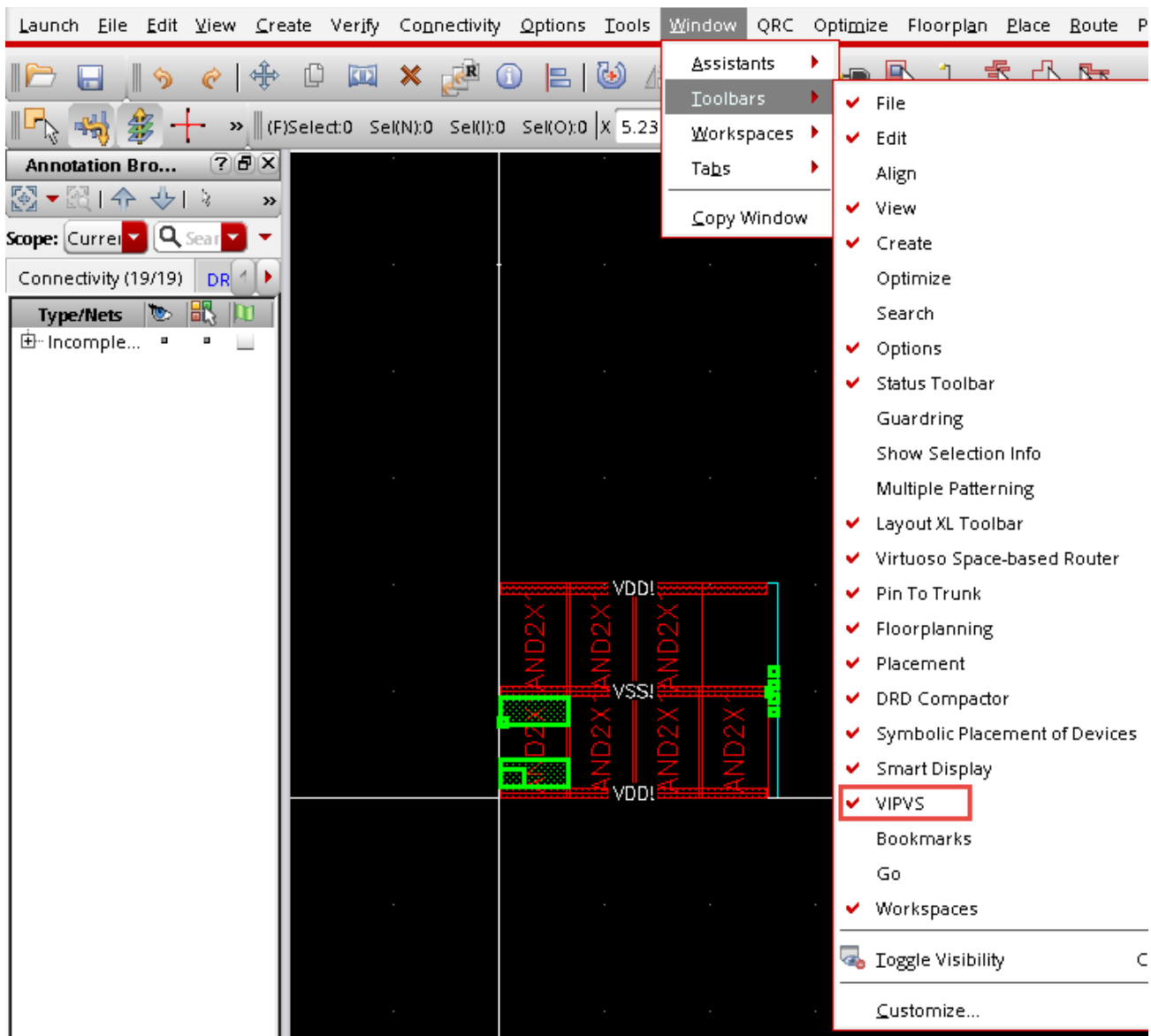
Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

- *PVS DRC Browser*: For more details on the usage of this toolbar refer to [Error Viewing](#).

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

Figure 2-2 Setting VIPVS toolbar visibility

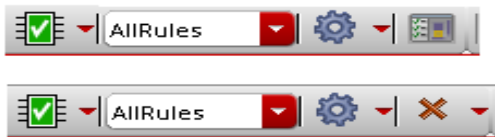


You can also set the visibility of VIPVS toolbar by setting `ipvsSettings.hideToolbar=true` in `.cdsinit`.

Toolbar Icons

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

Figure 2-3 VIPVS Toolbar Customization



Verify-Design

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

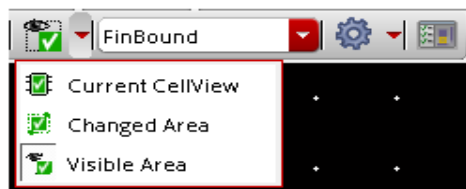
- Current CellView (Default)
- Changed Area
- Visible Area

For details of `Virtuoso_IPVS_initArea` setup, refer to the [Virtuoso CDS Environmental Variables](#) section.

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

Figure 2-4 Default Area check to "Visible Area"

```
setenv Virtuoso_IPVS_initArea viewport
```



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

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

customize a default active snapshot other than tool's selection, you can define it through `VirtuosoIPVS_initSnapshot`.

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

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

Figure 2-5 Set active snapshot

`Default Active Snapshot: AllRules`



`setenv Virtuoso_IPVS_initSnapshot FinBound`



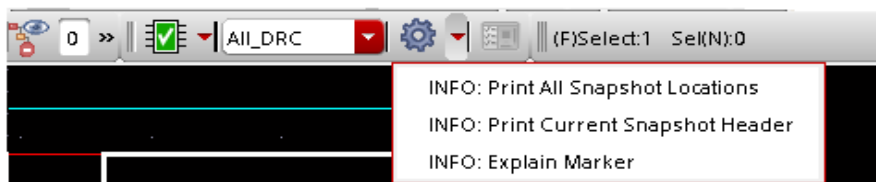
Run Option

You can query the snapshot header information through a snapshot info inquiry utility. It is part of the *Run Option* icon drop-down list. By default, this utility is not enabled.

`VirtuosoIPVS_Info` controls the visibility of this feature.

Figure 2-6 Enable Snapshot Info Inquiry

`setenv Virtuoso_IPVS_info yes`



For details of snapshot info inquiry, refer to [Snapshots Inquiry](#).

Error Browsing Utility

VIPVS allows you to view violations through following options:

- Annotation Browser
- DRC Design Environment (DRCDE)
- PVS Results Viewer (PVS RV)

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

Figure 2-7 Customize default Error Browser

```
setenv Virtuoso_IPVS_markerBrowser PVSDRCDE
```



```
setenv Virtuoso_IPVS_markerBrowser AnnotationBrowser
```



Note: If `Phys_Ver_Sys_Results_Viewer` license is not available, PVS RV option is not visible and DRCDE is the default error browser.

Virtuoso IPVS Run Options

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

Figure 2-8 Virtuoso IPVS Run form

Virtuoso IPVS Run Options

Verify Design | SignOff Fill | Density Analysis

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

Verify Design Setting

Always Reset Changed Area

Run Options

Marker Limit Per Rule: 5000

Total Marker Limit: 5000

Exclude Cells: From Form From File

Library: Cell: View: + Browse

Library	Cell	View

Override snapshot bloat: 0

Abort on layout errors

Licensing: Release License

Marker Browser

Annotation Browser PVS DRC DE PVS Results Viewer

Snapshots

Create Snapshot

Load current snapshot settings

OK Cancel Defaults Apply Save To Load From Help

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 VIPVS behavior, you can invoke the *Virtuoso IPVS Run Options* form by clicking on the *VIPVS 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 *Virtuoso IPVS Run Options* form. Once Virtuoso session is closed, the options interactively set by you are not retained.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

Figure 2-9 Default Option Form

Virtuoso IPVS Run Options

Verify Design | SignOff Fill | Density Analysis

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

Verify Design Setting

Always Reset Changed Area

Run Options

Marker Limit Per Rule: 5000

Total Marker Limit: 5000

Exclude Cells: From Form From File

Library	Cell	View

Override snapshot bloat: 0

Abort on layout errors

Licensing: Release License

Marker Browser

Annotation Browser PVS DRC DE PVS Results Viewer

Snapshots

Create Snapshot

Load current snapshot settings

OK Cancel Defaults Apply Save To Load From Help

Dynamic Rule Filtering (DRF)

Virtuoso IPVS has a built-in intelligence to select rules based on the layer visibility. In other words, you can ask VIPVS 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, VIPVS 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 *Virtuoso IPVS 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.

Figure 2-10 Dynamic Rule Filtering Options (IC6.1.7ISR2 or ICADV12.2ISR2 or newer)

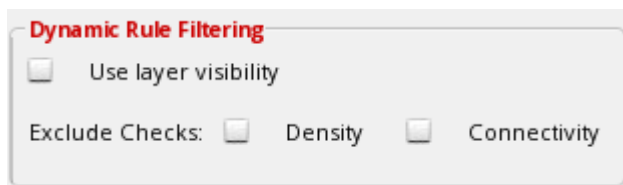
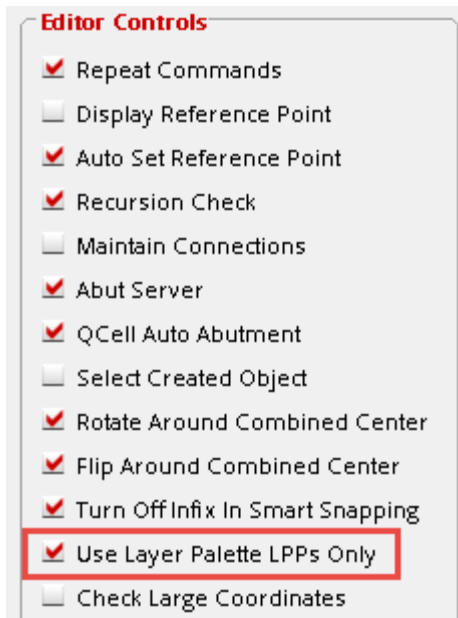


Figure 2-11 Dynamic Rule Filtering Options (IC6.1.7ISR1 or ICADV12.2ISR1 or older)



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.

Figure 2-12 Setting Valid LPP utility



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:

- If you use Virtuoso IC6.1.6ISR1 or ICADV12.2ISR1 or older then to exclude density or connectivity checks you need to enable *Use layer visibility* and then *No density* or *No connectivity* options.
- If you use Virtuoso IC6.1.6ISR2 or ICADV12.2ISR2 or newer then to exclude density or connectivity checks you need to enable *Exclude checks: Density* or *Connectivity* options. These options are independent of *Use layer visibility*.

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

Note: Environment variable settings depend on Virtuoso version.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

Figure 2-13 Customize default DRD setting (IC6.1.7ISR1 or ICADV12.2ISR1 or older)

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

Dynamic Rule Filtering

Use layer visibility

No density No connectivity

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

Dynamic Rule Filtering

Use layer visibility

No density No connectivity

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

Dynamic Rule Filtering

Use layer visibility

No density No connectivity

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

Dynamic Rule Filtering

Use layer visibility

No density No connectivity

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

Dynamic Rule Filtering

Use layer visibility

No density No connectivity

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

Figure 2-14 Customize default DRD setting (IC6.1.7ISR2 or ICADV12.2ISR2 or newer)

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

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

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

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

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

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

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

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

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

Dynamic Rule Filtering

Use layer visibility

Exclude Checks: Density Connectivity

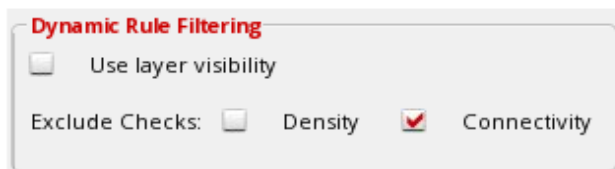
Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

`envSetVal("layout" "drdPVSTFunctionalPresets" 'string "Ignore Visible nodensity")`



`envSetVal("layout" "drdPVSTFunctionalPresets" 'string "Ignore Visible noconnect")`



`envSetVal("layout" "drdPVSTFunctionalPresets" 'string "Ignore Visible noconnect nodensity")`



Run Options

Four elements are grouped under general run option field:

- Marker Limit
- Exclude Cells
- Run Termination Control
- Licensing Control

Figure 2-15 General Run Option content outline



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:

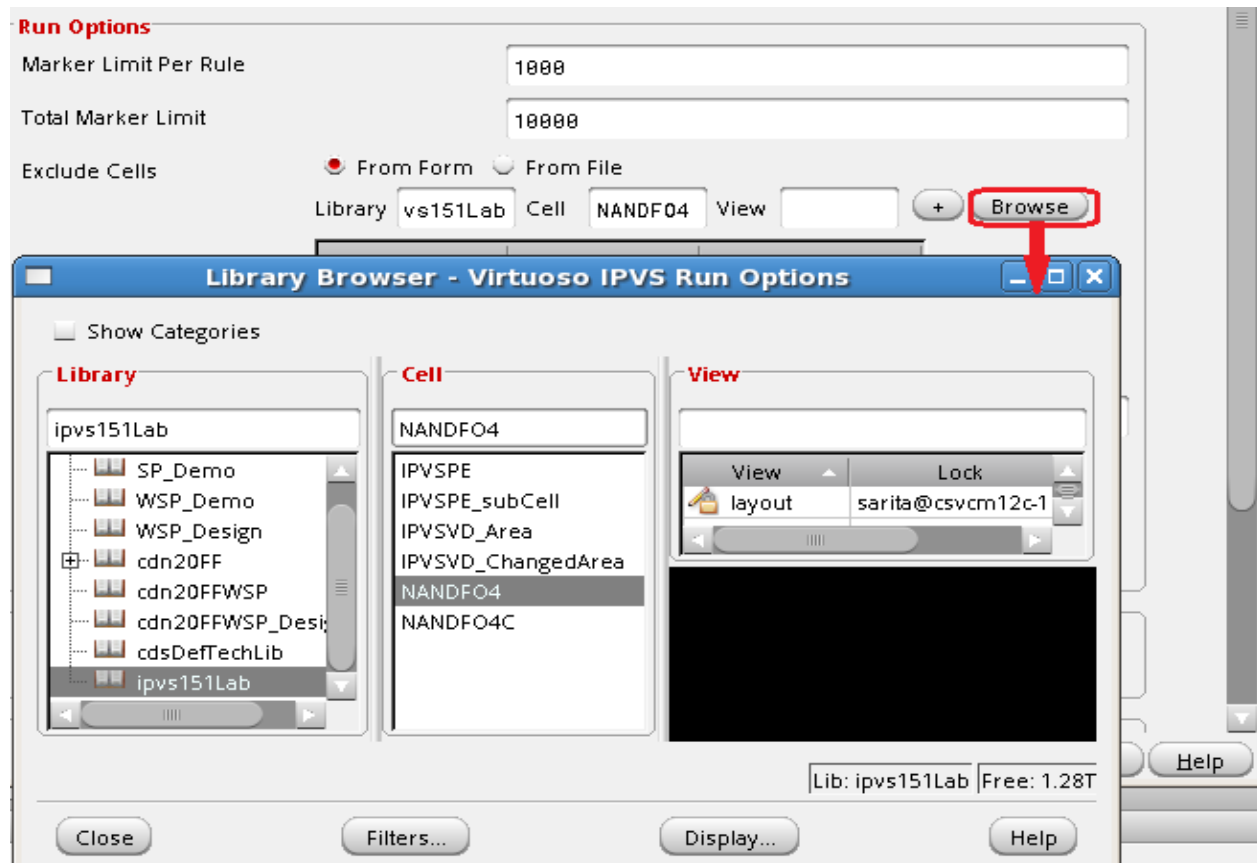
- *From Form*: Library management user interface.
- *From File*: A file containing cellview information.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

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.

Figure 2-16 Set Cellview information through 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:

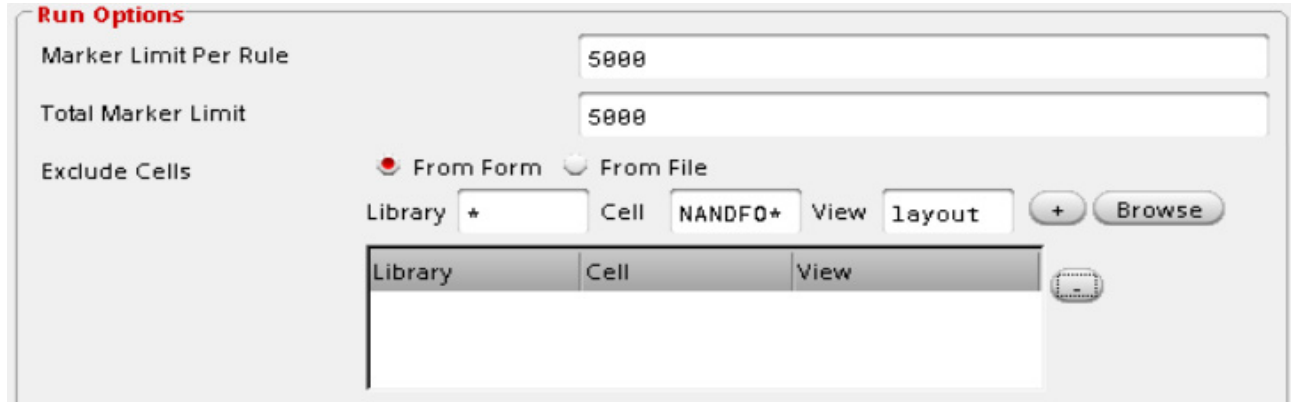
- * : Match zero or more characters
- ? : Match exact one character

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

Virtuoso IPVS User Guide

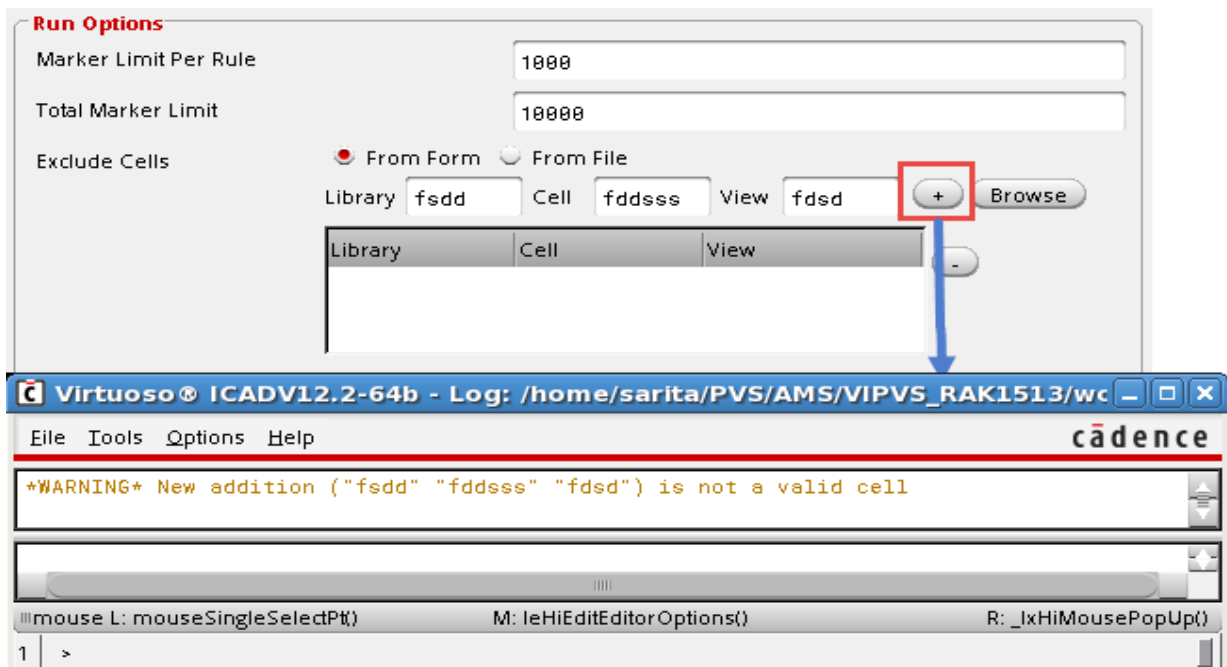
Virtuoso IPVS Toolbar

Figure 2-17 Exclude Cell based on searching criteria



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 VIPVS to process.

Figure 2-18 Cellview validity check



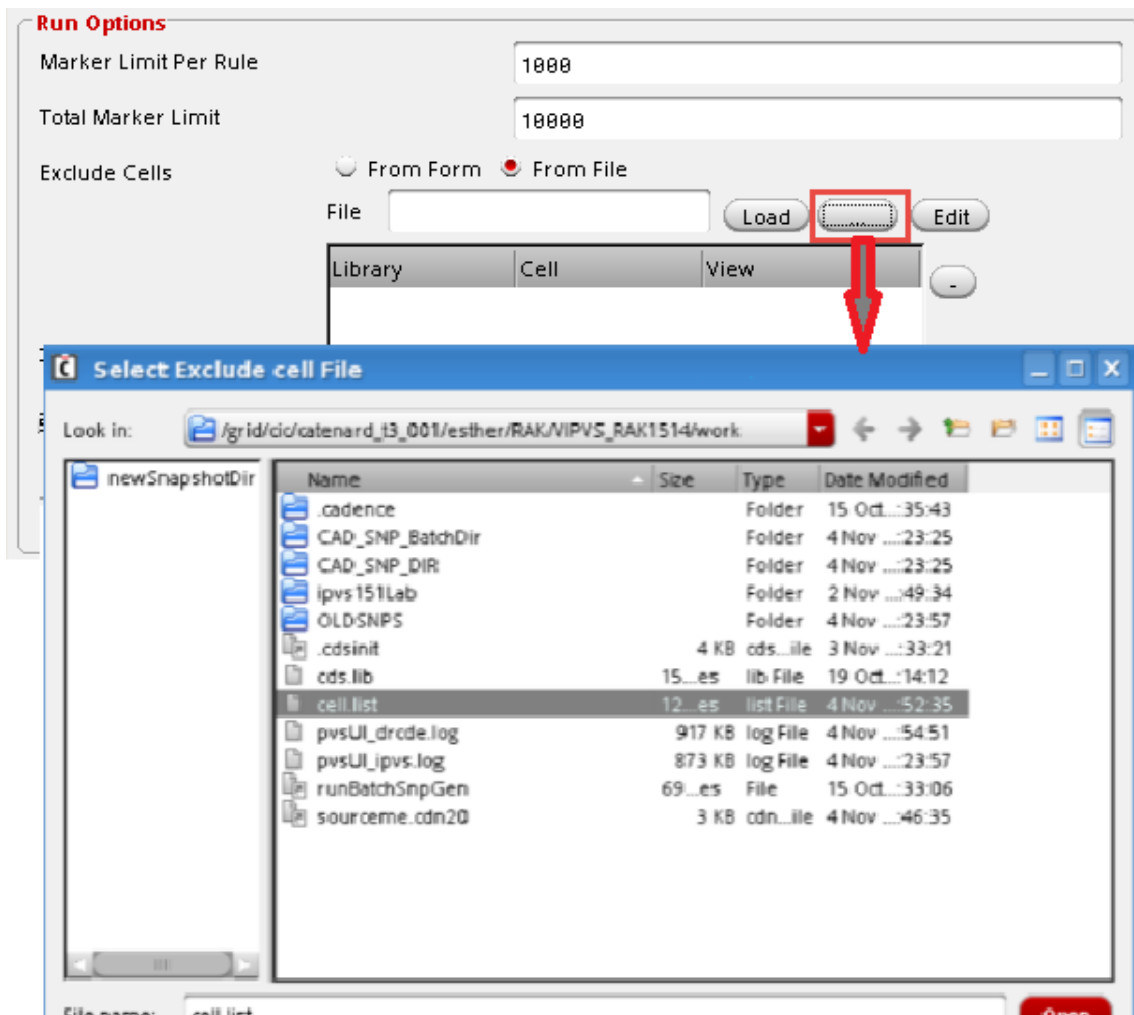
In general, VIPVS checks cellview information against libraries declared in cds.lib. If it is not in cds.lib, it is considered not accessible by VIPVS.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

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.

Figure 2-19 Select file contains cell information



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, VIPVS checks:

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

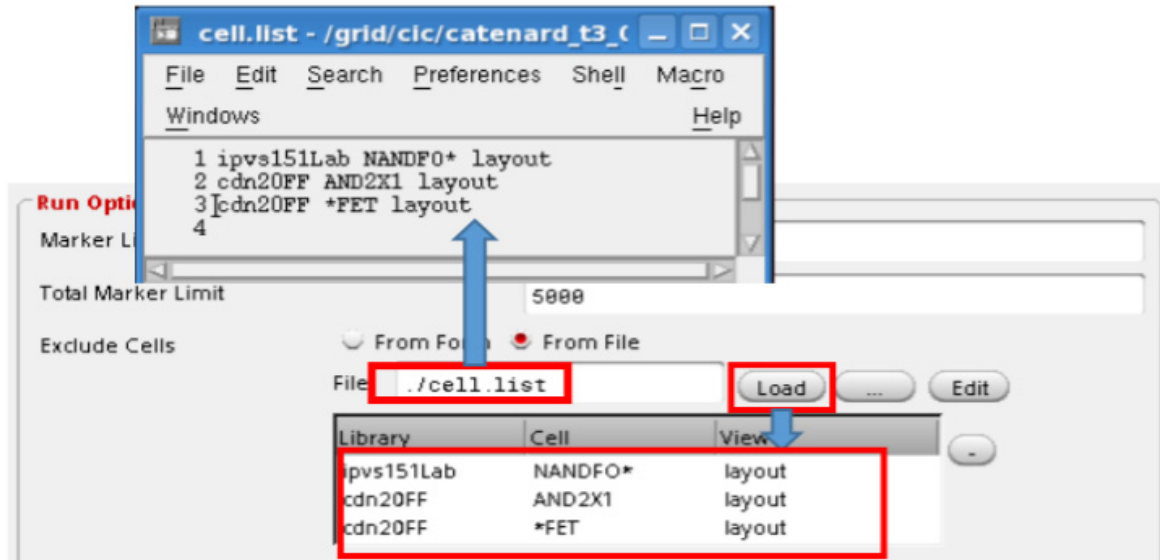
If any insufficient information is detected, warning messages are displayed in CIW.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

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*.

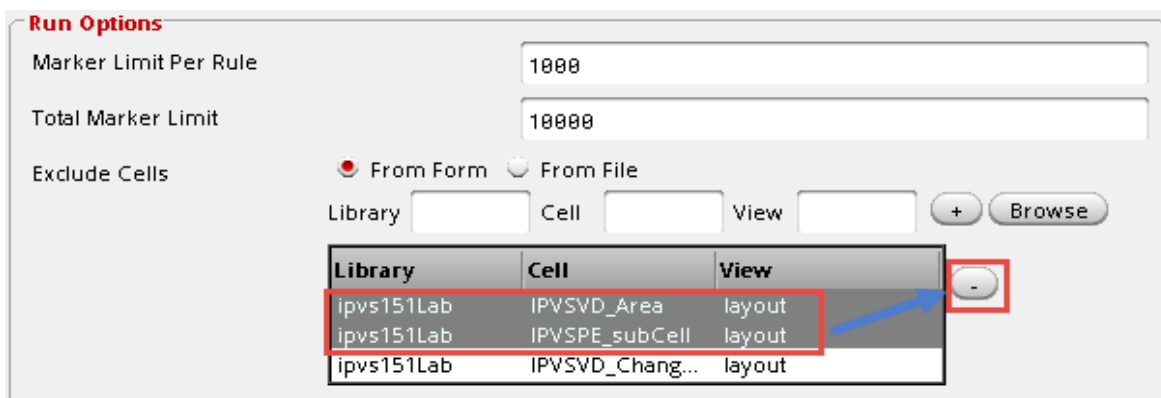
Figure 2-20 Load cell.list content



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.

Figure 2-21 Remove cell from Exclude Cells table



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

Bloat Override Control

Virtuoso IPVS 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 or defined through `drdPVSMInPostEditBloat` CDS environmental variable.

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

Figure 2-22 Override snapshot bloat option of Virtuoso IPVS Run Option form



The image shows a user interface element for the 'Override snapshot bloat' option. It consists of a checkbox with a red checkmark to its left, followed by the text 'Override snapshot bloat'. To the right of this text is a rectangular text input field containing the number '1'.

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 VIPVS DRC run:

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







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

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

Virtuoso IPVS User Guide

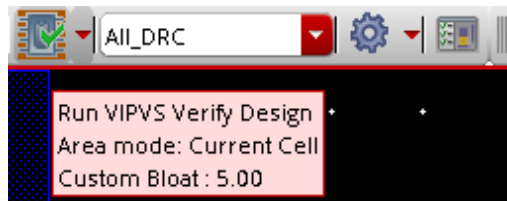
Virtuoso IPVS Toolbar

Also, enabling of *Override snapshot bloat* triggers the *Verify Design* and *Post-Edit* icons update in the VIPVS 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:

Figure 2-23 Bloat Value tooltip



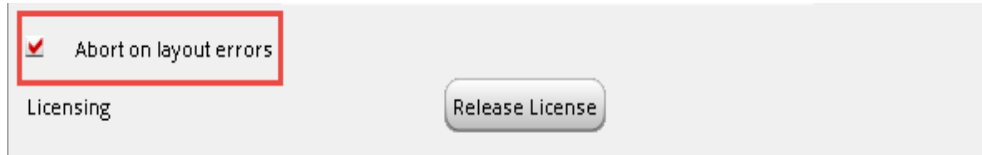
Run Termination Control

By default, VIPVS 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.

Figure 2-24 Default VIPVS run termination setting

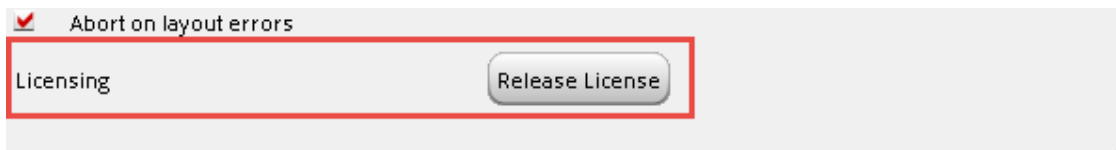


To continue DRC despite of outcome of layout validity, deselect the check box. In this case, errors found during layout validity check will be treated as soft errors and run proceeds as it-is.

Licensing Control

As described in *Licenses Requirements and Behaviors*, VIPVS 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 VIPVS session. Only license(s) documented in snapshot header is released. License pertaining to DRCDE is controlled by its own activity process.

Figure 2-25 Release License Trigger

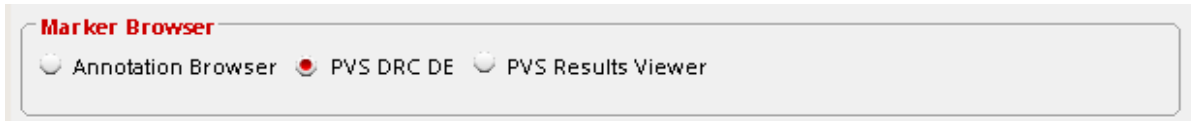


Marker Browser

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

Since VIPVS retains last run result in an ascii format, you can reload this data any time by invoking DRCDE or PVS Results Viewer through the toolbar. On the contrary, you cannot view last run results via Annotation Browser on demand. For example, VIPVS' latest run generates OA markers on the layout. You would like to first view results in DRCDE or PVS 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 DRCDE or PVS Results Viewer when you toggle from Annotation Browser to DRCDE or PVS Results Viewer. However, OA markers are not restored once you toggle back to Annotation Browser from DRCDE or PVS Results Viewer.

Figure 2-26 Default Browser setting



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:

- Virtuoso_IPVS_maintenance
- Virtuoso_IPVS_enableSNPCreation
- Virtuoso_IPVS_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 `Virtuoso_IPVS_enableSNPCreation`. Depending on the number of snapshot directories, CAD can decide whether to disable or enable Snapshots management field through `Virtuoso_IPVS_enableSNPManagement`.

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

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

Figure 2-27 Disable Snapshots Field

```
setenv Virtuoso_IPVS_enableSNPCreation no
setenv Virtuoso_IPVS_enableSNPManagement no
```

Dynamic Rule Filtering

- Use layer visibility
- No density No connectivity

Verify Design Setting

- Always Reset Changed Area

Run Options

Marker Limit Per Rule: 1000

Total Marker Limit: 10000

Exclude Cells: From Form From File

Library: vs151Lab Cell: NANDF04 View: + Browse

Library	Cell	View

Override snapshot bloat: 1

Abort on layout errors

Licensing: Release License

Marker Browser

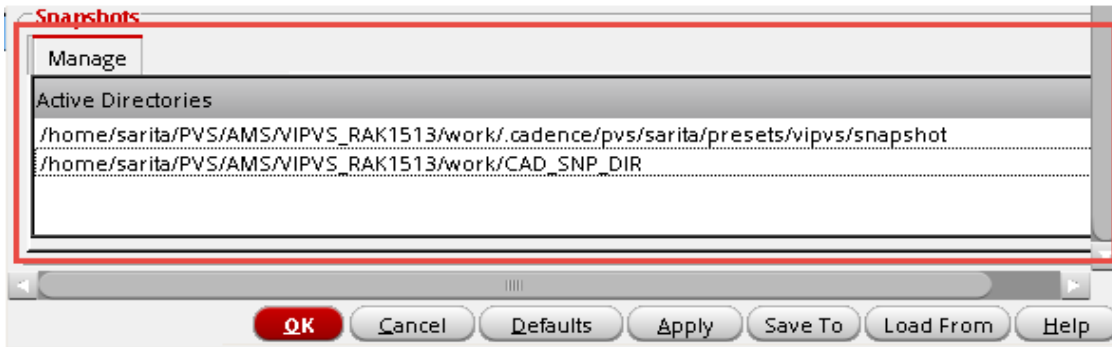
Annotation Browser PVS DRC DE PVS Results Viewer

OK Cancel Defaults Apply Save To Load From Help

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:

Figure 2-28 Snapshot management utility

```
setenv Virtuoso_IPVS_enableSNPCreation no  
setenv Virtuoso_IPVS_enableSNPManagement yes
```

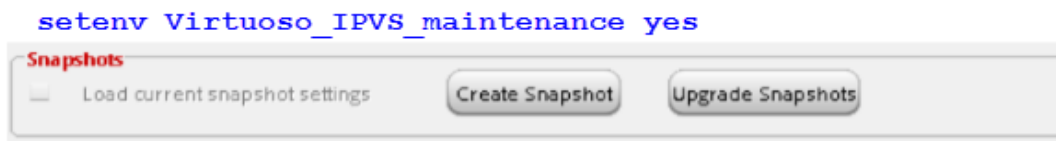


For details of snapshot management, refer to [Manage Snapshots](#).

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 VIPVS. This utility can be enabled through Virtuoso_IPVS_maintenance.

Figure 2-29 Enable Snapshot management



For details of creating snapshot, refer to [Quick Start: Create Snapshot](#).

For details of upgrading snapshot, refer to [Snapshots Upgrade](#).

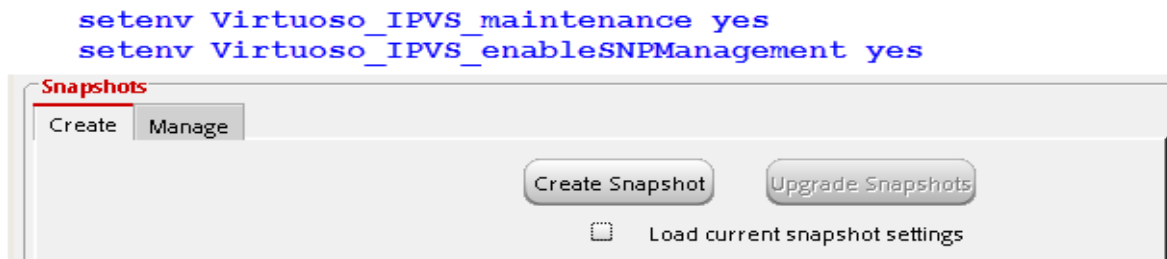
Scenario Three: Global and Local Snapshot Control by User

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.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

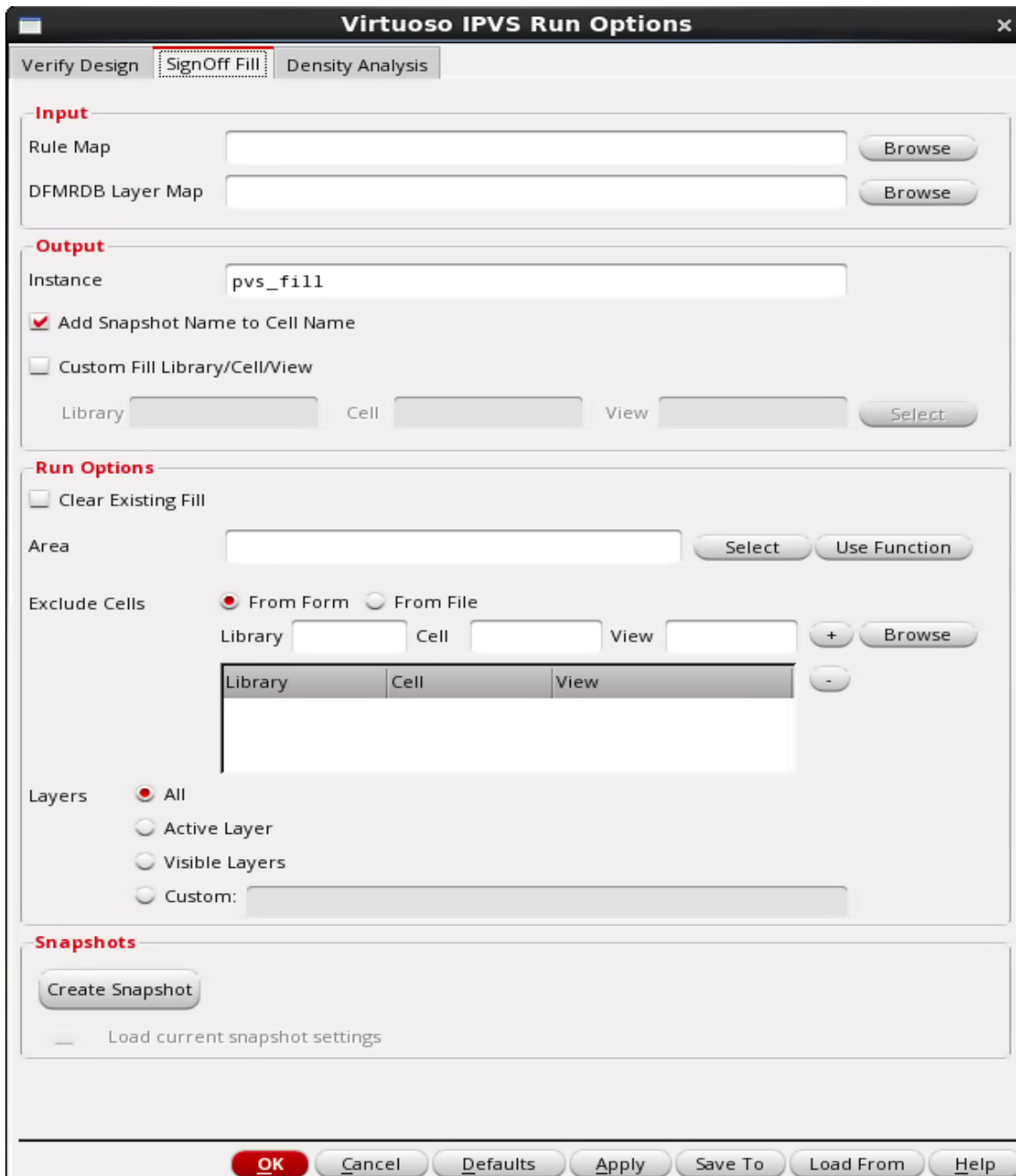
Figure 2-30 Complete Snapshots field



Run Settings for SignOff Fill

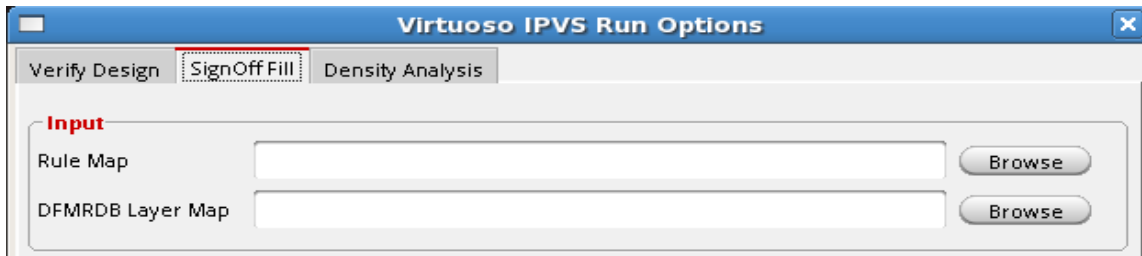
Click the *SignOff Fill* tab to define run settings for SignOff Fill.

Figure 2-31 SignOff Fill tab



Input

Figure 2-32 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 VIPVS 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

Figure 2-33 Output Section



- **Instance:** Specifies prefix of instance name of fill cell. VIPVS SignOff 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 `pvs_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 `pvs_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

Figure 2-34 Run Options Section



Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

- **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.
- **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.
- **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 37.
- **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/purpose in the *Custom* layer field is the following:

`<layername>/<purposename> ... <layername>/<purposename>`

Wildcards are supported.

Example 1:

`Metal1/drawing`

Fills will be generated for `Metal1` layer.

Example 2:

`Metal1/drawing Metal2/drawing Metal3/drawing`

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

Example 3:

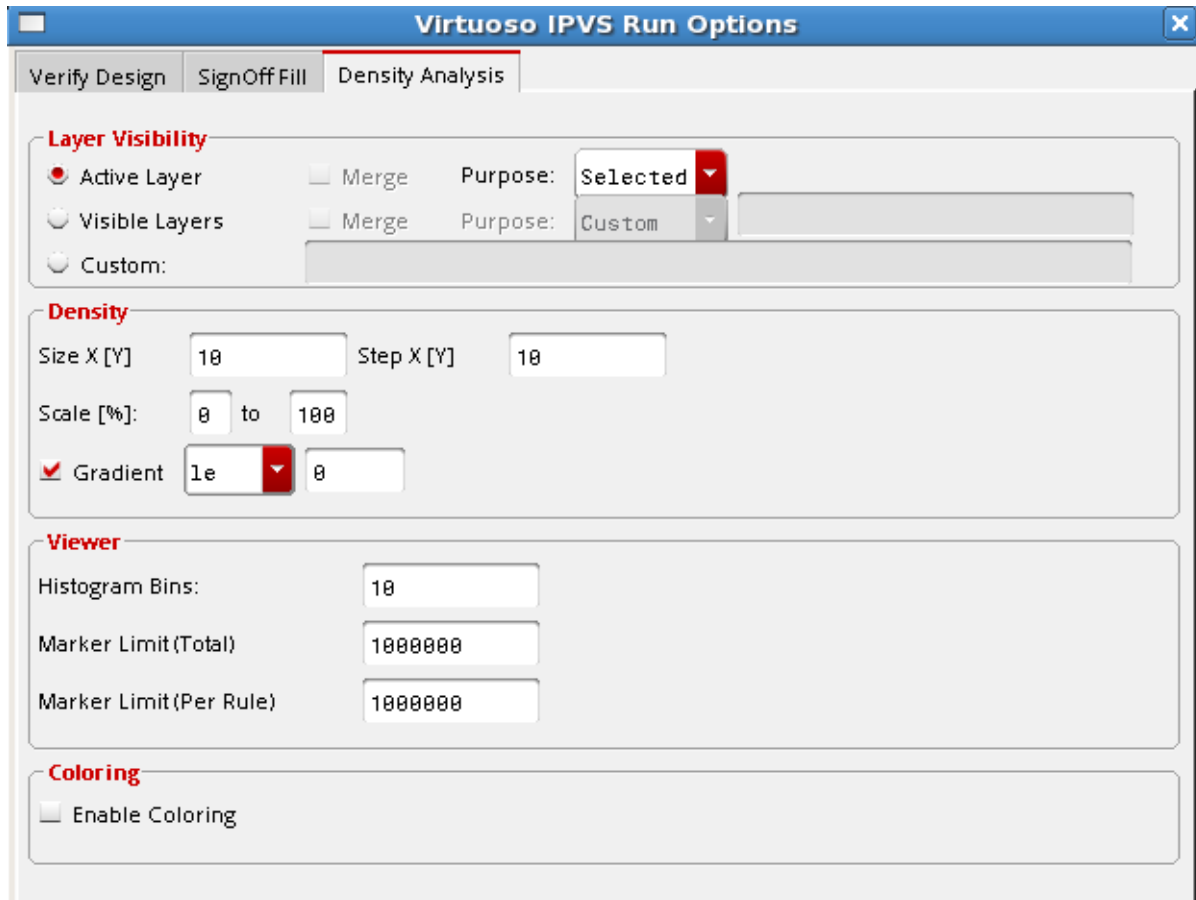
`Metal*/drawing`

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

Run Settings for Density Analysis

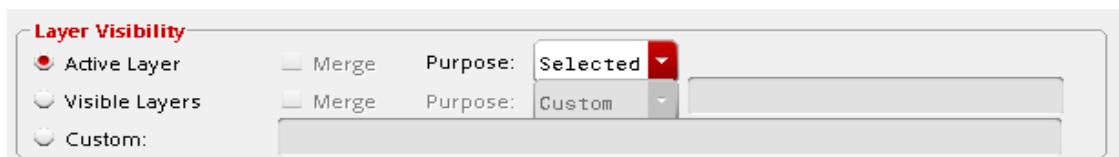
Density analysis generates density results from VIPVS toolbar for layers of interest. Rule deck is not required. Results are shown in PVS Results Viewer as heatmap and histograms.

Figure 2-35 Density Analysis Tab



Layer Visibility

Figure 2-36 Layer Visibility Section



Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

- **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:

Metall/drawing

Density Anlalysis of Metall/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.

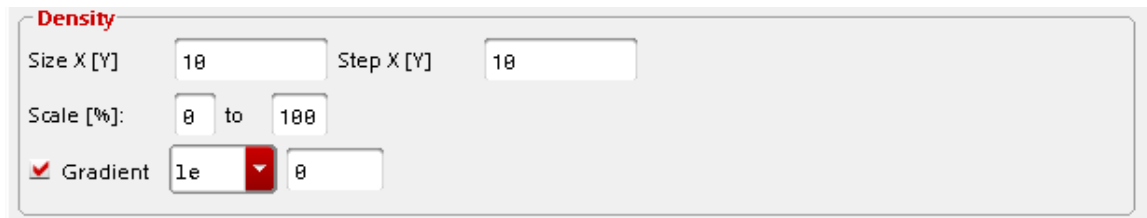
Example 4:

(Met*/drawing)

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

Density


Figure 2-37 Density Section



- **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 *PVS Developer Guide*.

Viewer

Figure 2-38 Viewer Section



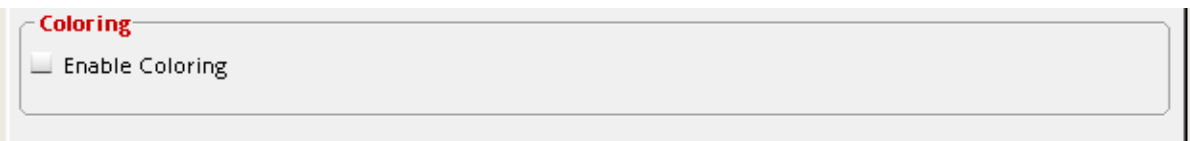
Viewer

Histogram Bins:	<input type="text" value="10"/>
Marker Limit (Total)	<input type="text" value="1000000"/>
Marker Limit (Per Rule)	<input type="text" value="1000000"/>

- *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

Figure 2-39 Coloring Section



Coloring

Enable 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, VIPVS 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.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

- **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 Virtuoso IPVS Preferences* form:

- **File Name:** Enter the file name for the VIPVS 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/pvs/$USER` directory and `.cadence/pvs/$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 Virtuoso IPVS 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 `$PVS_INSTALL_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.

Virtuoso IPVS User Guide

Virtuoso IPVS Toolbar

\$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.

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](#) on page 60
- [Snapshot Directory Types and Customizations](#) on page 62
- [Quick Start: Create Snapshot](#) on page 63
 - [First Time](#) on page 64
 - [From Preset](#) on page 66
 - [From Current Active Snapshot Setting](#) on page 67
- [Virtuoso IPVS Snapshot Creator Form](#) on page 69
 - [Snapshot Content Customization Field](#) on page 69
 - [Snapshot Output Field](#) on page 86
- [Manage Snapshots](#) on page 88
 - [Snapshots Inquiry](#) on page 89
 - [Snapshots Management](#) on page 93
 - [Snapshots Upgrade](#) on page 96
- [Portability of Snapshots](#) on page 99
- [Checking that Snapshot is Generated by Up-to-Date DK Version](#) on page 100

Snapshot Definition and Files

A snapshot captures design rules in a compiled binary format, which only PVS understands. This format allows PVS 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.

Table 3-1 Snapshot Files Outline

Snapshot Files	Description
<i>SnpName.preset</i>	<ul style="list-style-type: none"> ■ Preset file that contains Snapshot Creator GUI configuration setting for a given snapshot
<i>SnpName.tag</i>	<ul style="list-style-type: none"> ■ Reference for Dynamic Rule Filtering
<i>SnpName.cap</i>	<ul style="list-style-type: none"> ■ Cap file contains default rule selectivity, rule caption, rule types and rule description. ■ Reference for Dynamic Rule Filtering
<i>SnpName.snp</i>	<ul style="list-style-type: none"> ■ Snapshot: the compiled version rule deck
<i>SnpName.layermap</i>	<ul style="list-style-type: none"> ■ Technology layer mapping file
<i>SnpName.objectmap</i>	<ul style="list-style-type: none"> ■ VIPVS uses this reference to lookup OA specific object to its equivalent stream pair
<i>SnpName.rules</i>	<ul style="list-style-type: none"> ■ VIPVS uses this file to store rule file names, used to generate snapshot.
<i>SnpName.sum</i>	<ul style="list-style-type: none"> ■ VIPVS uses this file to check the validity of overall snapshot reference files.

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

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

For dynamic rule selection, VIPVS 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, VIPVS transfers these LPP specific polygons to their Layer-Datatype stream pair equivalents and then passes them directly to PVS. This is a direct communication between Virtuoso and PVS through VIPVS interface socket. In addition to transporting geometry data, VIPVS 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.

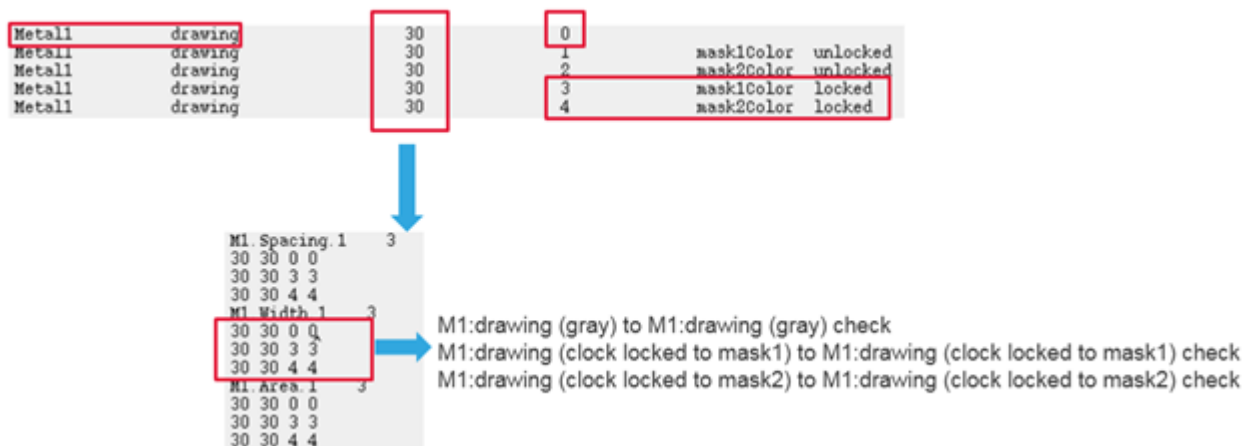
```
<Rule Caption> <# of different layer pair's check>
<LayerStream1#> <LayerStream2#> <Datatype1#> <Datatype2#>
:
```

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.

Figure 3-1 Layer Lookup Example



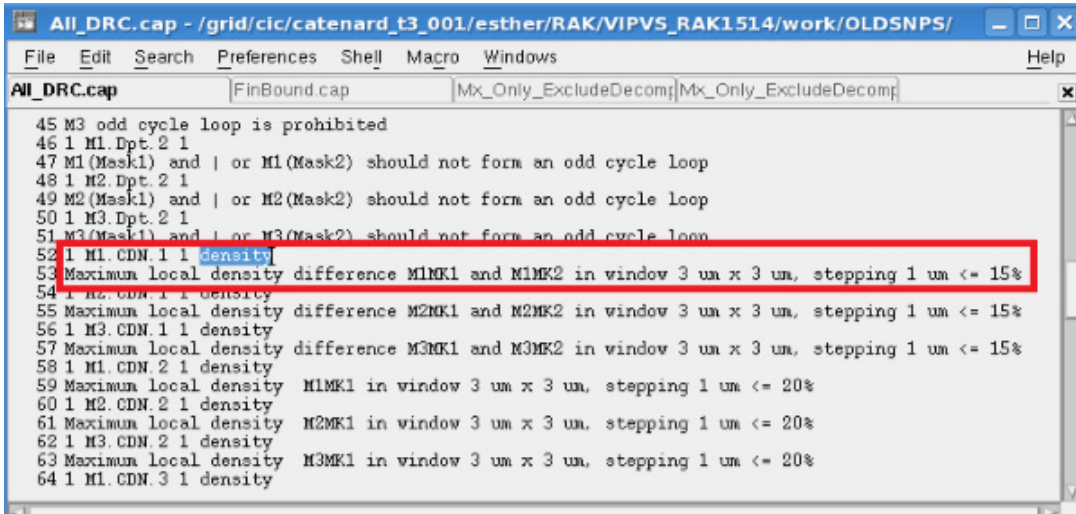
For the .cap file, it contains rule types that VIPVS refers to for either density or connectivity type rules exclusion. Below example shows a typical content of .cap file. Its rule status' syntax is composed as following:

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

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

Figure 3-2 .cap general content



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 PVS 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, VIPVS refers to .cap file to ensure rule selection only on rules existed in the snapshot (.snp).

Finally .sum is for VIPVS to check snapshot files completeness. For a given snapshot validity check, VIPVS 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, VIPVS by default stores generated snapshot files under .cadence/pvs/<userid>/presets/vipvs/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/pvs. However, VIPVS only checks and loads snapshots from the snapshot directory matched current user information that is queried through 'echo \$user' or 'whoami'.

During VIPVS 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.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

If the potential snapshot passes these checks, it is included in the VIPVS's snapshot drop-down list. Otherwise, VIPVS issues warning messages in CIW and this in-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.

VIPVS allows you to define one or multiple snapshot directories through `PVS_SNAP_SHOTS_DIRS`. Once this shell environmental variable is set, VIPVS checks and honors the contents of snapshot directories declared through this shell variable. If the snapshot directory is not available, VIPVS 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 `PVS_SNAP_SHOTS_DIRS`, VIPVS 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`:

Figure 3-3 Example of declaring user specific snapshot directories

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

setShellEnvVar("PVS_SNAP_SHOTS_DIRS" strcat(localsnpdir ":" globalsnpdir))
```

`PVS_SNAP_SHOTS_DIRS` supports path inputs containing following syntax:

- Relative Path
- Environmental variables

VIPVS checks path validity and displays the absolute snapshot directory path in the *Virtuoso IPVS Snapshot Creator and VIPVS Manage* field. The order of directory paths in `PVS_SNAP_SHOTS_DIRS` dictates its display order in both of *Virtuoso IPVS Snapshot Creator and VIPVS Manage* field.

Quick Start: Create Snapshot

This section introduces different ways of creating snapshots

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

Virtuoso IPVS User Guide

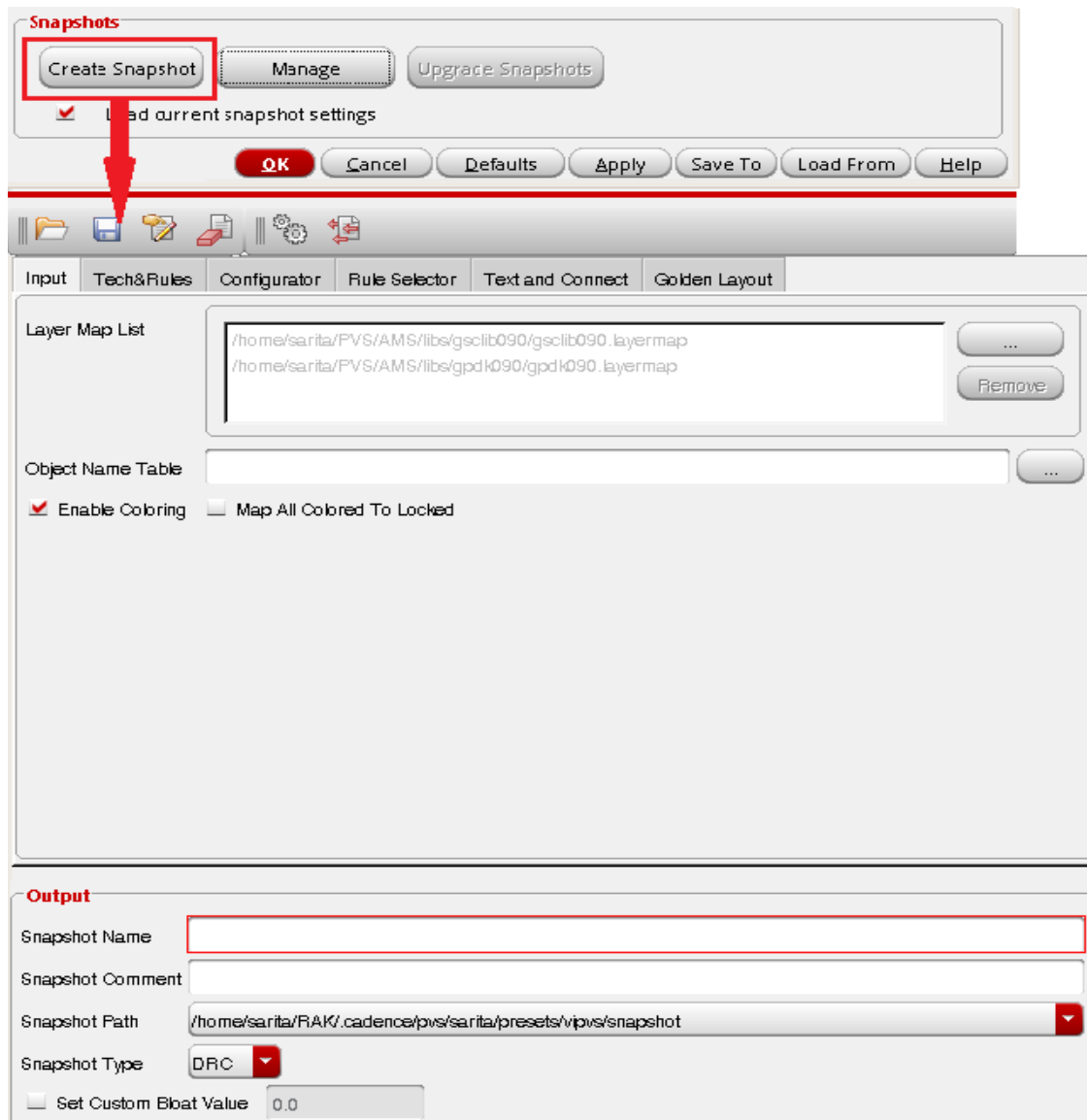
All about Design Rules: Snapshot

- Create a snapshot from current snapshot setting

First Time

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

Figure 3-4 Invoke VIPVS Snapshot Creator GUI



Virtuoso IPVS User Guide

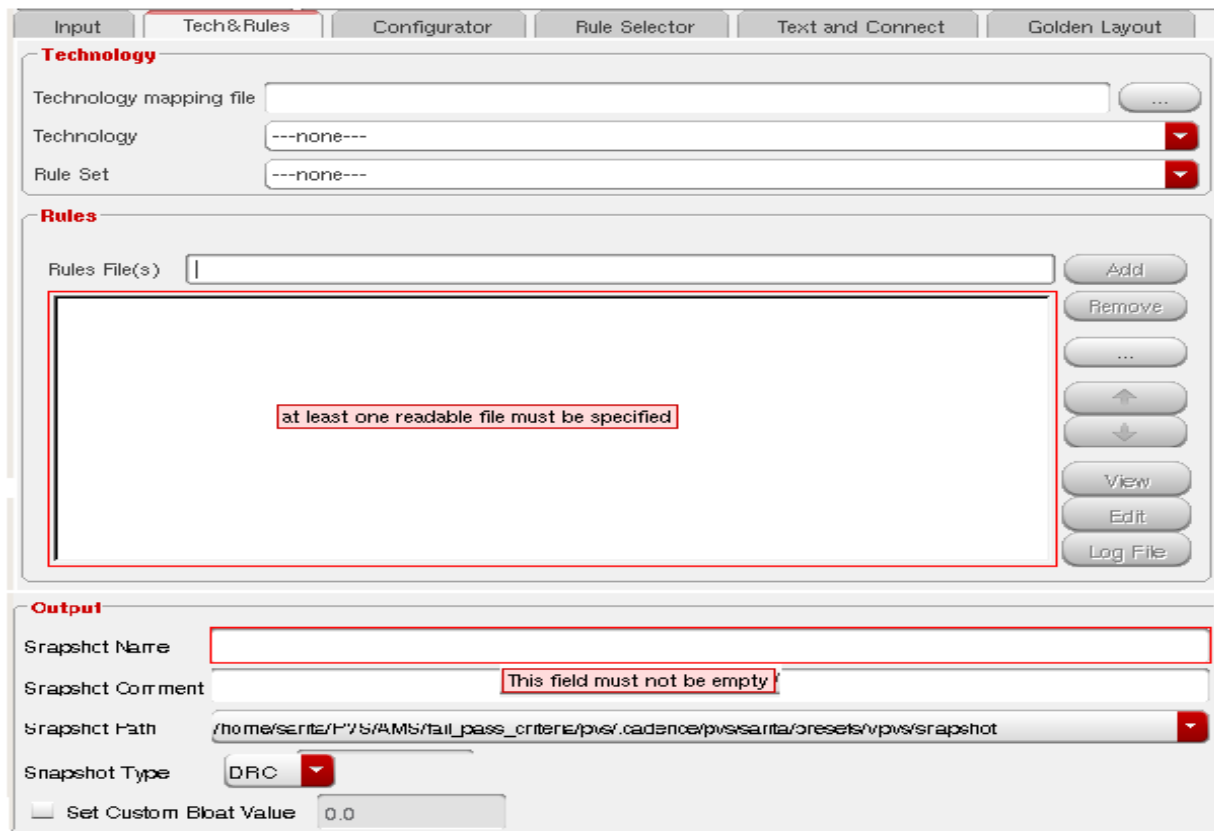
All about Design Rules: Snapshot

To generate a snapshot using Virtuoso IPVS Snapshot Creator, three minimum inputs are required:

- Layermap file
- Rule Deck
- Output Snapshot name

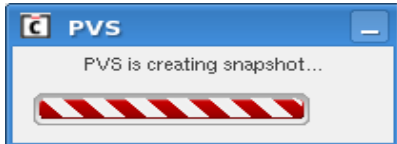
Assuming no technology is set up, VIPVS flags missing required inputs visually by boxing fields with the red highlight. Once the cursor hovers over the red-box, error message is displayed as a tooltip. Following example illustrates error dictated by the interface. Till required inputs are incomplete,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

Figure 3-5 Input Validity Check



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

Figure 3-6 Snapshot Creation in Progress



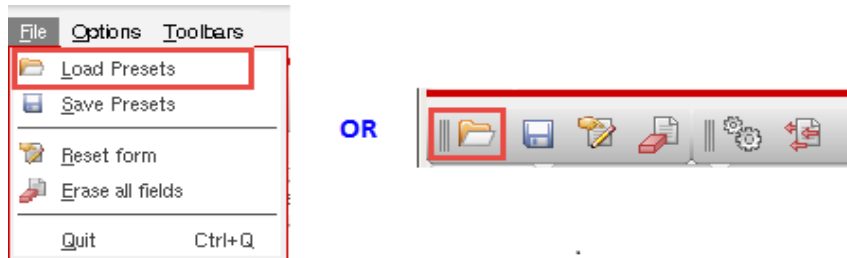
The newly created snapshot is automatically populated under VIPVS toolbar's snapshot combo field.

From Preset

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

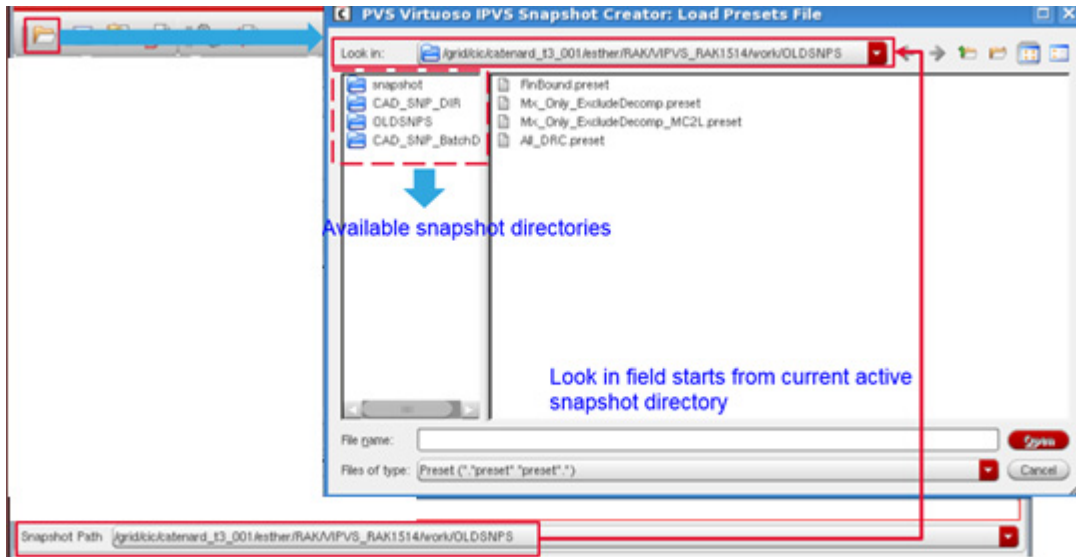
To load the preset, you can interactively click the *Load Presets* icon on the *File* toolbar or *File -> Load Presets*.

Figure 3-7 Invoke load presets GUI



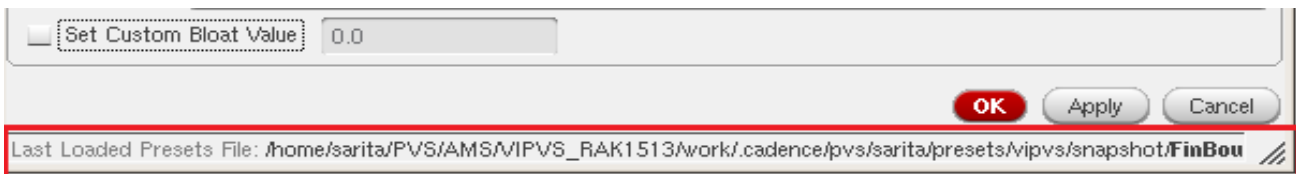
It opens a file searching form. For the current *Look in* directory, files' name with suffix `.preset` are populated in the window. Current *Look in* directory is initially set to current snapshot directory populated under the *Snapshot Path* field. You can navigate to any of available snapshot directory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

Figure 3-8 Loading Presets GUI



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

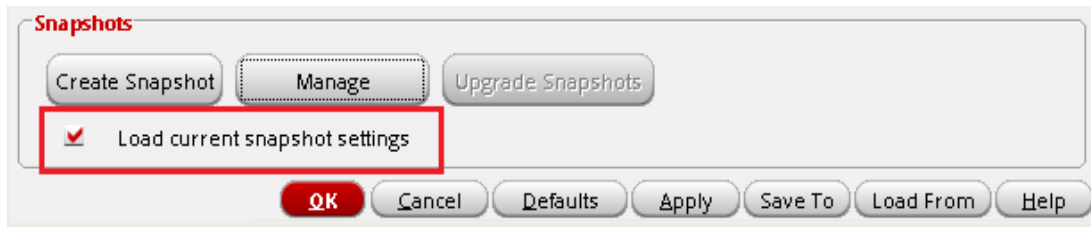
Figure 3-9 Preset loading status



From Current Active Snapshot Setting

Similar to creating snapshot from a preset file, VIPVS 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.

Figure 3-10 Load current snapshot setting



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

Virtuoso IPVS Snapshot Creator Form

This form is another derivation of PVS run submission form. However, its contents are specific to VIPVS snapshot creation only. You can invoke this GUI within Virtuoso session or standalone through `ipvs` invocation utility. You can refer to [Appendix A - Preset to Snapshot Batch Migration](#) for standalone GUI invocation.

Similar to PVS submission form, Virtuoso IPVS 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

Snapshot Content Customization Field

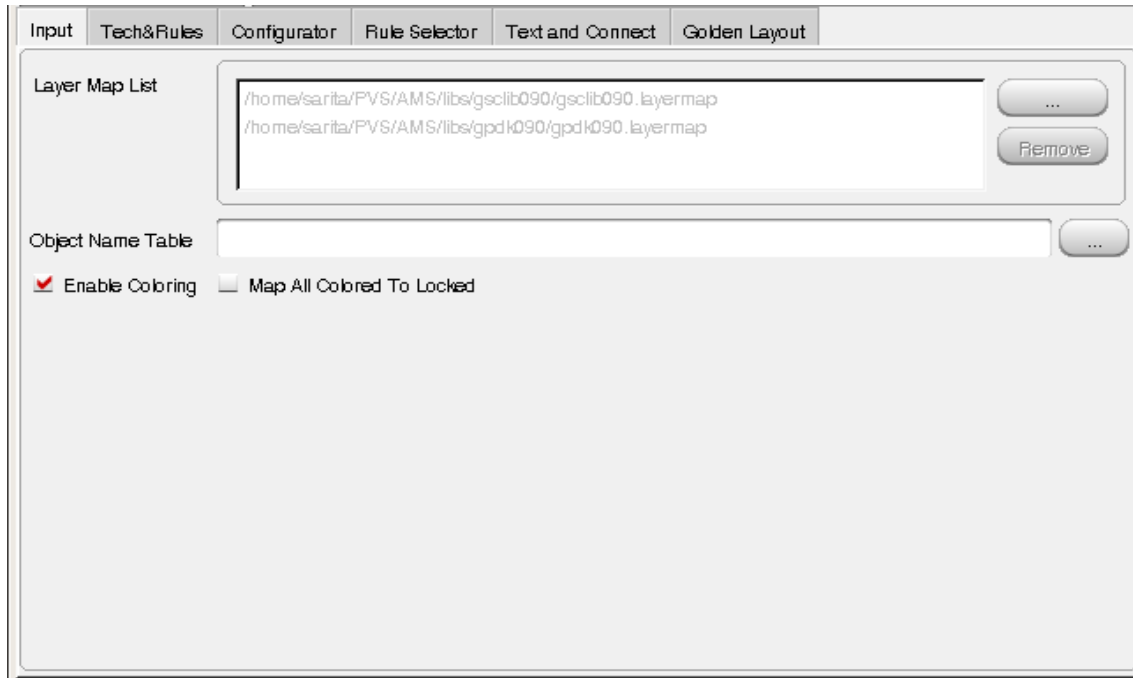
Right above the *Output* field, six tabs are available for customization. These options allow you to specify DRC criteria that will be stored in the snapshot.

- *Input*: Defines polygon handling information used between Virtuoso and PVS.
- *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 VIPVS to follow.
- *Golden Layout*: Defines Golden layout pattern files for graphical DRC.

Input

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

Figure 3-11 VIPVS Snapshot Creator: Input tab



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

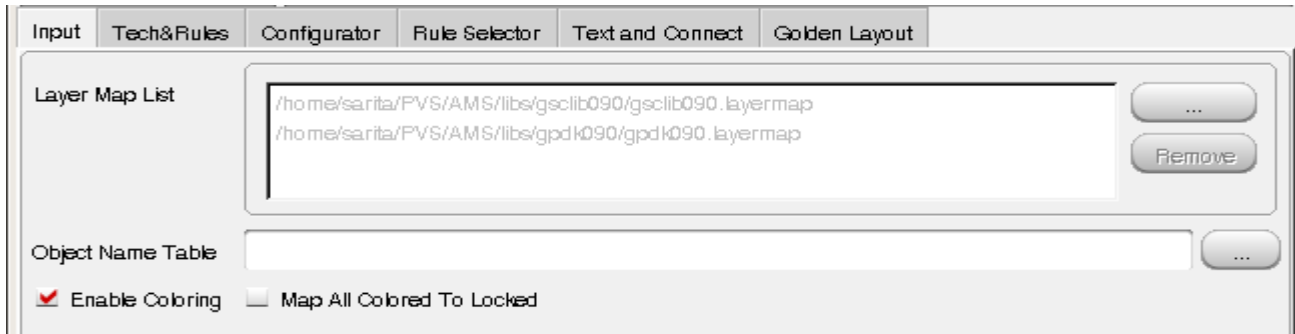
Similarly, you can specify the object mapping file by either of the following ways:

- Entering file path in the *Object Name Table* text box
- Clicking the (...) button and browse to the location to select it.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-12 Layer mapping file setting and related options



Since layermap file is one of required inputs, you can specify its initial value in the *Layer Map List* text box through the `Virtuoso_IPVS_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.

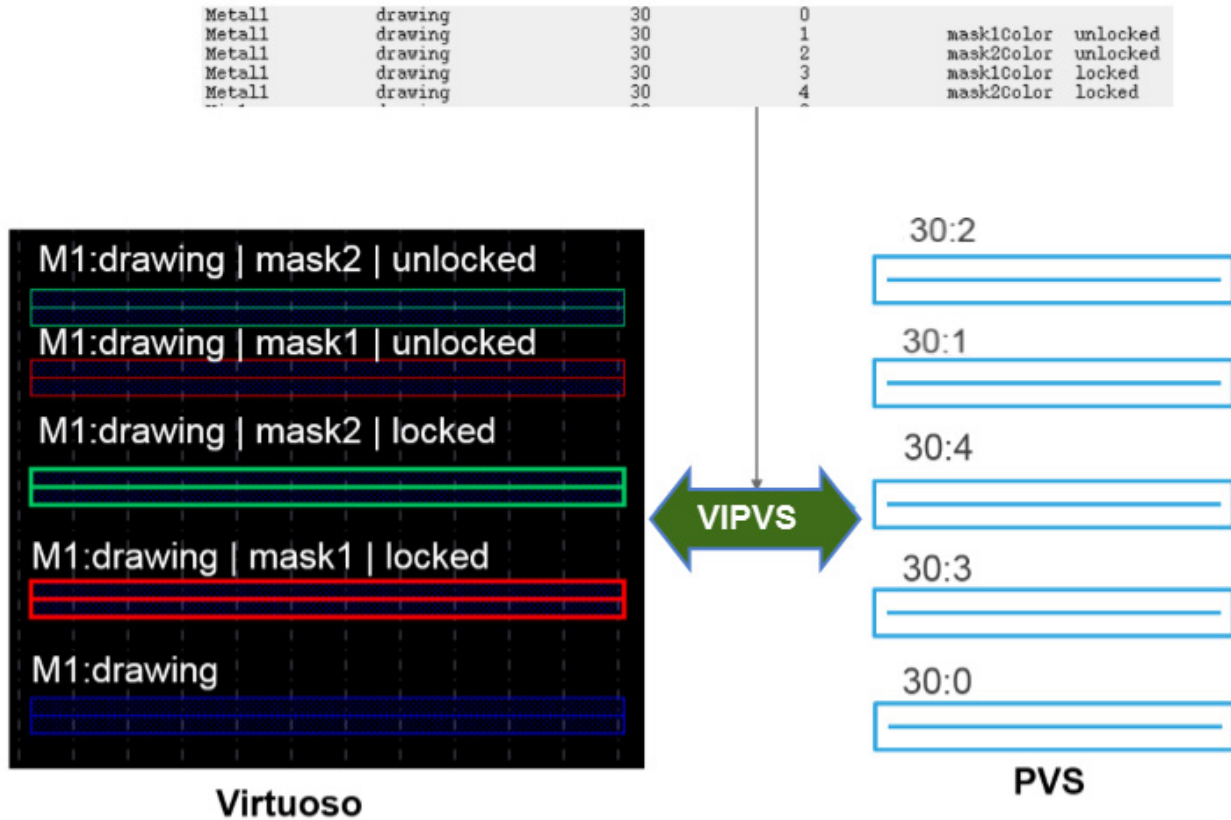
Depending on the version of Virtuoso release, *Enable Coloring* and *Map All Colored To Locked* options are available to specify color mapping criteria.

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, VIPVS 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 PVS prior to DRC. Similarly data preparation is done after DRC during reporting violations transport from PVS to Virtuoso. Following example shows how VIPVS refers layer mapping definition to translate data between Virtuoso and PVS.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-13 VIPVS data preparation between Virtuoso and PVS



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, VIPVS 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 `Virtuoso_IPVS_hideMapAllColorToLocked N` to make it visible on the form.

The priority of controls is the following:

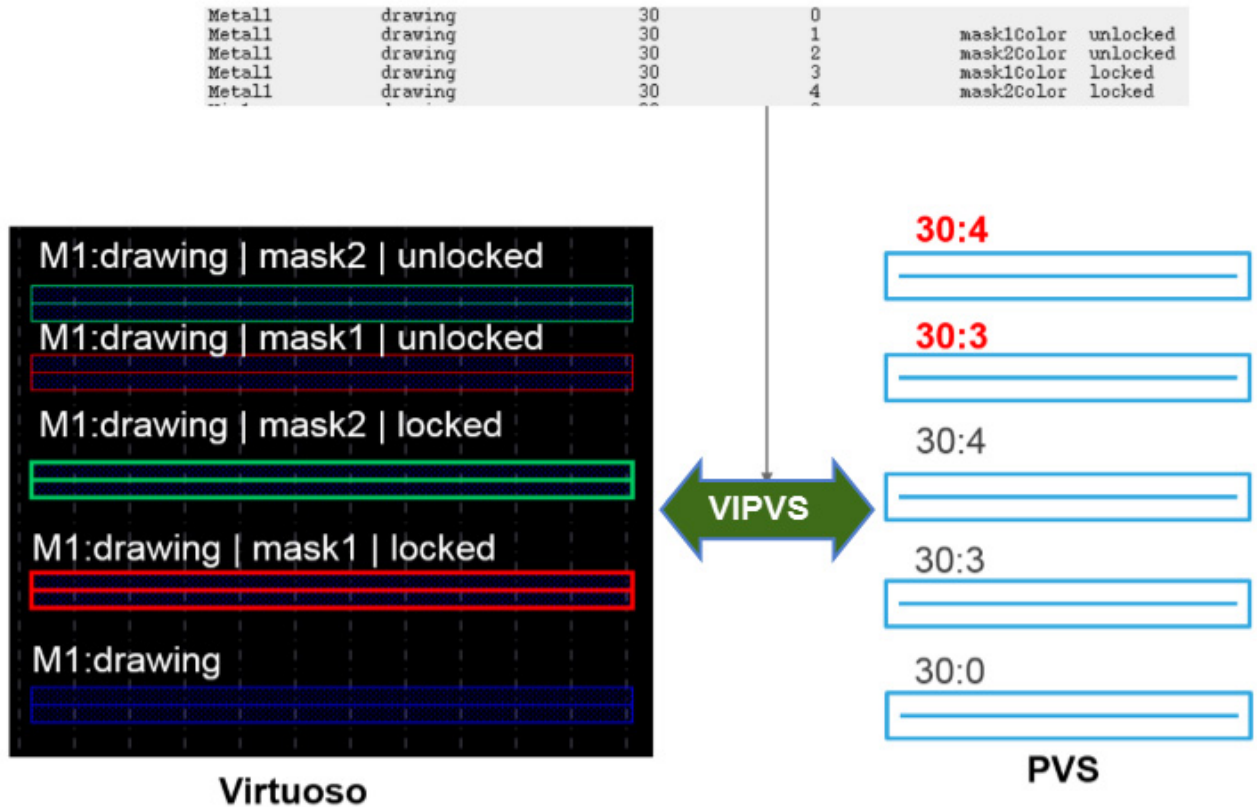
1. `envGetVal("xstream" "xstHideMapAllColorToLocked")` if defined.
2. If not defined, shell envvar `"Virtuoso_IPVS_hideMapAllColorToLocked"`.
3. If both are not defined - `default:hide`.

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

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-14 VIPVS data translation when "Map All Colored To Locked" is on



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:

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-15 VIPVS Snapshot Creator: Tech & Rules tab via rule deck

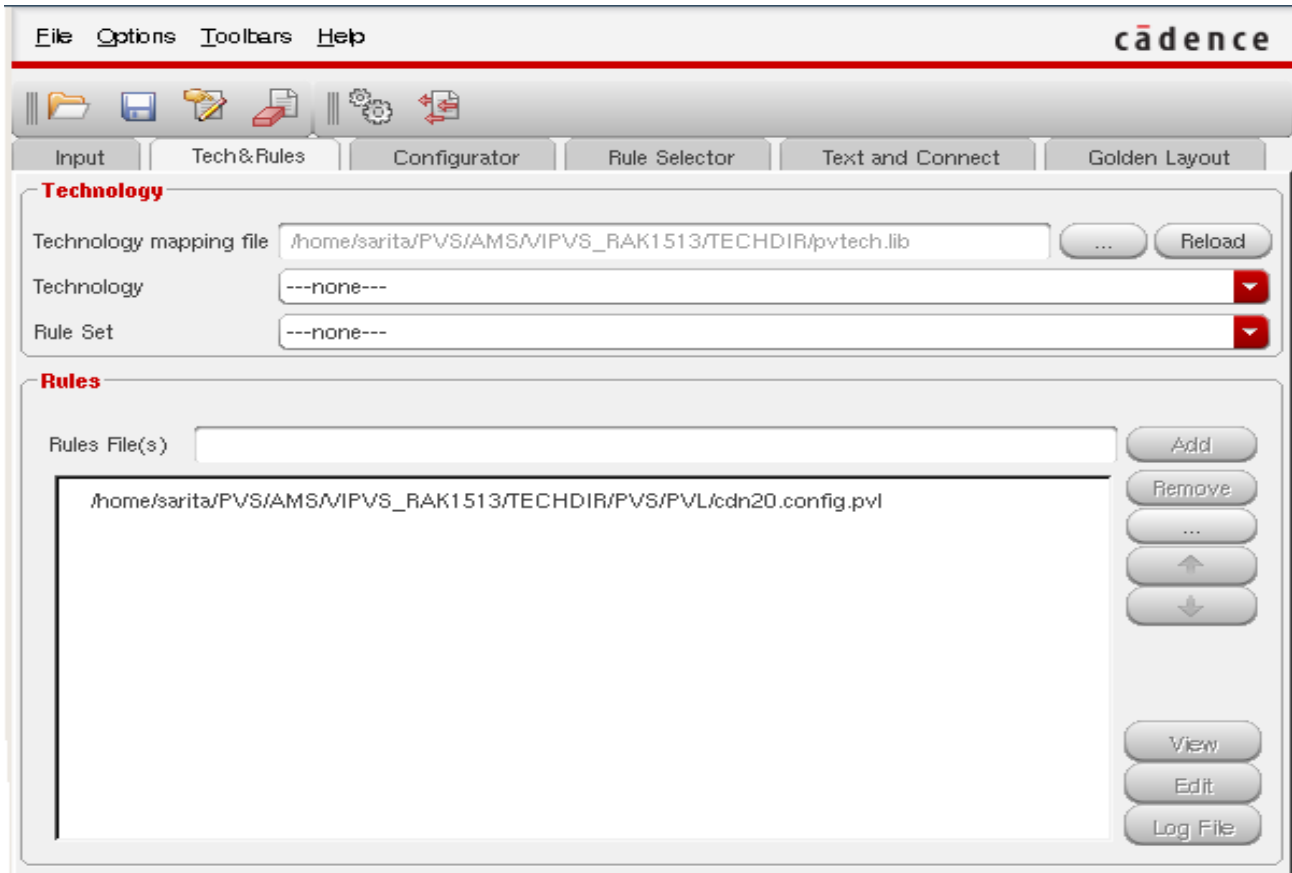
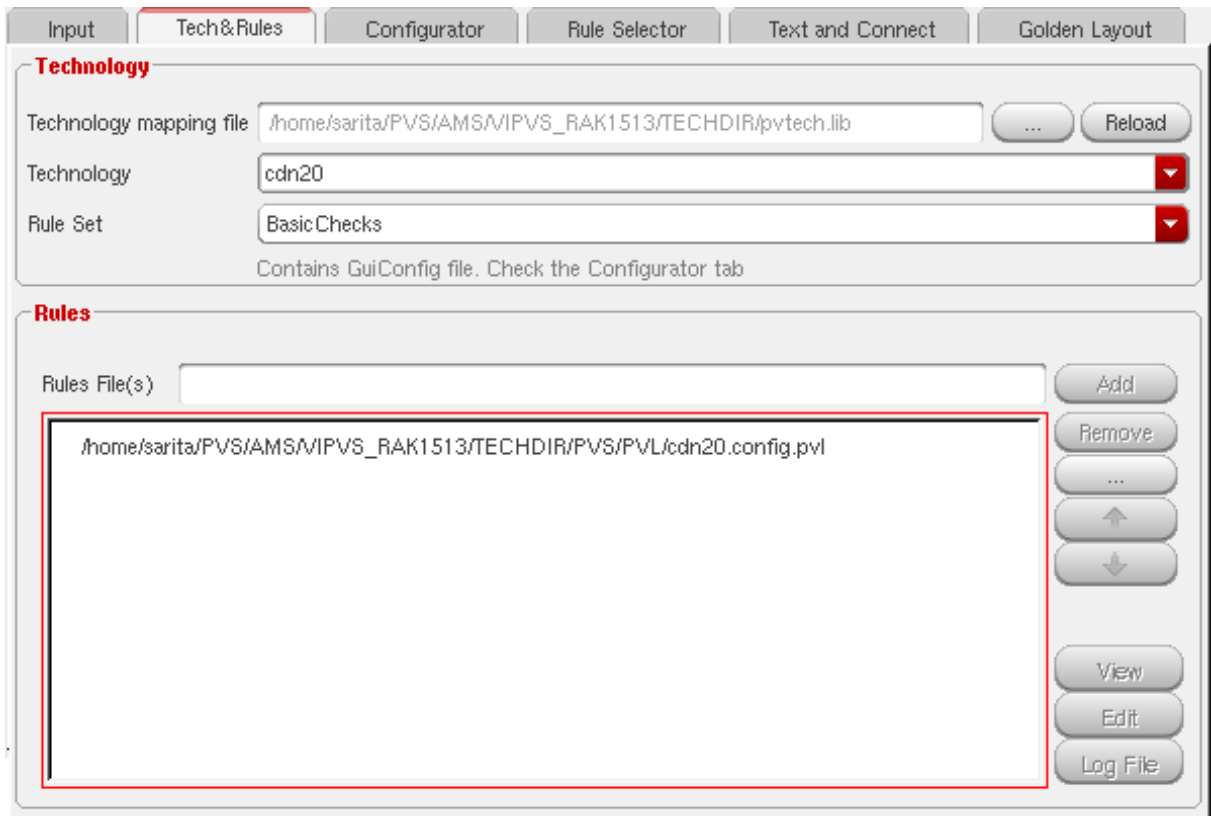


Figure 3-16 VIPVS Snapshot Creator: Tech & Rule Tab via Technology



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 occurrences 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 `Virtuoso_IPVS_defaultSNPRuleDeck` shell variable prior to Virtuoso launch. For details of setting rule set using Technology, see *Appendix C - Handling Technology Data* in PVS in the PVS 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.

Figure 3-17 VIPVS Snapshot Creator: Enable Configurator Flow



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.

Figure 3-18 VIPVS Snapshot Creator Flow: Configurator Enablement

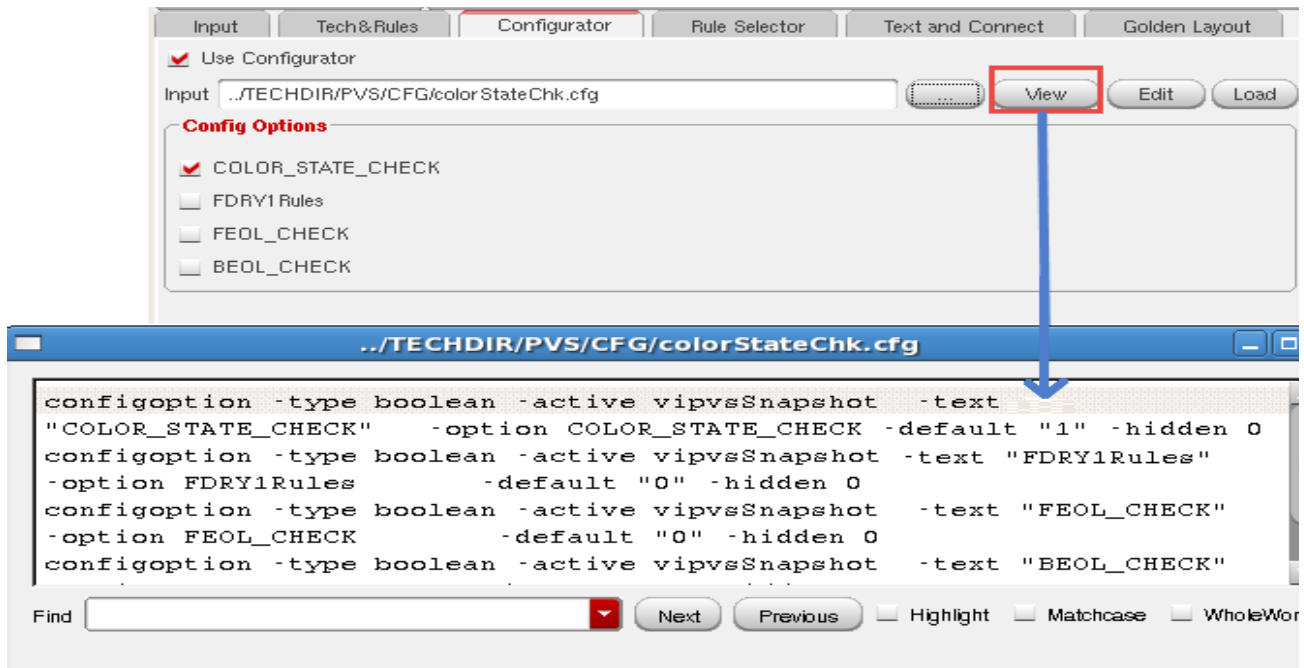


Virtuoso IPVS User Guide

All about Design Rules: Snapshot

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.

Figure 3-19 Snapshot Creator Flow: view configuration file



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

Generally, Configuration file is maintained globally by CAD. CAD can declare its initial value in the *Input* field through the `Virtuoso_IPVS_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 PVS. 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*.

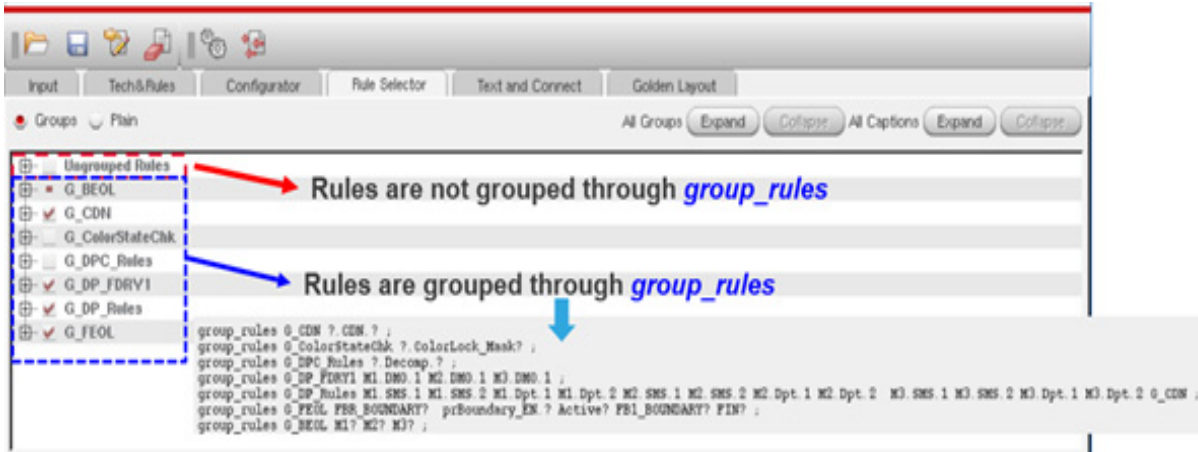
The default view mode is *Groups*. In this default view mode, rules are displayed either by the group they belong to or by *Ungrouped Rules*. Following example shows how groups'

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

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.

Figure 3-20 Rule display reflect rule grouping definition 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

Figure 3-21 Rule contents display control

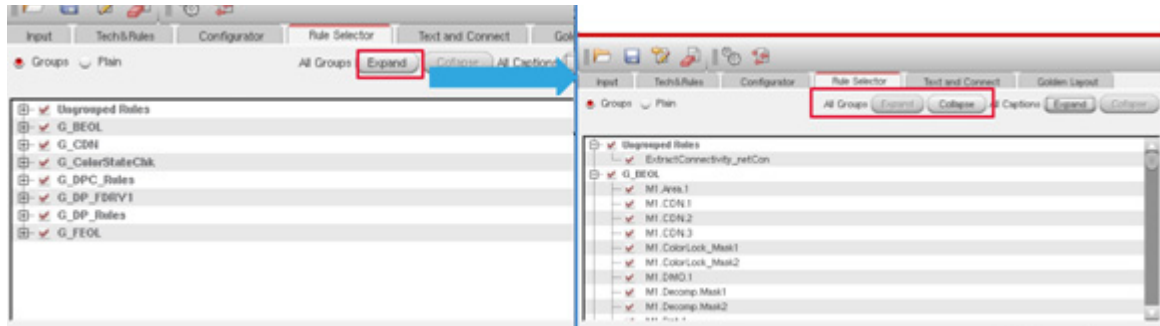


Following example shows what happens when you click the *All Groups: Expand* button. All the rules that belong to a group are displayed.

Virtuoso IPVS User Guide

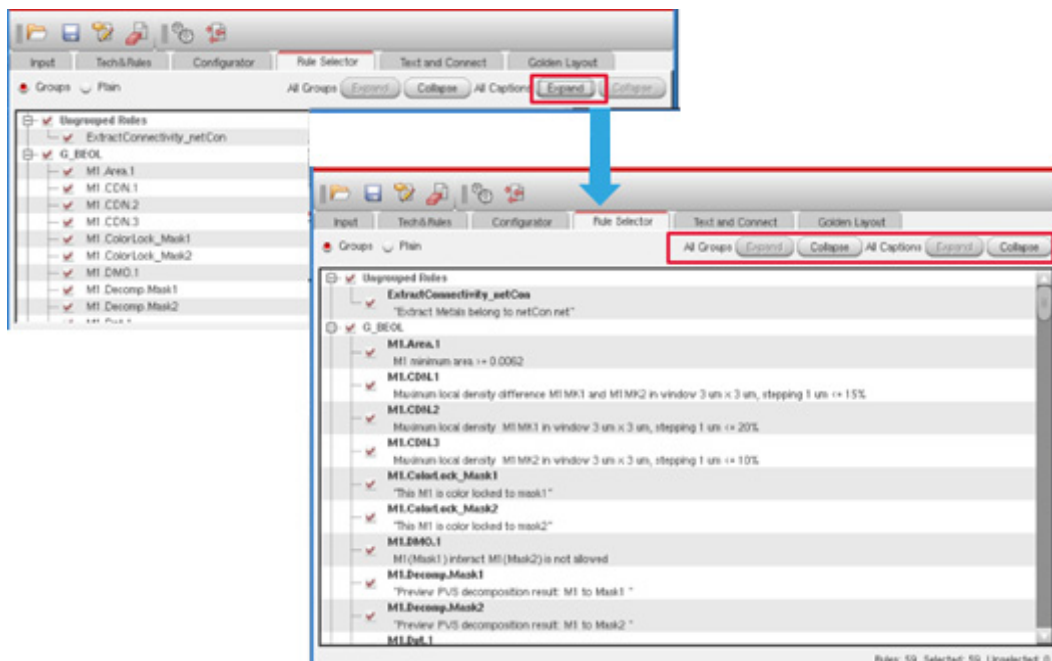
All about Design Rules: Snapshot

Figure 3-22 Expand content of all groups



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.

Figure 3-23 Expand content of all captions



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.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-24 Plain View: display rules based on its 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*.

Figure 3-25 Display rules in Plain view mode

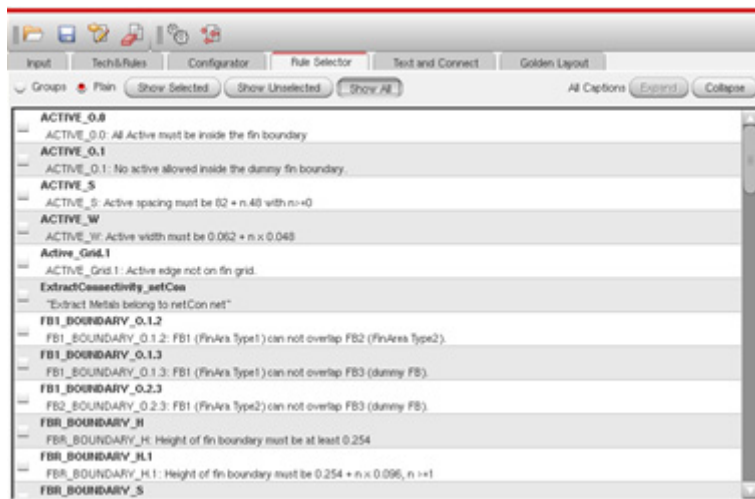
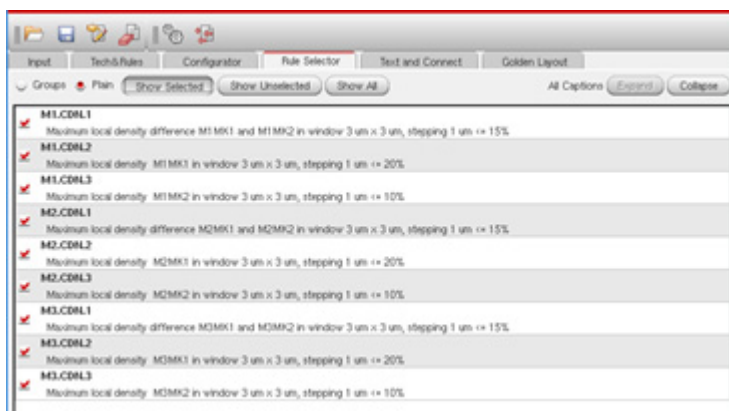


Figure 3-26 Display selected only rules in Plain view mode



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.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

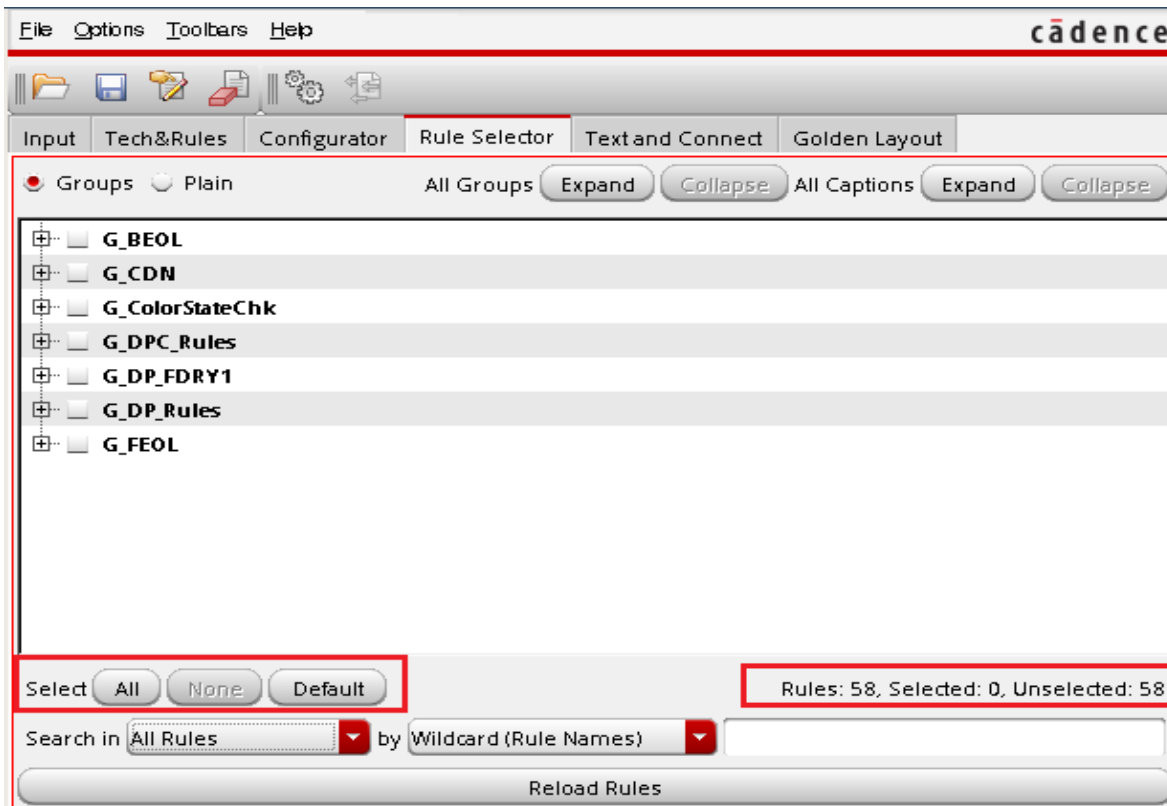
Figure 3-27 Rule select utility overview



Search based Rule Select

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

Figure 3-28 Deselect all of rules

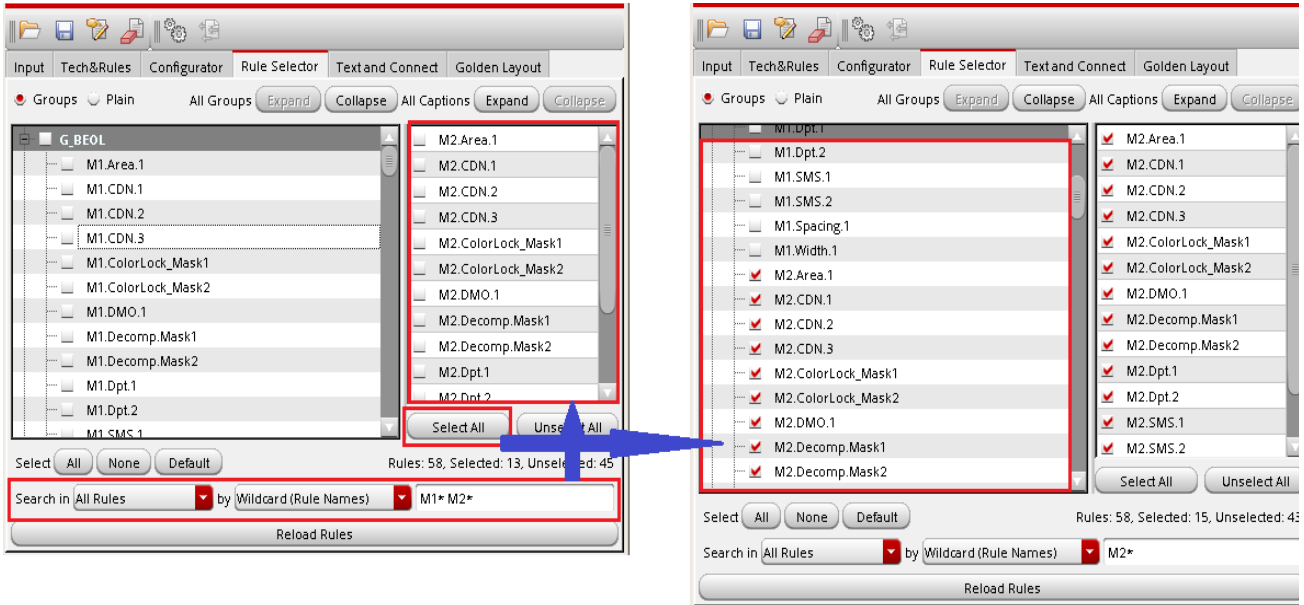


For search based rule selection, you can first set the search filter and then further define searching criteria by declaring searching condition in the entry field. Once matched rule names or captions are found, they can be pre-viewed. You can then perform rule selection by clicking the *Select* or *Unselect* buttons. The preview column is only displayed where there is searching condition declared in the entry field.

Virtuoso IPVS User Guide

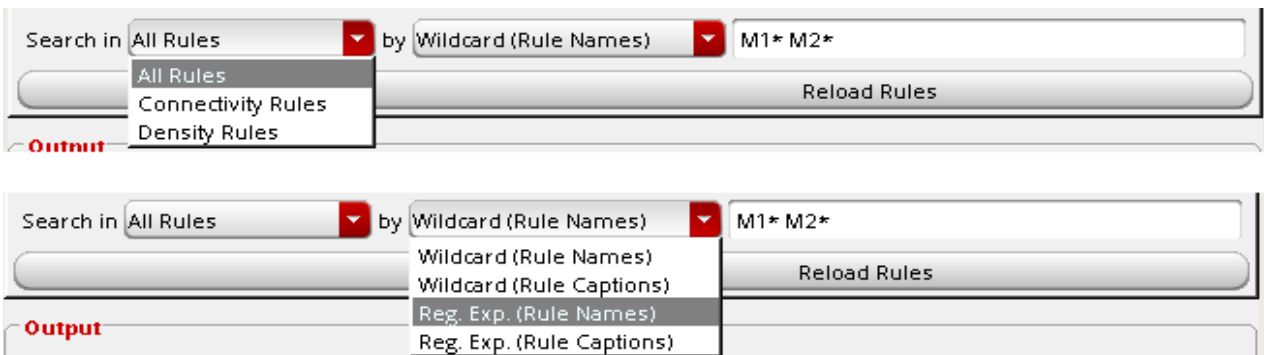
All about Design Rules: Snapshot

Figure 3-29 Search based rule selection flow



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)*.

Figure 3-30 Available pre-defined search filters



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

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

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.

Figure 3-31 Search rule caption matching searching criteria

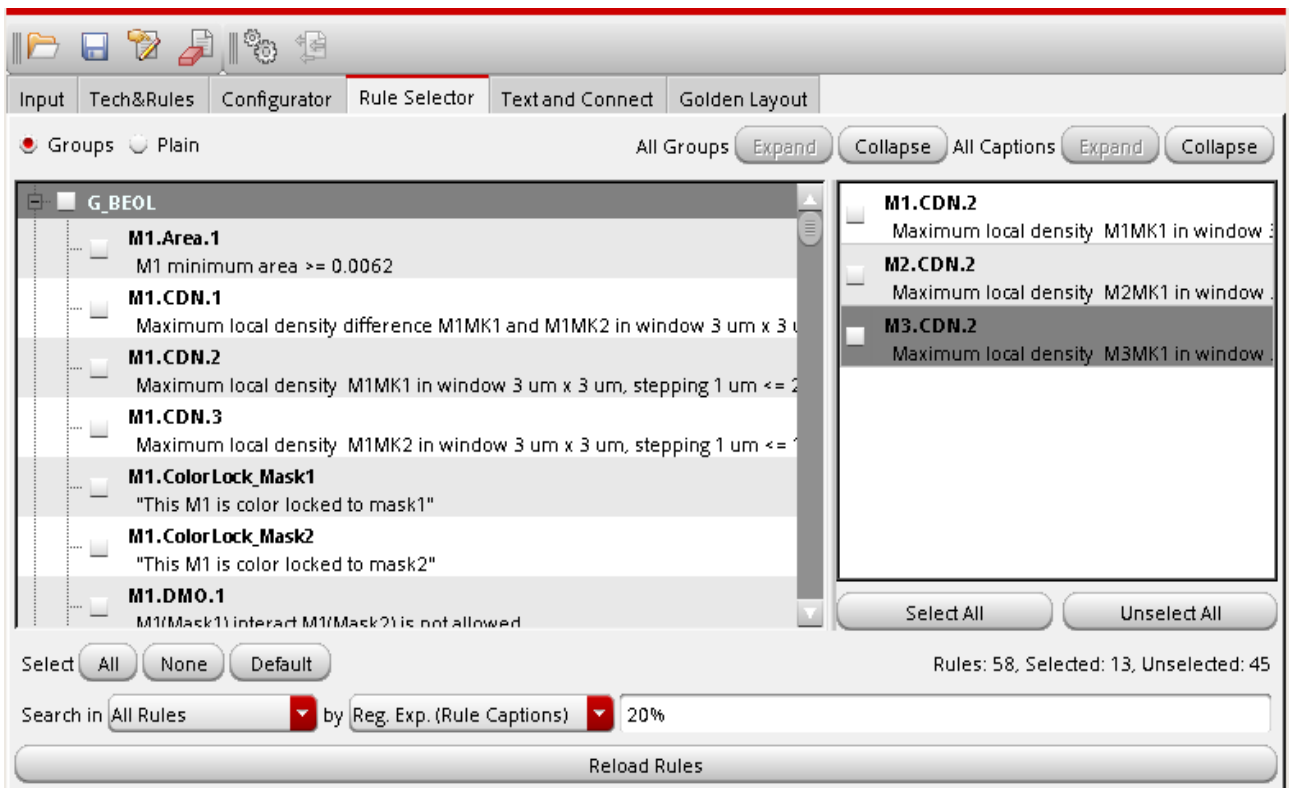
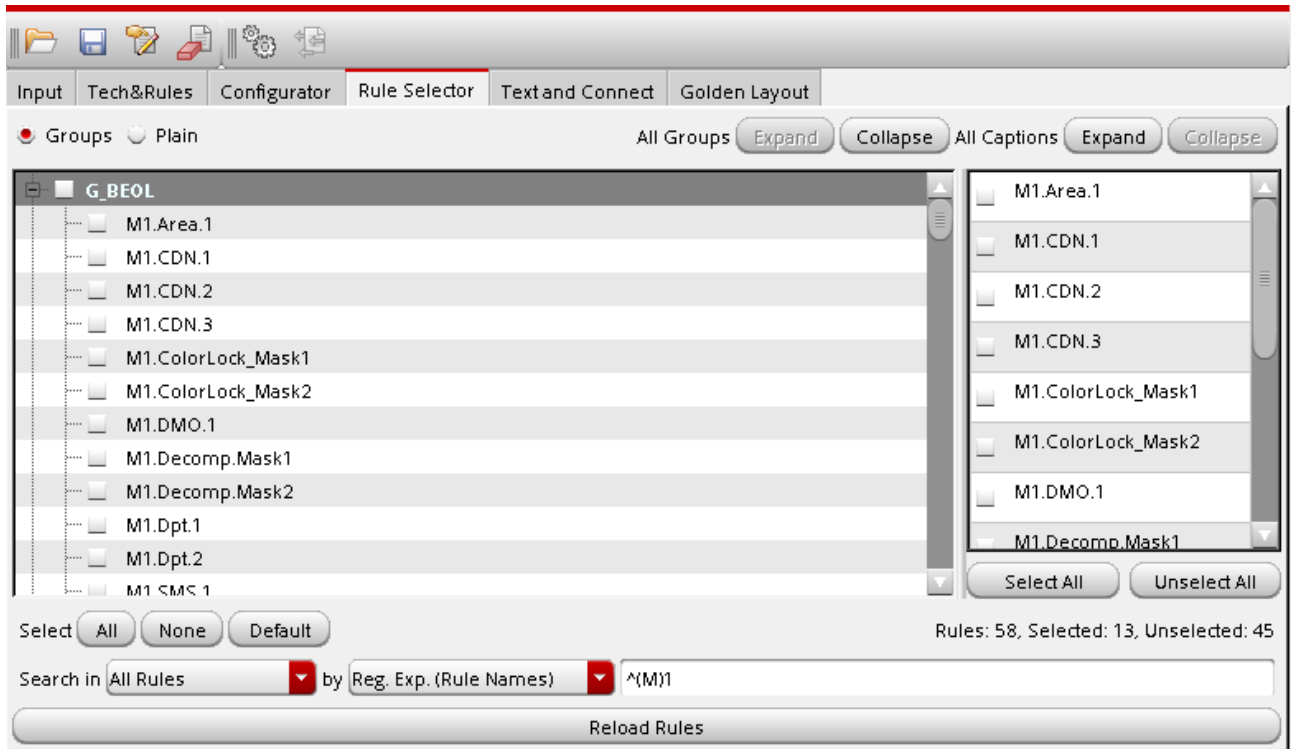


Figure 3-32 Search rule name through regex syntax



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.

Figure 3-33 Recompile rule content

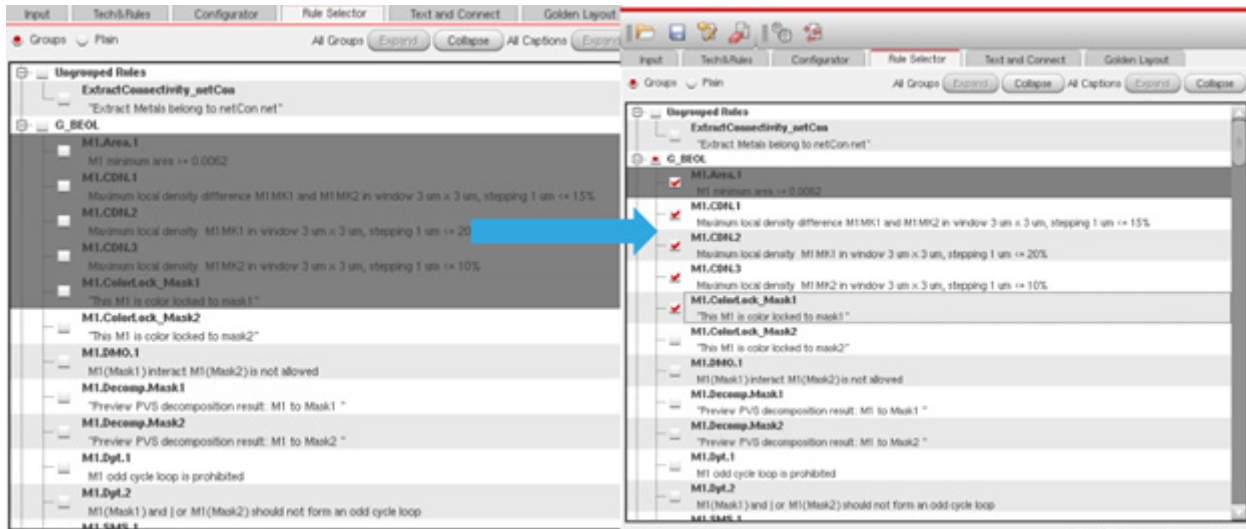


Finally, you can interactively select rules through the *rule display* field by clicking the check box located in the front of rule names or group names.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-34 Interactive rule selection



Text and Connect

For detailed usages of this tab, refer to the *Text and Connect Tab* section of *DRC Run Submission Form - Virtuoso in PVS User Guide*.

Golden Layout

For detailed usages of this tab, refer to the *Additional Layout Tab* section of *DRC Run Submission Form - Virtuoso in PVS User Guide*.

Snapshot Output Field

Figure 3-35 Snapshot Output Field

Output

Snapshot Name:

Snapshot Comment:

Snapshot Path:

Snapshot Type:

Set Custom Bloat Value:

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

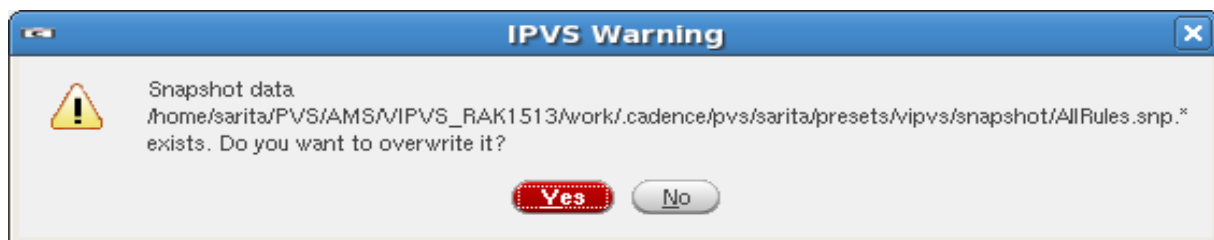
In the *Snapshot Name* field, you can enter the snapshot name. The naming syntax should not include any special characters, SPACE or TAB. If any special character is detected in the snapshot name input, following pop-up message box appears and requests you to modify the input.

Figure 3-36 Error message on detecting special characters declared in Snapshot Name



If the snapshot name you have defined already exists, the following message is displayed:

Figure 3-37 Snapshot Name overwrite message



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 `PVS_SNAP_SHOTS_DIRS`. If

`PVS_SNAP_SHOTS_DIRS` 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 `PVS_SNAP_SHOTS_DIRS`. Thus, the active snapshot directory path is the first item in the list. Also, path declared in `PVS_SNAP_SHOTS_DIRS` is expanded and displayed in its absolute path format.

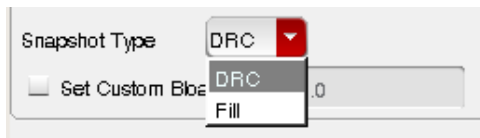
Following example shows all four paths declared in `PVS_SNAP_SHOTS_DIRS` and how they are displayed under the *Snapshot Path* drop-down list.

Figure 3-38 VIPVS acknowledged snapshot directories



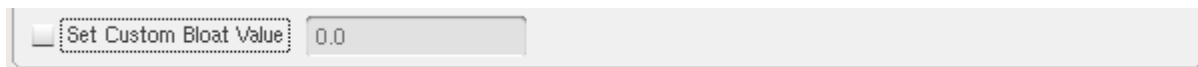
The *Snapshot Type* Drop-Down list allows you to select the type of snapshot you want to create. Select *DRC* to generate a snapshot of type DRC to perform in-design DRC checks and select *Fill* to generate a snapshot to perform signoff fills runs.

Figure 3-39 Snapshot Type



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 PVS 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.

Figure 3-40 Custom Bloat Value Setting



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 PVS 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

VIPVS allows you to create unlimited snapshots and store them in various snapshot directories. However, accessing snapshots can be cumbersome when multiple snapshots

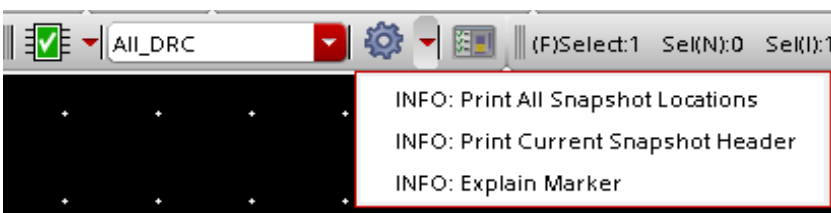
created using different sources are available. To effectively manage snapshots, following utilities are available:

- Snapshot information inquiry
- Snapshot display management

Snapshots Inquiry

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

Figure 3-41 Snapshot Info inquiry



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

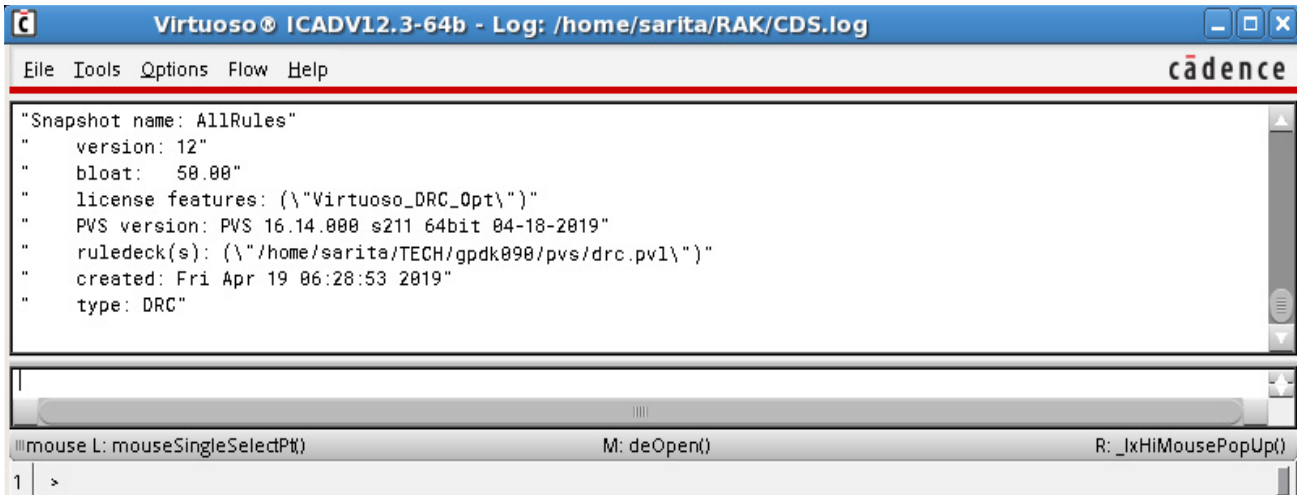
- INFO: Print All Snapshot Locations
- INFO: Print Current Snapshot Header
- INFO: Explain Marker

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

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-42 Snapshot header information in CIW



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

Virtuoso IPVS User Guide
All about Design Rules: Snapshot

Table 3-2 Snapshot Header Overview

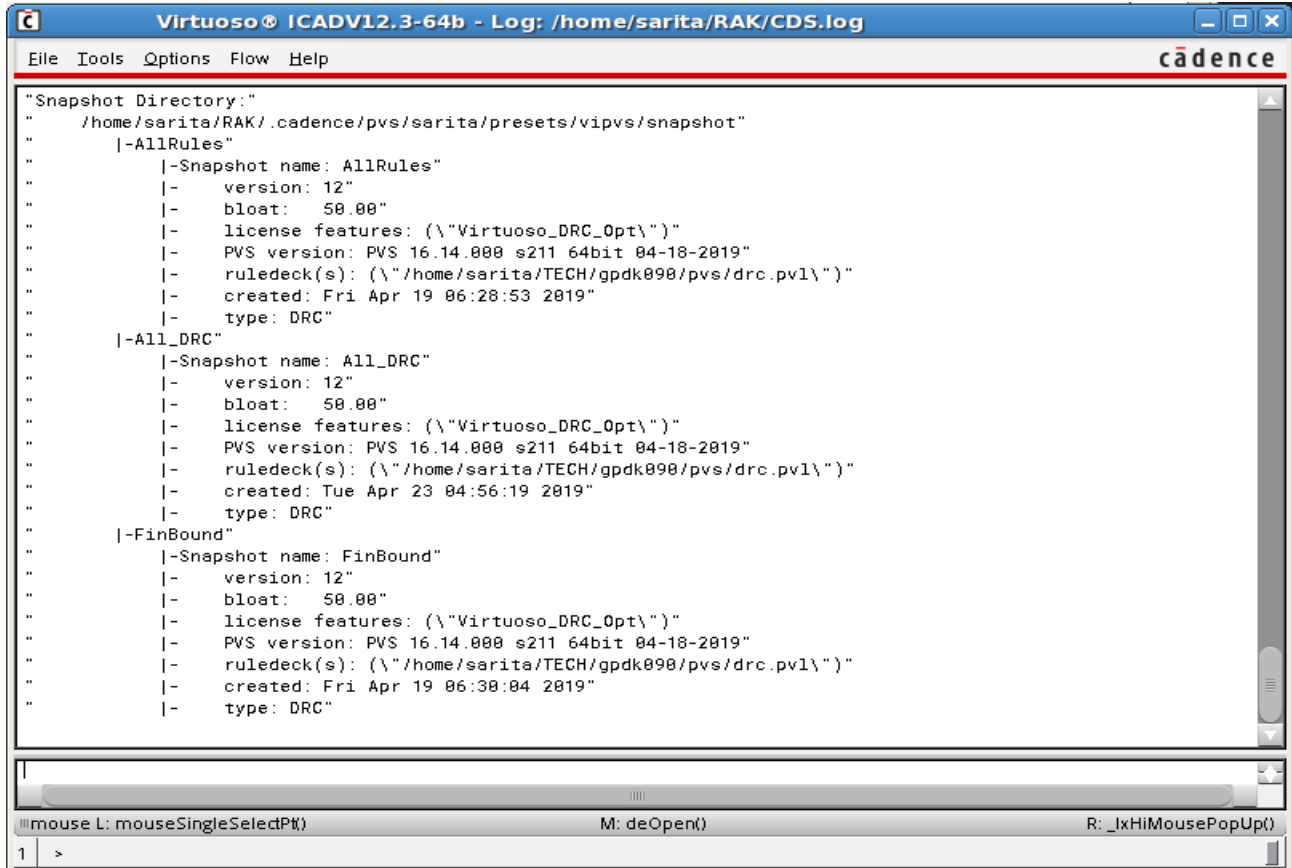
Header Info	Description
Snapshot Name	<ul style="list-style-type: none"> ■ Current active snapshot
Snapshot version	<ul style="list-style-type: none"> ■ Snapshot version issued by PVS ■ Version number is updated when PVS rule-deck to snapshot compiler engine supports new PVS command.
The bloat value	<ul style="list-style-type: none"> ■ Area expansion threshold in X and Y directions for PVS 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	<ul style="list-style-type: none"> ■ License required to run VIPVS for a given selected checks stored in the snapshot. ■ License issuing is dependent on the content of selected rule decks.
PVS Version	<ul style="list-style-type: none"> ■ PVS binary used to create this snapshot
Source Rule Deck	<ul style="list-style-type: none"> ■ The source rule deck(s) this snapshot is generated from
Date of creation	<ul style="list-style-type: none"> ■ Time stamp when snapshot is first created
type	<ul style="list-style-type: none"> ■ Type of snapshot: DRC or Fill.

You can print all available snapshots' header information and locations where they are stored into by invoking *INFO: Print All Snapshot Locations*. VIPVS composes a snapshot lookup map and populates this information in CIW. It is grouped by snapshot directory. Following figure shows lookup map outline.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-43 Snapshot lookup map



```
Virtuoso® ICADV12.3-64b - Log: /home/sarita/RAK/CDS.log
File Tools Options Flow Help
cadence

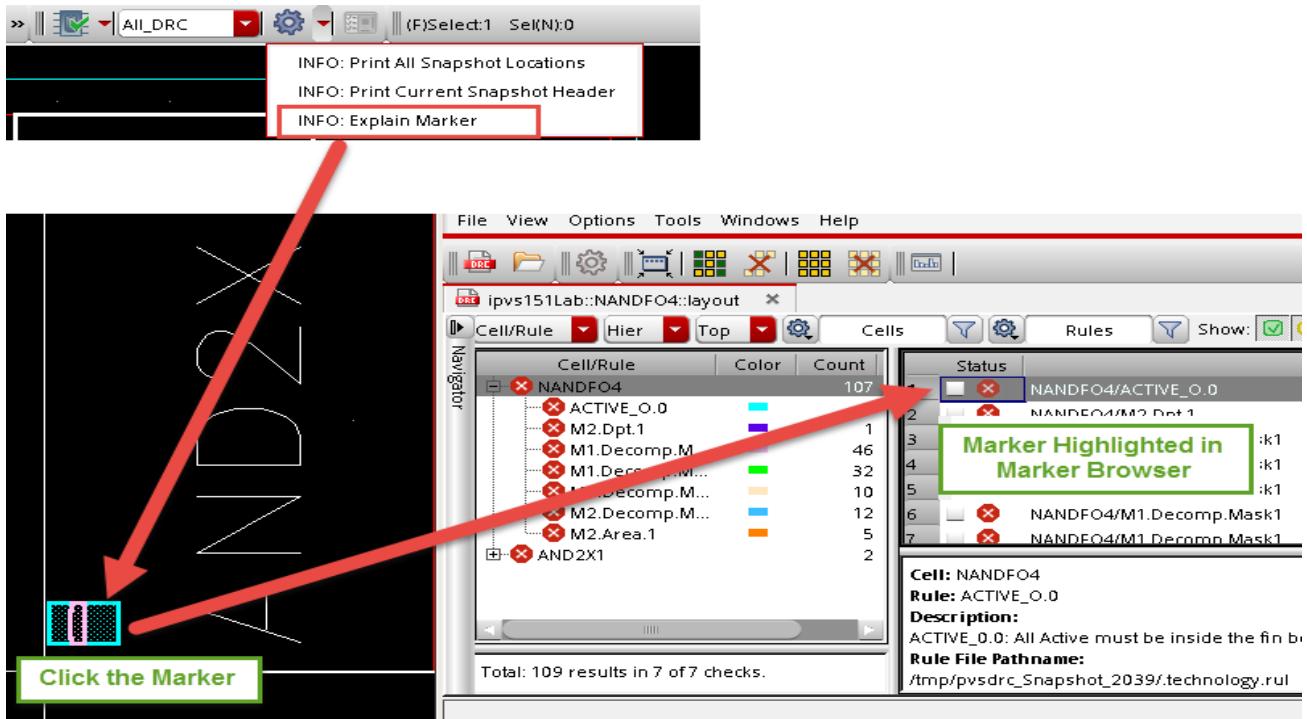
"Snapshot Directory:"
"/home/sarita/RAK/.cadence/pvs/sarita/presets/vipvs/snapshot"
|-AllRules"
  |-Snapshot name: AllRules"
  |- version: 12"
  |- bloat: 50.00"
  |- license features: (\\"Virtuoso_DRC_Opt\\")"
  |- PVS version: PVS 16.14.000 s211 64bit 04-18-2019"
  |- ruledeck(s): (\\"/home/sarita/TECH/gpdk090/pvs/drc.pv1\\")"
  |- created: Fri Apr 19 06:28:53 2019"
  |- type: DRC"
|-All_DRC"
  |-Snapshot name: All_DRC"
  |- version: 12"
  |- bloat: 50.00"
  |- license features: (\\"Virtuoso_DRC_Opt\\")"
  |- PVS version: PVS 16.14.000 s211 64bit 04-18-2019"
  |- ruledeck(s): (\\"/home/sarita/TECH/gpdk090/pvs/drc.pv1\\")"
  |- created: Tue Apr 23 04:56:19 2019"
  |- type: DRC"
|-FinBound"
  |-Snapshot name: FinBound"
  |- version: 12"
  |- bloat: 50.00"
  |- license features: (\\"Virtuoso_DRC_Opt\\")"
  |- PVS version: PVS 16.14.000 s211 64bit 04-18-2019"
  |- ruledeck(s): (\\"/home/sarita/TECH/gpdk090/pvs/drc.pv1\\")"
  |- created: Fri Apr 19 06:38:04 2019"
  |- type: DRC"

||mouse L: mouseSingleSelectPt() M: deOpen() R: _JxHiMousePopUp()
1 >
```

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

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 DRC Results Viewer as browser.

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

Snapshots Management

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, VIPVS 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 *Virtuoso IPVS Run Options*. You can enable it through `Virtuoso_IPVS_enableSNPManagement`.

Clicking the *Manage* button opens the *Virtuoso IPVS Snapshot Manager* form:

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-44 Virtuoso IPVS Snapshot Manager form



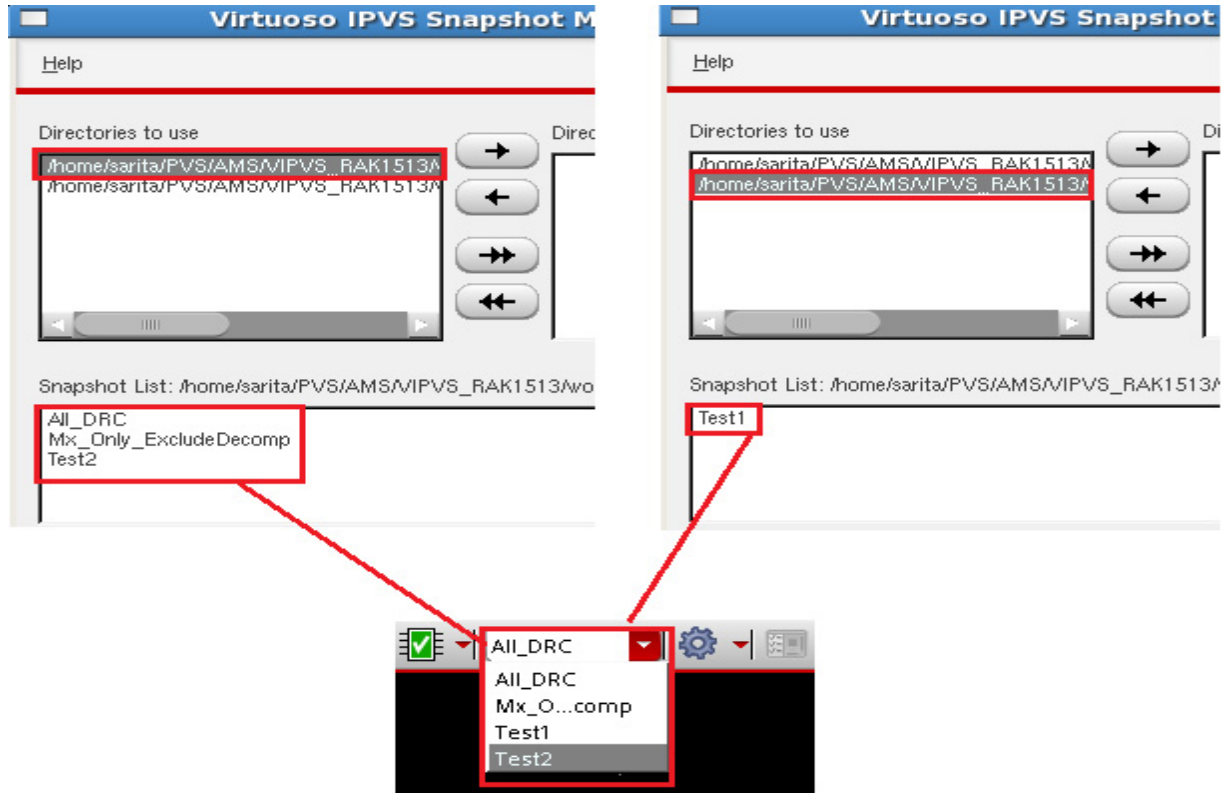
The form lists all snapshot directories defined in environment variable `PVS_SNAP_SHOTS_DIRS`. If `PVS_SNAP_SHOTS_DIRS` is undefined, then the form lists default snapshots directory `<$PWD>/ .cadence/pvs/<$userid>/presets/vipvs/snapshot`.

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.

Virtuoso IPVS User Guide

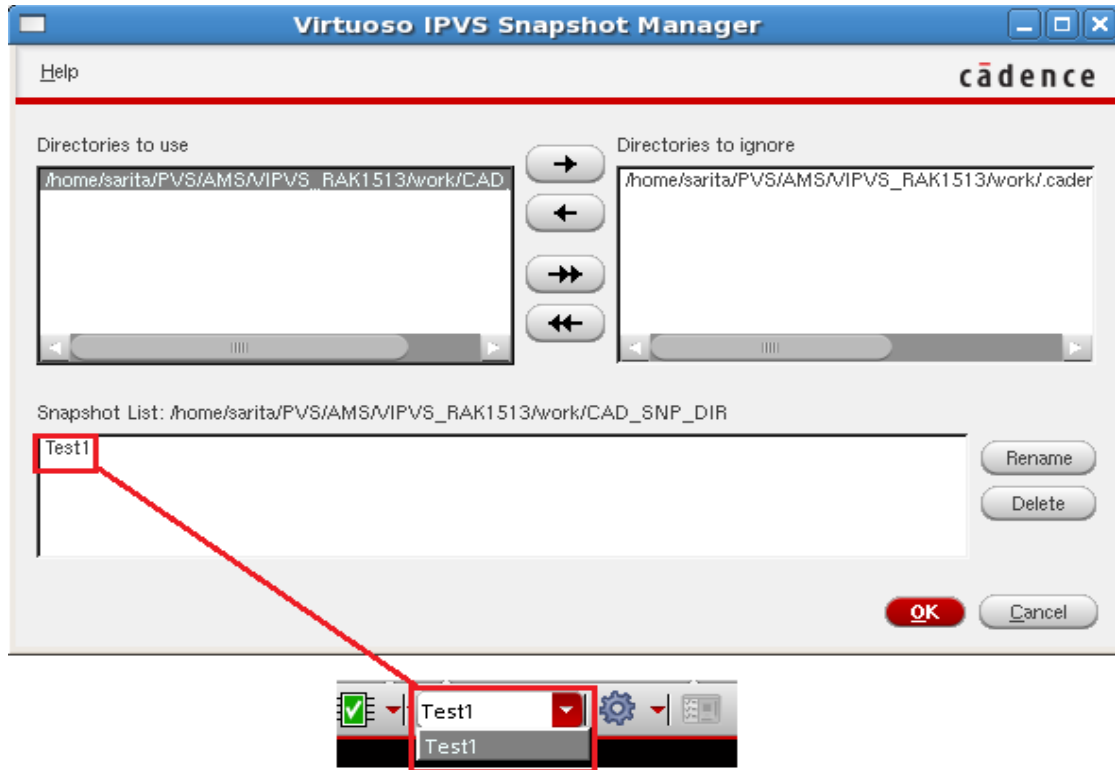
All about Design Rules: Snapshot

Figure 3-45 Manage Snapshots



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:

Figure 3-46 Snapshot display based on selective directory



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.

Snapshots Upgrade

PVS 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, VIPVS still checks snapshot readability during layout opening. If the incompatibility is detected, it outputs following warning messages in CIW:

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

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

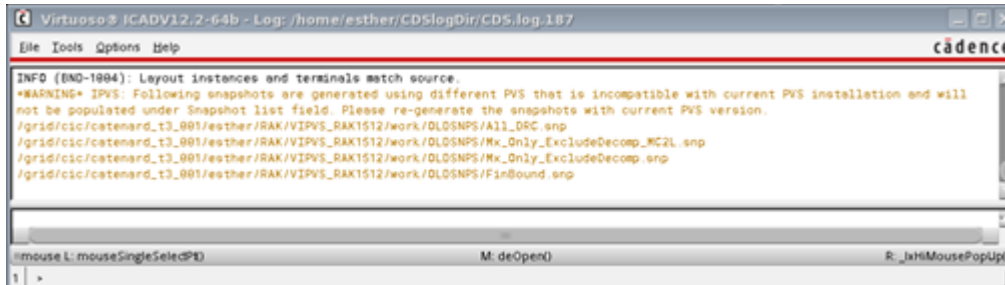
Note: PVS rule parser may be changed in ISR or hot fix to achieve better results. In this case snapshot version will be changed. New VIPVS version will read old snapshots generated by

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

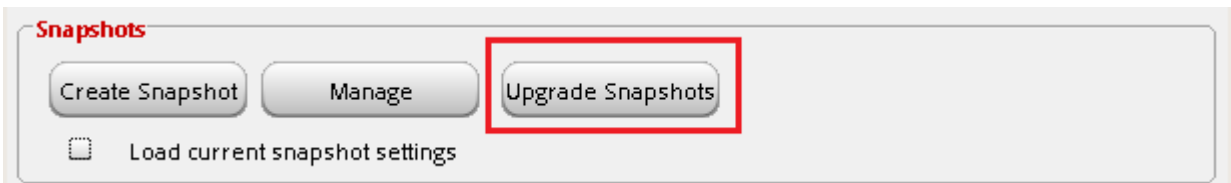
previous VIPVS versions of the same base release. However, changes in rule parser will not be visible. To take advantage of new PVS rule parser, snapshot need to be upgraded.

Figure 3-47 VIPVS Incompatibility Check



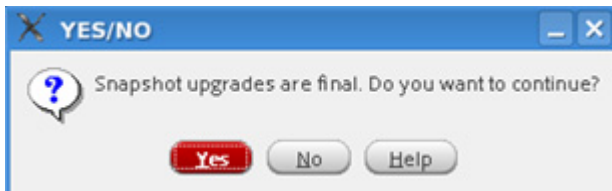
You can invoke *Upgrade Snapshot* utility that runs a batch upgrade on all incompatible snapshots. The *Upgrade Snapshot* button in the *Snapshots* section of the *Virtuoso IPVS Run Options* form is active if there are incompatible snapshots in snapshots directory. You can make it available through `Virtuoso_IPVS_maintenance`.

Figure 3-48 Upgrade Snapshots



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

Figure 3-49 Precede with snapshot upgrade



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, VIPVS precedes with snapshot upgrade and upgrade status is printed out in CIW. During the time of snapshot batch upgrade, Virtuoso is not accessible.

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

Figure 3-50 Snapshot upgrade status in CIW



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

Figure 3-51 Snapshot Batch Upgrade



Portability of Snapshots

VIPVS 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, VIPVS 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 `Virtuoso_IPVS_data` to make snapshot portable between different location. This is because VIPVS snapshot stores absolute paths which may be not visible from other locations.

Example:

```
> setenv PVS_DK /usr/dk/pvs_data
```

Rule deck contains line:

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

In the VIPVS snapshot these lines will be stored as:

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

Thus, if path `/usr/dk/pvs_data/GOLDEN_GDS/` is not visible from end-user location, VIPVS 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 Virtuoso_IPVS_data /usr/dk/pvs_data`

- b. Generate snapshot. Paths to golden GDS files will be stored as:

```
${Virtuoso_IPVS_data}/GOLDEN_GDS/golden1.gds.  
${Virtuoso_IPVS_data}/GOLDEN_GDS/golden2.gds.
```

2. End-user:

- a. Install design kit in local directory. For example:

```
/usr/user/pvs_data
```

- b. `setenv Virtuoso_IPVS_data /usr/user/pvs_data`

- c. Run VIPVS.

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 `VIPVSValidateSnapshot` 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 `VIPVSValidateSnapshot(snp_name snp_dir snp_comment)` returns:

a. `t`

- i. Snapshot name will show up on VIPVS Toolbar
- ii. There will be no message in CIW

b. `nil`

- i. Snapshot name will not show up on VIPVS Toolbar
- ii. Following warning message will be shown in CIW:

```
VIPVS: 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 VIPVS Toolbar
- ii. Following warning message will be shown in CIW:

```
"VIPVS: <error_message>"
```

d. Empty `error_message` (string)

- i. Snapshot name will not show up on VIPVS 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:

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

```
procedure( VIPVSValidateSnapshot( snp_name snp_dir snp_comment)
  let((project_dir valid)
    project_dir = getShellEnvVar("PVS_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( VIPVSValidateSnapshot( snp_name snp_dir snp_comment)
  let((
    rexMatchp( " N[5-7]" snp_comment)
  );let
);procedure
```

Example

1. Create snapshot and add comment:

Output

Snapshot Name	default
Snapshot Comment	PDK v.4.21
Snapshot Path	/vols/pvs_pe/olegd/GUI/PVS_RV/PVS_ShortBrowser_LAB/pvs/.cadence/pvs/olegd/presets/vipv: ▼

2. Create another snapshot and add comment:

Output

Snapshot Name	default_old
Snapshot Comment	PDK v.4.19
Snapshot Path	/vols/pvs_pe/olegd/GUI/PVS_RV/PVS_ShortBrowser_LAB/pvs/.cadence/pvs/olegd/presets/vipv: ▼

Virtuoso IPVS User Guide

All about Design Rules: Snapshot

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( VIPVSValidateSnapshot( 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.

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 on page 104
 - Run Results on page 104
 - Violation Markers Handling on page 104
- Verify Design Mode on page 105
 - Current CellView on page 105
 - Visible Area on page 106
 - Changed Area on page 107

Overview: Run Behavior

Run Results

The run results are stored under `$PWD/PVS_RUNDIR_<VD, PE>`. 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 PVS sign-off run, VIPVS Verify Design only outputs following commands defined in the rule deck:

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

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

All run logs can be found by default under `$PWD/PVS_RUNDIR_<VD, PE>/<cellName>_<viewName>_<libraryName>/vipvs.log`.

You can change the default VIPVS run directories through `Virtuoso_IPVS_dir`.

Violation Markers Handling

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

Also, VIPVS 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 PVS DRCDE 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, DRCDE can populate the latest run results on demand. As long as ASCII results are accessible by DRCDE, you can invoke DRCDE on the toolbar and violation highlights are populated in the layout.

Verify Design Mode

This VIPVS 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:

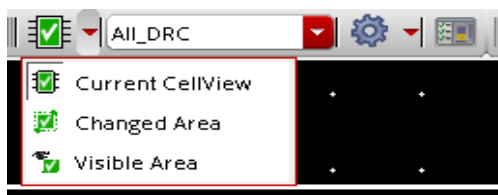
- Current CellView
- Changed Area
- Visible Area

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 VIPVS. 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 VIPVS completes DRC.

Regardless the hierarchy depth displayed in the layout, VIPVS 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 `Virtuoso_IPVS_initArea`.

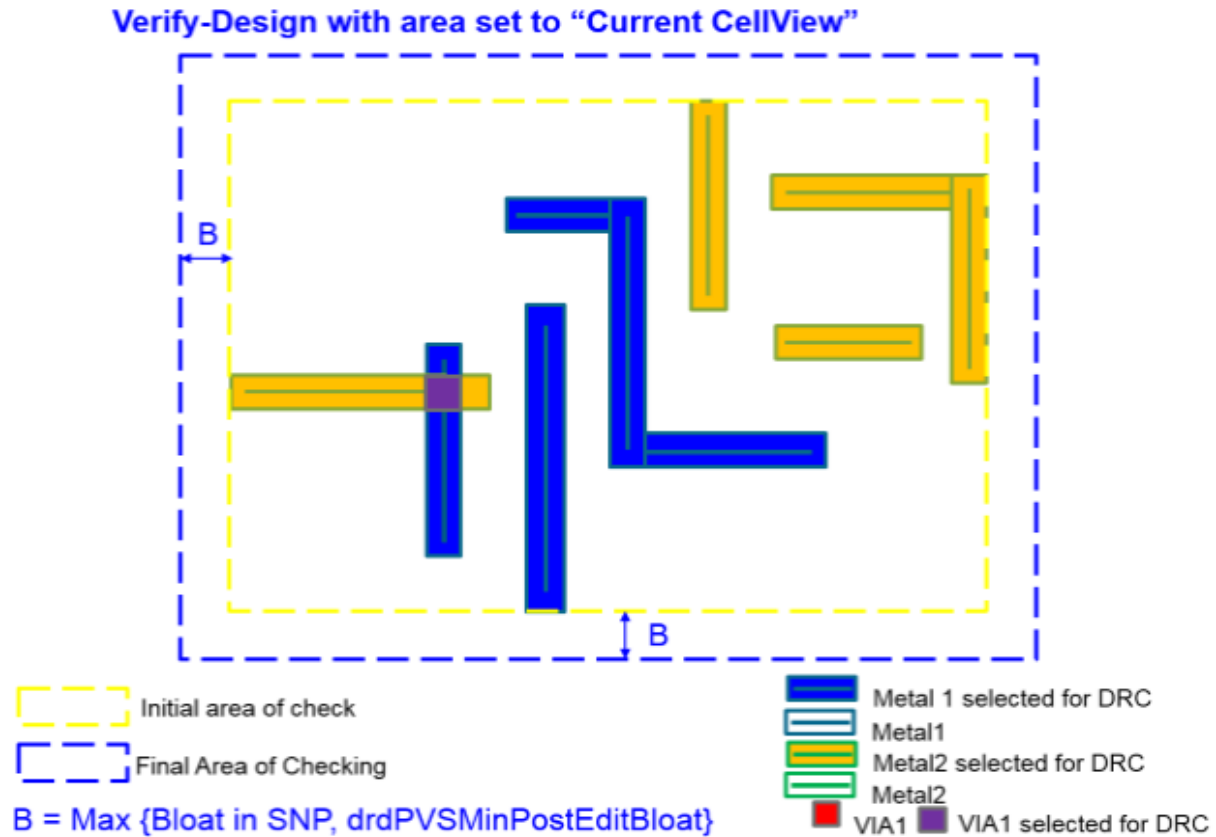
Figure 4-1 VIPVS Verify Design Mode area options



Current CellView

Verify-Design mode area checking is default to *Current CellView*. Regardless of current viewing area of the layout, VIPVS 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.

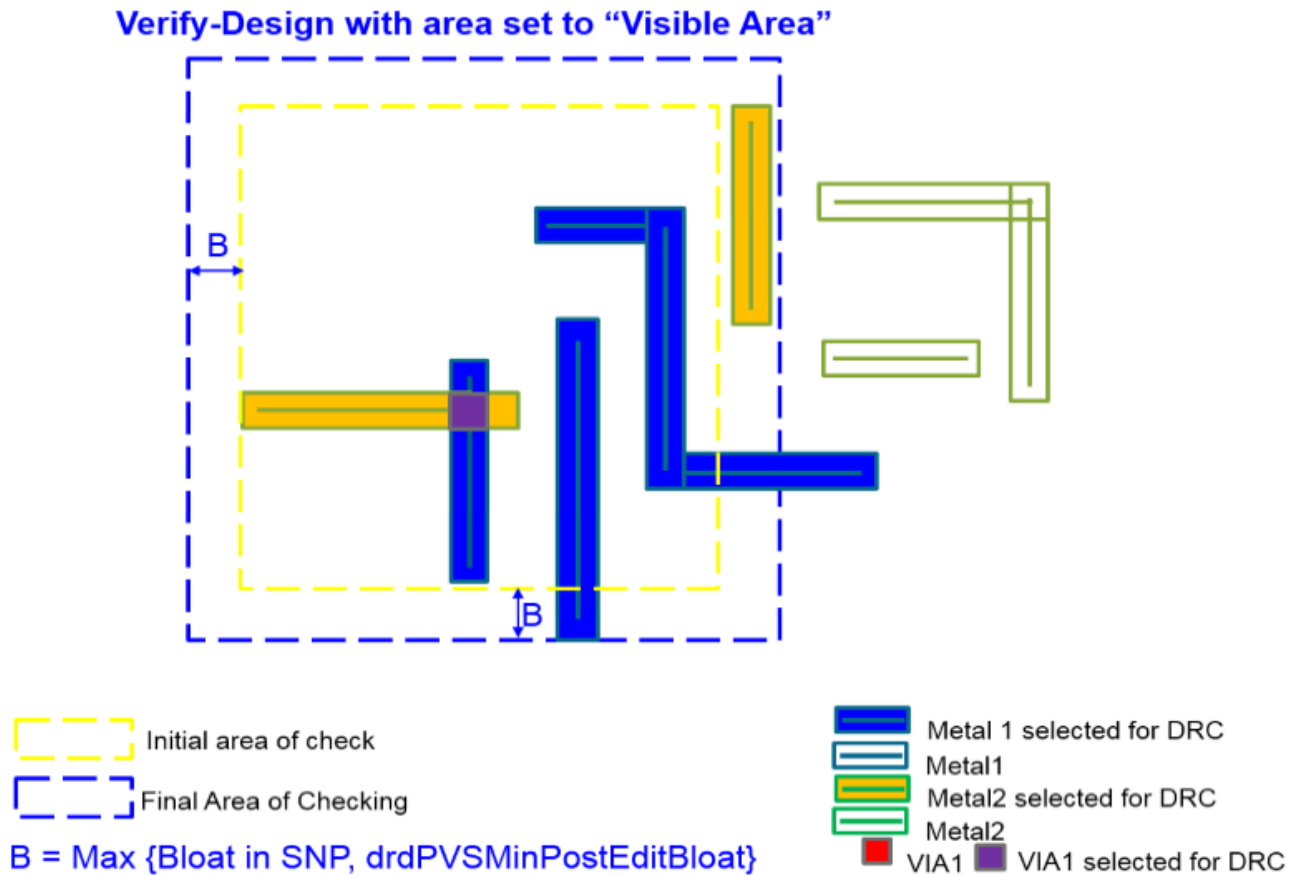
Figure 4-2 Final Area of Checking when Area sets to Current CellView



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. VIPVS then computes final area of checking by factoring bloat value defined in the current active snapshot.

Figure 4-3 Final Area of Checking when Area sets to Visible Area

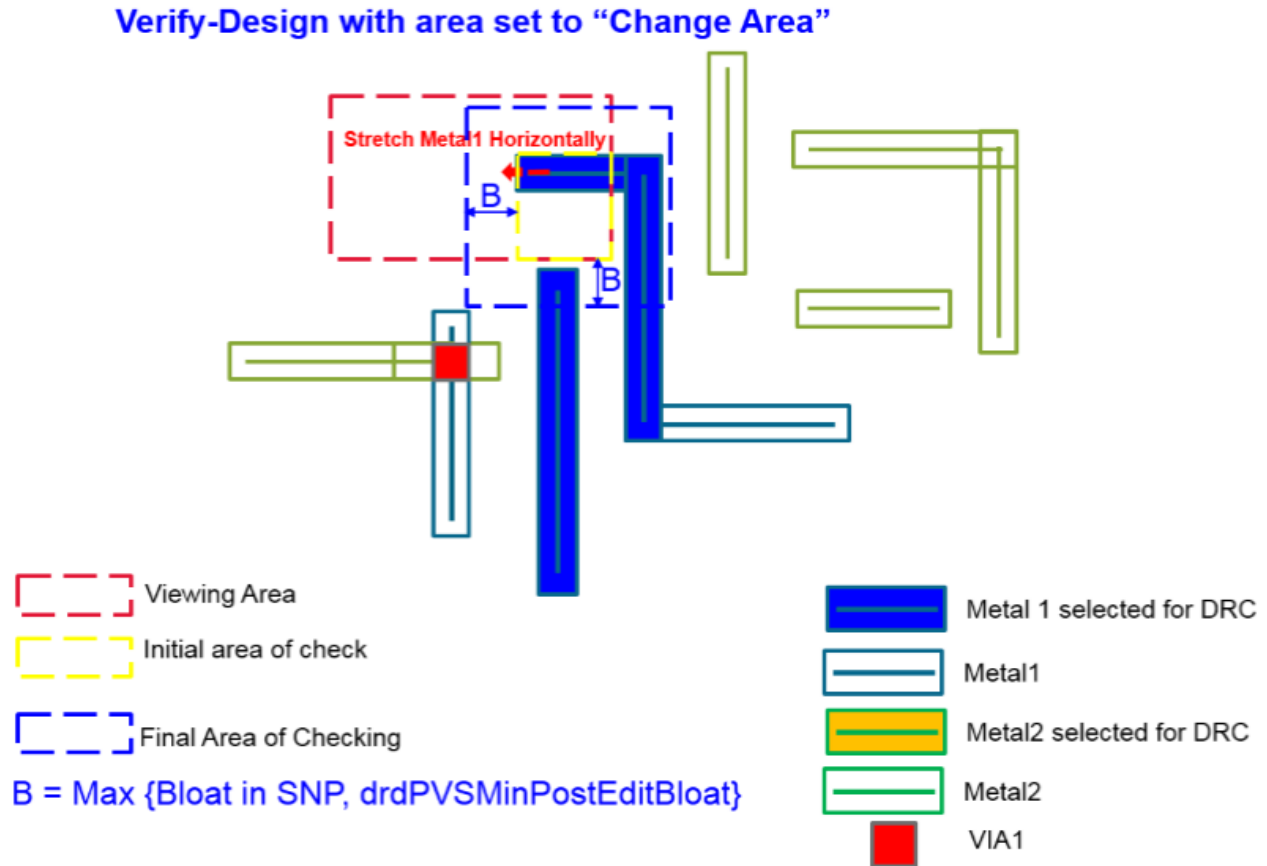


Changed Area

This mode is a relaxed version of Post-Edit mode. It allows you to conduct multiple layout editing and tracks these modification between each DRC. Once you are satisfied with the editing, you can invoke VIPVS 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.

VIPVS 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 VIPVS.

Figure 4-4 Final Area of Checking when Area sets to Change Area



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

Figure 4-5 Always Reset Changed Area setting



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

Figure 4-6 Clear Changed Area from the toolbar



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.

Virtuoso IPVS User Guide

Invoke Design Rule Checking

SignOff Fill: Snapshot

This chapter introduces you to the concept of SignOff fill and how to setup the form to run signoff fill. For the basics of Snapshot creation, see [All about Design Rules: Snapshot](#) on page 59.

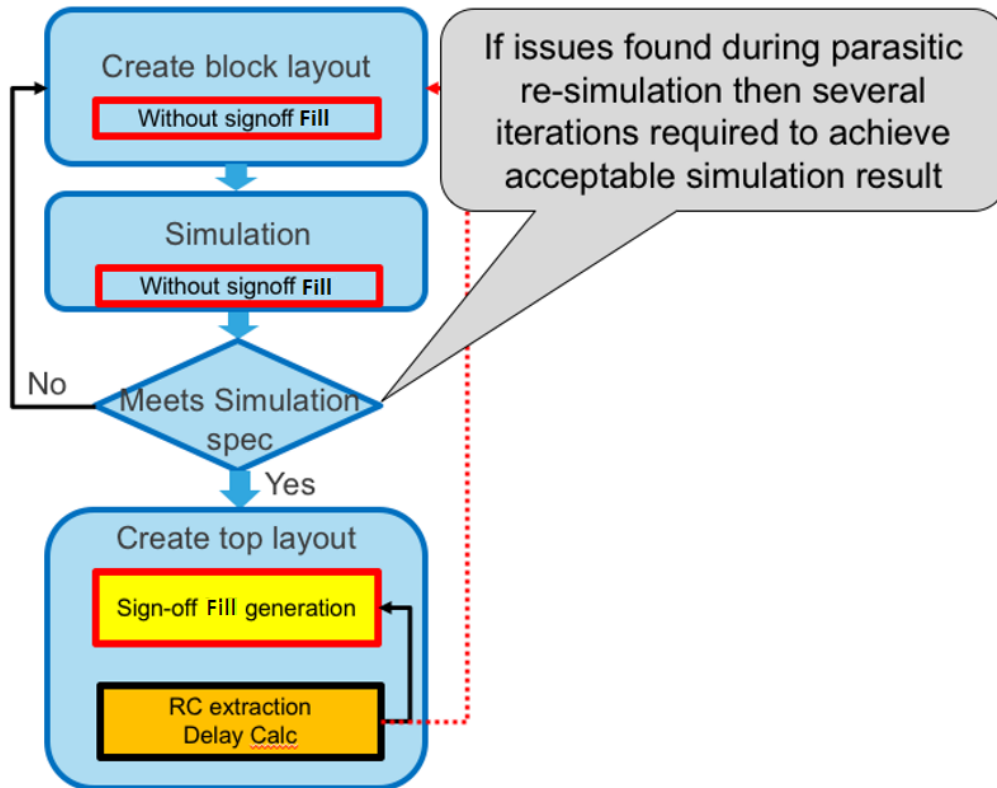
The content of this chapter is organized as follows:

- [Virtuoso IPVS SignOff Fill Overview](#) on page 112
 - [Traditional Fill Flow](#) on page 112
 - [Virtuoso IPVS SignOff Fill Use Model](#) on page 113
 - [Virtuoso IPVS SignOff Fill features](#) on page 113
- [Technology Setup](#) on page 114
- [Virtuoso IPVS SignOff Fill Overview](#) on page 112
- [Creating PVS Fill Snapshot](#) on page 115

Virtuoso IPVS SignOff Fill Overview

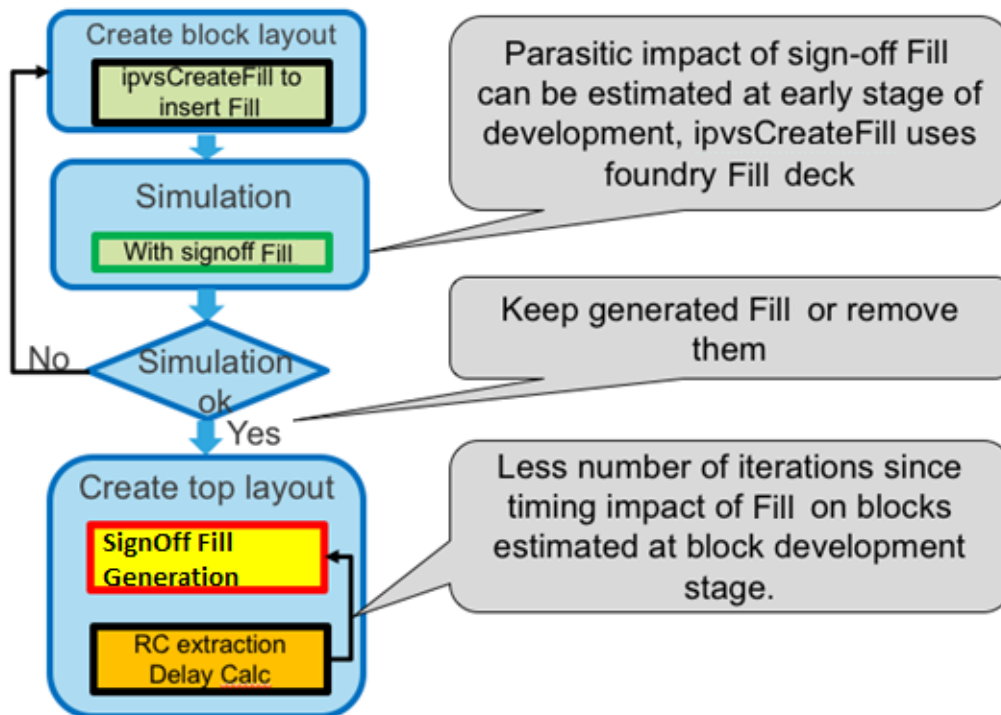
Virtuoso IPVS SignOff Fill allows you to generate dummy fills using foundry fill decks and automatically map generated dummy fills into Virtuoso OA database.

Traditional Fill Flow



Virtuoso IPVS SignOff Fill Use Model

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



Virtuoso IPVS SignOff Fill features

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

Virtuoso IPVS User Guide

SignOff Fill: Snapshot

- Blockages support:
 - Draw blockage on layout using blockage purpose layer and run fill.

Technology Setup

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

The image illustrates the configuration for blockage fills in Virtuoso. It consists of four main components:

- Foundry Rule Deck:** A code block defining a rule named `DFM_RDB` that references a GDS file `fill.gds`. It lists fill layers: `fillm1` (7 5), `fillm2` (9 5), and `fillv1` (8 5). A red box highlights the values 7, 5, 8, and 5.
- Technology layermap file:** A table mapping layer names to purposes and their corresponding layer numbers and data types. A red box highlights the `fill` entries for Metal1, Metal2, and Vial1.
- Layers palette:** A screenshot of the software's layer management interface. It shows a list of layers for Metal1 and Metal2, including `blk`, `bnd`, `drw`, `fil`, `ibl`, `net`, `pin`, and `slot`. The `fill` layer for Metal1 is highlighted in blue.
- Layout View:** A screenshot of a PCB layout showing blue blockage fills. A red arrow points from the `fillm1` entry in the Foundry Rule Deck to the `fill` entry in the Technology layermap file.

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

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

Creating PVS Fill Snapshot

Virtuoso IPVS SignOff Fill feature is snapshot based, same as VIPVS Verify Design. For detailed description of fields in the Virtuoso IPVS Snapshot Creator form, see [All about Design Rules: Snapshot](#) on page 59.

Perform following steps to create PVS fill snapshot:

1. Click *VIPVS Run Options* icon on the *VIPVS* toolbar. The *Virtuoso Run Options* form opens.
2. Select the *SignOff Fill* tab.

The screenshot shows the 'Virtuoso IPVS Run Options' dialog box with the 'SignOff Fill' tab selected. The dialog is divided into several sections:

- Input:** Contains two text fields: 'Rule Map' and 'DFMRDB Layer Map', each with a 'Browse' button to its right.
- Output:** Contains an 'Instance' text field with the value 'pvs_fill'. Below it are two checkboxes: 'Add Snapshot Name to Cell Name' (checked) and 'Custom Fill Library/Cell/View' (unchecked). Under the second checkbox are three text fields: 'Library', 'Cell', and 'View', followed by a 'Select' button.
- Run Options:** Contains a 'Clear Existing Fill' checkbox. Below it is an 'Area' text field with 'Select' and 'Use Function' buttons. Underneath are radio buttons for 'Exclude Cells': 'From Form' (selected) and 'From File'. Below these are 'Library', 'Cell', and 'View' text fields, a '+' button, and a 'Browse' button. A table with columns 'Library', 'Cell', and 'View' is visible below these fields, with a '-' button to its right.
- Layers:** Contains radio buttons for 'All' (selected), 'Active Layer', 'Visible Layers', and 'Custom:'. A text field is located below the 'Custom:' option.
- Snapshots:** Contains a 'Create Snapshot' button and a 'Load current snapshot settings' button.

At the bottom of the dialog are several buttons: 'OK' (highlighted in red), 'Cancel', 'Defaults', 'Apply', 'Save To', 'Load From', and 'Help'.

Virtuoso IPVS User Guide

SignOff Fill: Snapshot

3. Click *Create Snapshot*. The *Virtuoso IPVS Snapshot Creator* form opens.

The screenshot shows the 'Virtuoso IPVS Snapshot Creator' dialog box. The 'Input' section contains a 'Layer Map List' with a text box containing the path 'FTECHs/RAK/TECH/GPDK045/qpdk045_preRel_15DEC11/qpdk045/qpdk045.layermap' and a 'Remove' button. Below this is an 'Object Name Table' with an empty text box and a button. A checked 'Enable Coloring' checkbox is also present. The 'Output' section includes a 'Snapshot Name' text box, a 'Snapshot Comment' text box, a 'Snapshot Path' dropdown menu showing '/home/sarita/RAK/cadence/pvs/sarita/presets/vipvs/snapshot', a 'Snapshot Type' dropdown menu set to 'DRC', and a 'Set Custom Bbat Value' checkbox with a value of '0.0'. At the bottom right are 'OK', 'Apply', and 'Cancel' buttons.

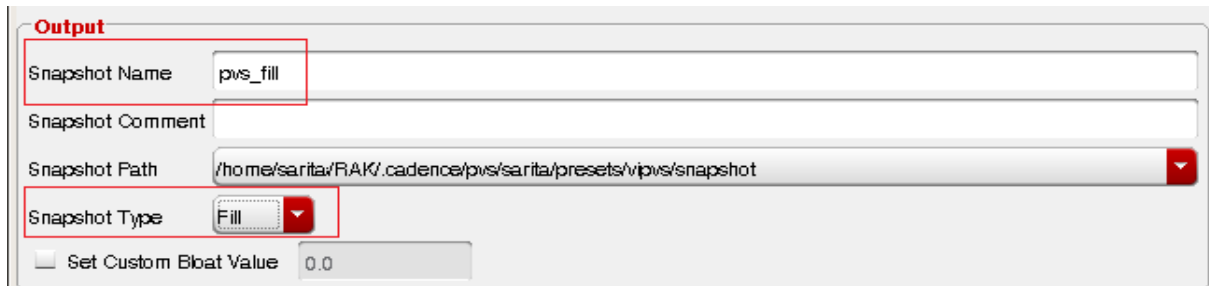
For detailed description of *Virtuoso IPVS Snapshot Creator* form fields, see [Virtuoso IPVS Snapshot Creator Form](#) on page 69.

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

Virtuoso IPVS User Guide

SignOff Fill: Snapshot

5. In the *Output* section, enter the snapshot name and set *Snapshot Type* as *Fill*.



Output

Snapshot Name: pvs_fill

Snapshot Comment:

Snapshot Path: /home/sarita/RAK/cadence/pvs/sarita/presets/vipvs/snapshot

Snapshot Type: Fill

Set Custom Bloat Value: 0.0

6. Click *OK* to generate snapshot and close the form. Then close *Virtuoso IPVS Run Options* form. *VIPVS* toolbar shows pvs_fill snapshot.



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

Virtuoso IPVS User Guide
SignOff Fill: Snapshot

Running SignOff Fill

This chapter includes the process of running Virtuoso IPVS SignOff Fill from Virtuoso and generating fill in OA database.

The content of this chapter is organized as follows:

- [Running VIPVS SignOff Fill](#) on page 120
- [Running VIPVS SignOff Fill with Blockages](#) on page 122
- [Running VIPVS SignOff Fill in Area](#) on page 124

Running VIPVS SignOff Fill

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

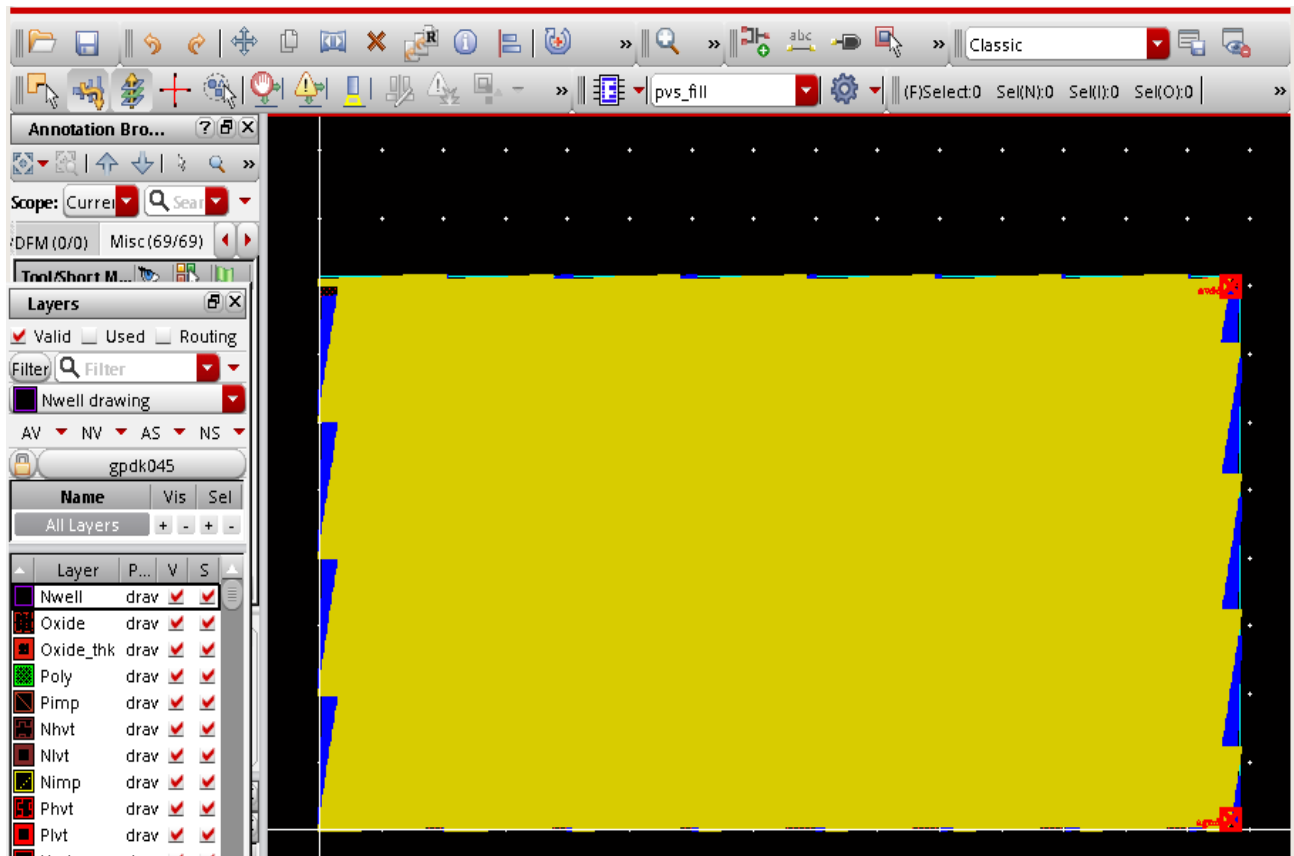
Figure 6-1 VIPVS signoff fill icon



Perform following steps to run VIPVS signoff fill:

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

Figure 6-2 Run VIPVS SignOff Fill



Virtuoso IPVS User Guide

Running SignOff Fill

The generated fills are inserted as instance.

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

The screenshot shows the 'Virtuoso IPVS Run Options' dialog box with the 'SignOff Fill' tab selected. The 'Input' section contains 'Rule Map' and 'DFMRDB Layer Map' fields with 'Browse' buttons. The 'Output' section contains an 'Instance' field with 'pvs_fill', a checked checkbox for 'Add Snapshot Name to Cell Name', an unchecked checkbox for 'Custom Fill Library/Cell/View', and three empty text boxes for 'Library', 'Cell', and 'View' with a 'Select' button.

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 specified area by making appropriate selections in the *Virtuoso IPVS Run Options* form. For more details see, [Run Settings for SignOff Fill](#) on page 49.

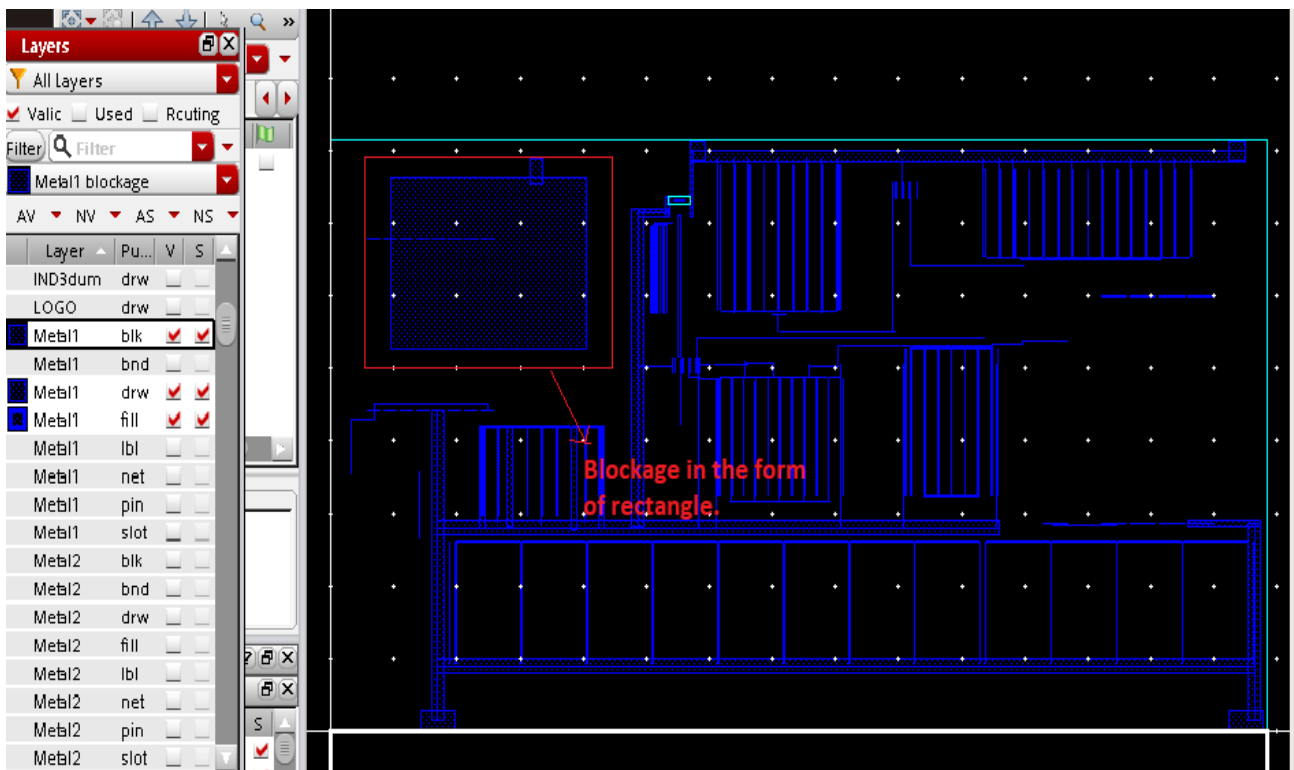
Running VIPVS SignOff Fill with Blockages

In Virtuoso IPVS 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 rule deck layer numbers and blockage layer/purpose in Virtuoso.

Perform following steps to create dummy fills under blockages:

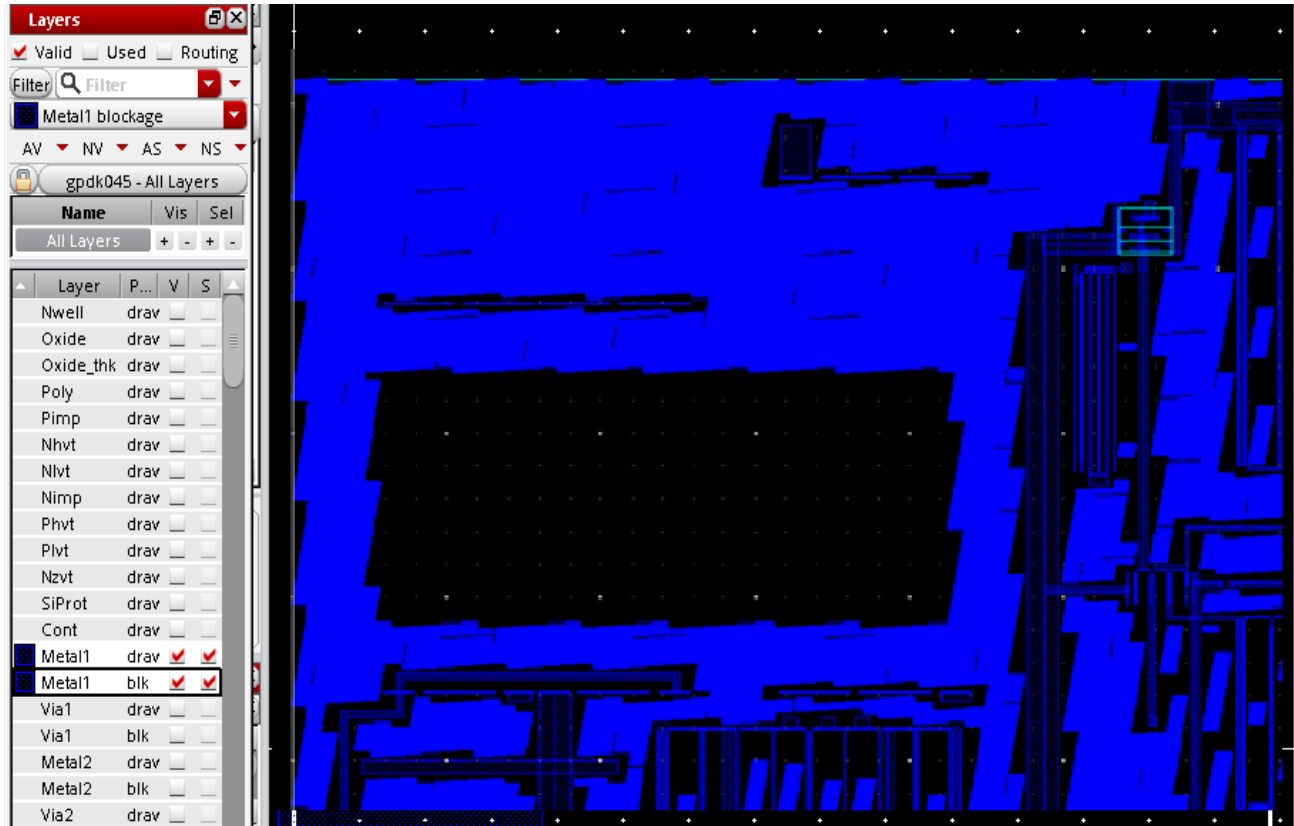
1. Open *Virtuoso IPVS Run Options* form and go to the *SignOff Fill* tab.
2. Select appropriate run options and click *OK*.
3. Select blockage layer on Virtuoso Layer Palette and create blockage by drawing a rectangle on the layer that is part of your SignOff Fill run.



Virtuoso IPVS User Guide

Running SignOff Fill

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

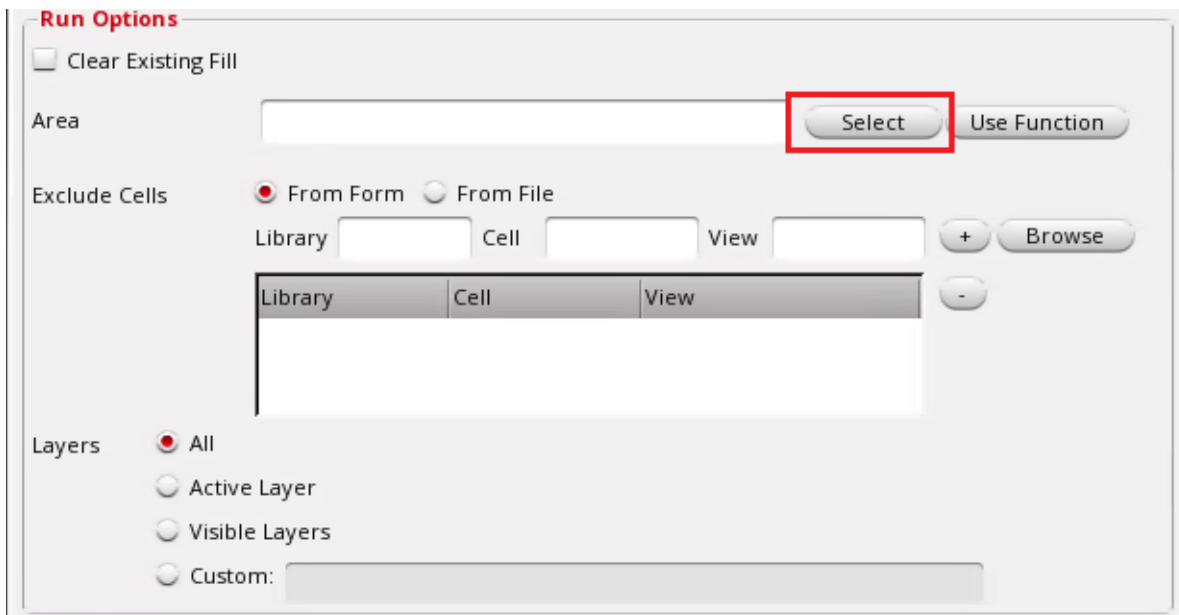


Running VIPVS SignOff Fill in Area

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

Perform following steps to generate fills incrementally:

1. Open the *Virtuoso IPVS Run Options* form and go to the *SignOff Fill* tab.
2. Make appropriate selection on layer visibility and in the *Run Options* block, click the *Select* button of the *Area* field.



Run Options

Clear Existing Fill

Area **Select** Use Function

Exclude Cells From Form From File

Library Cell View + Browse

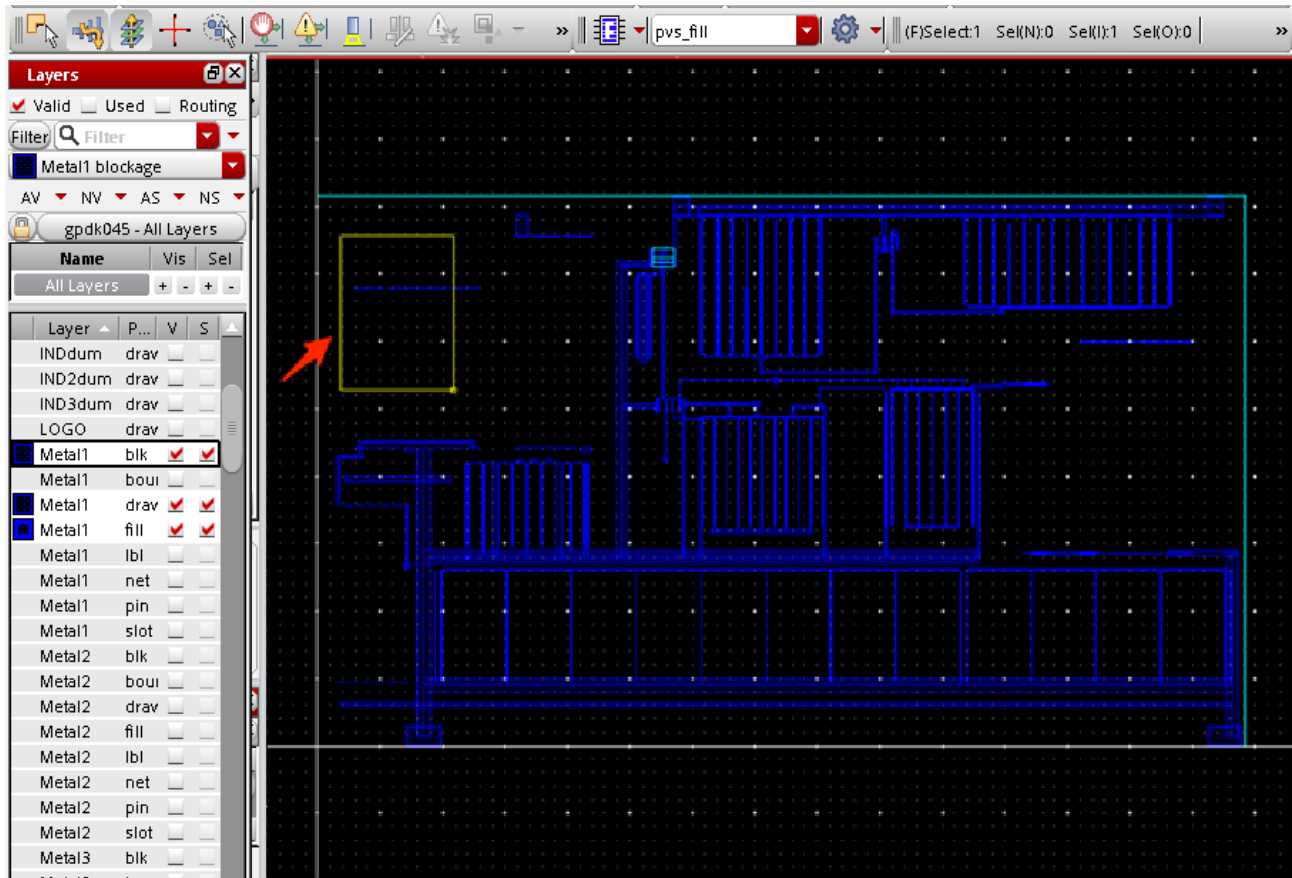
Library	Cell	View

Layers All Active Layer Visible Layers Custom:

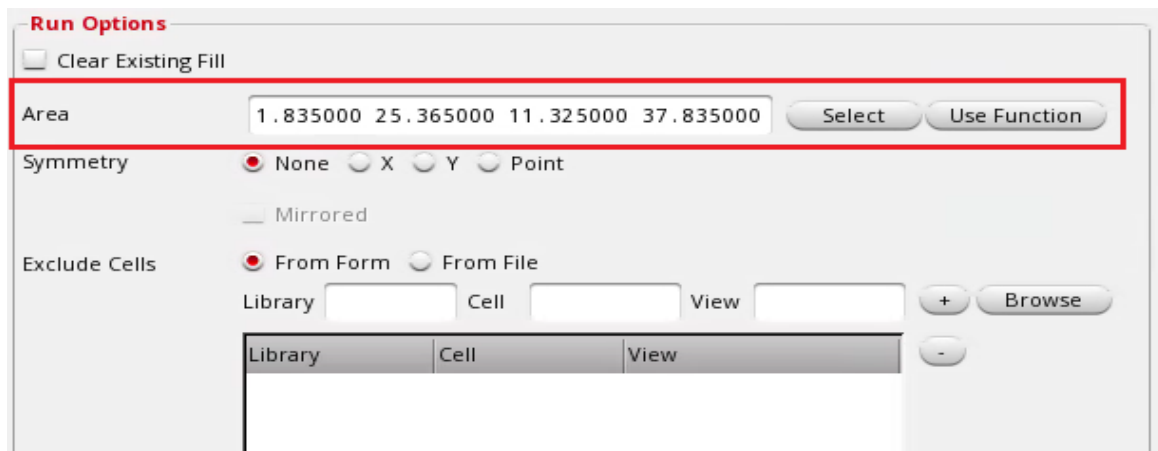
Virtuoso IPVS User Guide

Running SignOff Fill

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



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

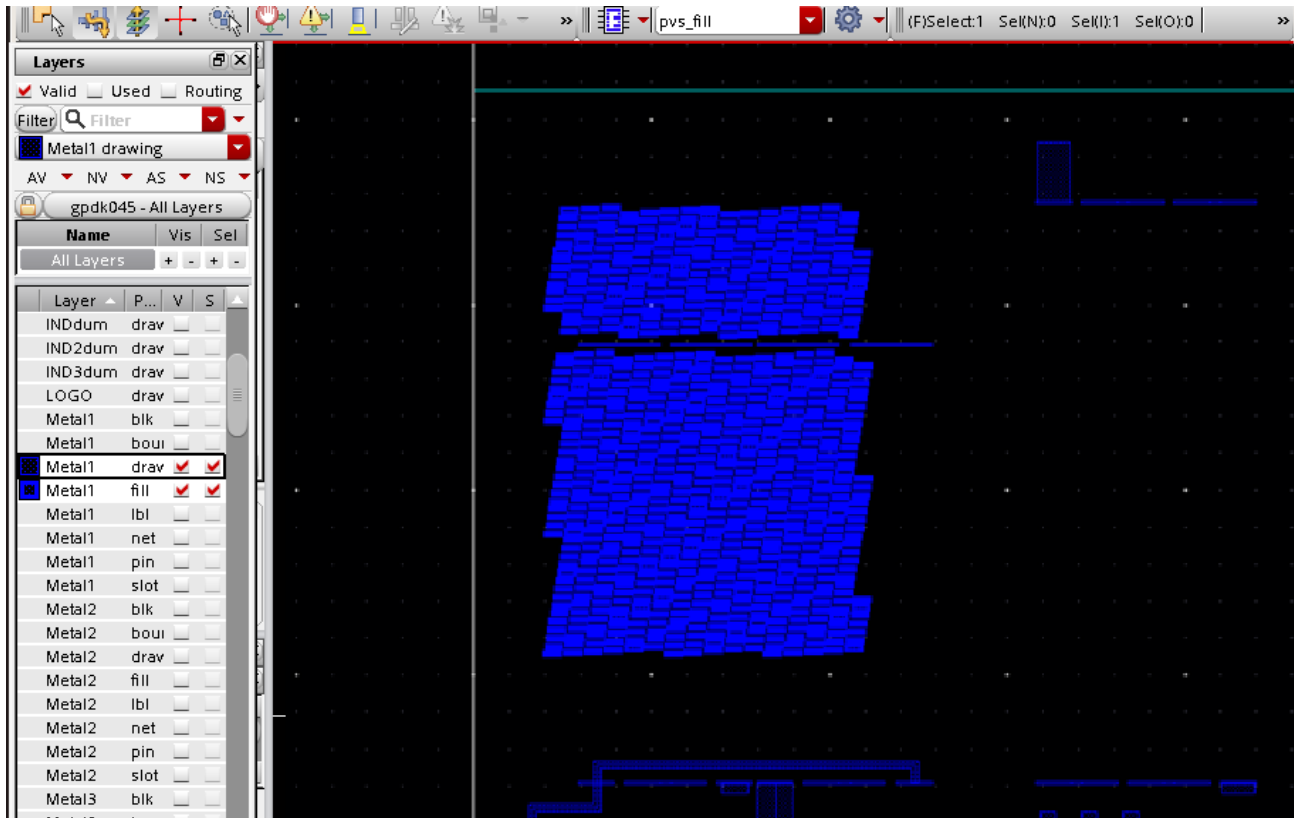


4. Click *OK* to close the form.

Virtuoso IPVS User Guide

Running SignOff Fill

5. Run VIPVS SignOff Fill.



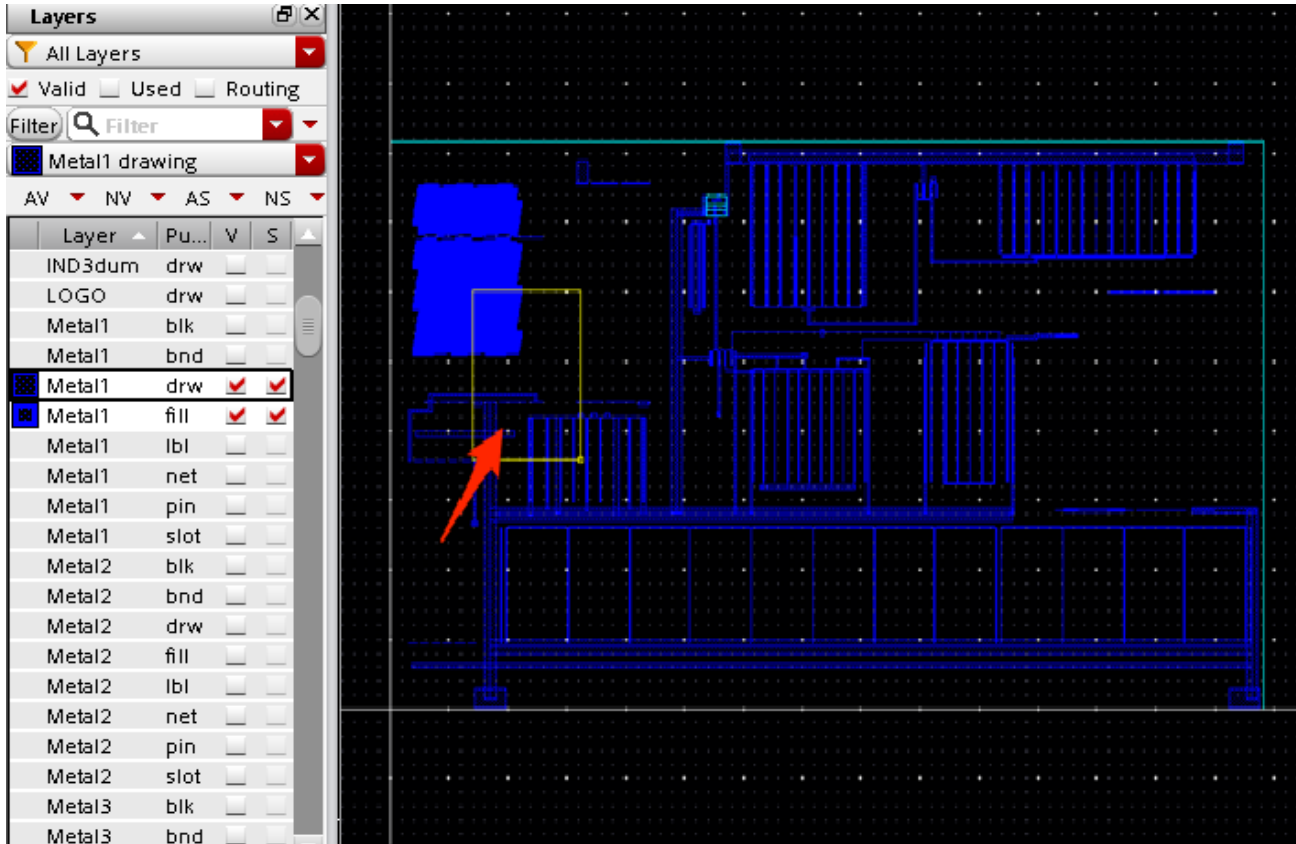
Dummy fills are generated for specified area. Dummy fills outside specified area are not generated.

6. Open the *Virtuoso IPVS Run Options* form and go to the *SignOff Fill* tab.

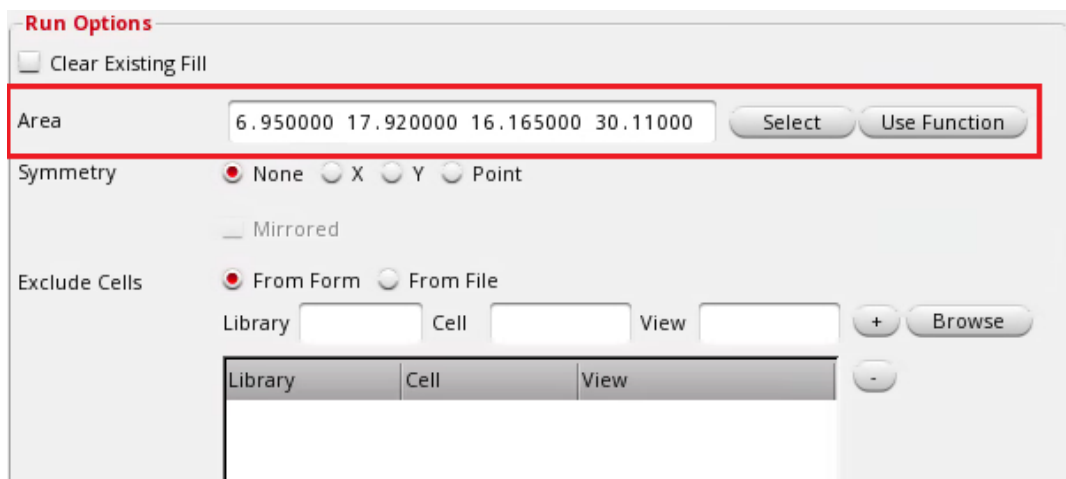
Virtuoso IPVS User Guide

Running SignOff Fill

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



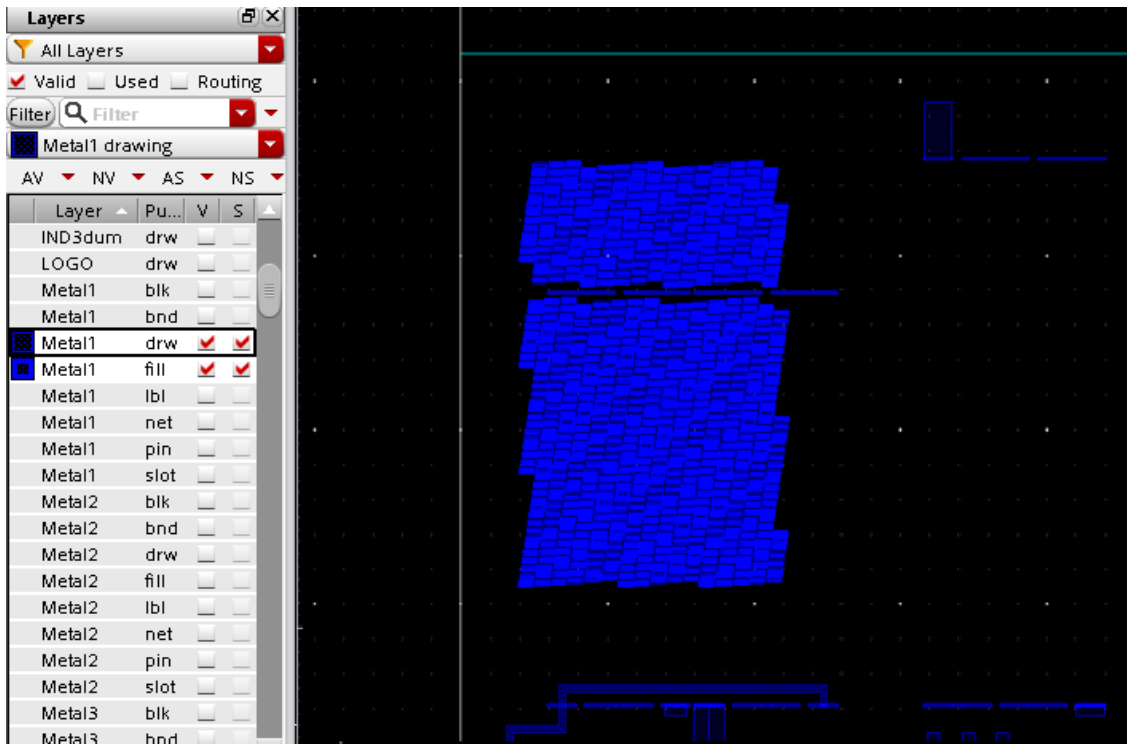
8. Once you do second click to define second point of rectangle, the *Area* field will be automatically populated with coordinates of the boundary box:



Virtuoso IPVS User Guide

Running SignOff Fill

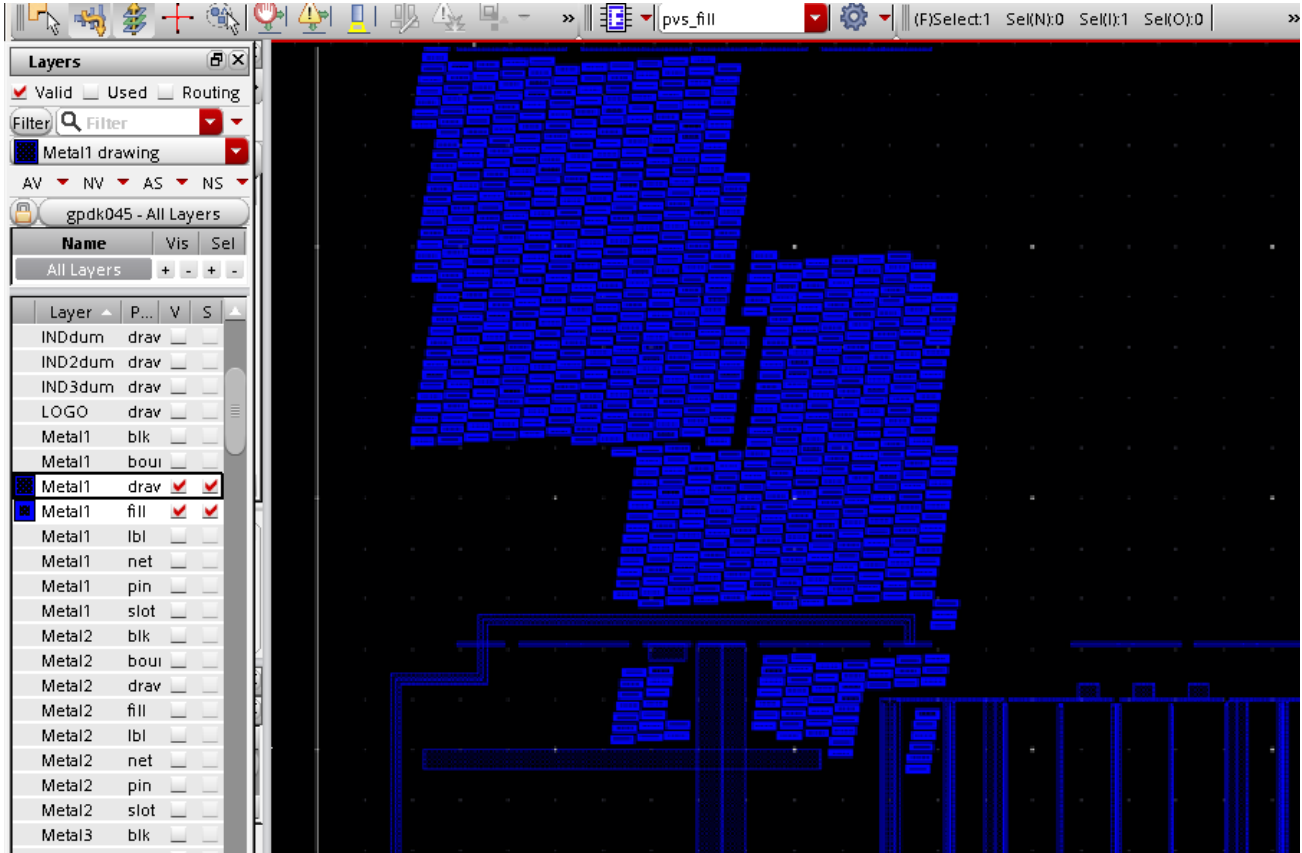
9. Click *OK* to close the form and run VIPVS signoff fill.



Virtuoso IPVS User Guide

Running SignOff Fill

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



Virtuoso IPVS User Guide

Running SignOff Fill

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 VIPVS Density Analysis](#) on page 132
- [Running Density Analysis](#) on page 133
- [Running Density Analysis on Visible Layers](#) on page 136
- [Merging Density Analysis Results](#) on page 140

Introduction to VIPVS Density Analysis

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

You can customize the results per your requirements using the options available in the Density analysis tab of the VIPVS Options form:

Figure 7-1 Density Analysis Tab

The screenshot shows the 'Virtuoso IPVS Run Options' dialog box with the 'Density Analysis' tab selected. The dialog has three tabs: 'Verify Design', 'SignOffFill', and 'Density Analysis'. The 'Density Analysis' tab contains several sections:

- Layer Visibility:** Includes radio buttons for 'Active Layer', 'Visible Layers', and 'Custom'. There are also 'Merge' checkboxes and 'Purpose' dropdown menus. The 'Active Layer' purpose is set to 'Selected'.
- Density:** Includes input fields for 'Size X [Y]' (10) and 'Step X [Y]' (10). A 'Scale [%]' field is set from 0 to 100. A 'Gradient' checkbox is checked, with a dropdown set to '1e' and a field set to '0'.
- Viewer:** Includes input fields for 'Histogram Bins' (10), 'Marker Limit (Total)' (1000000), and 'Marker Limit (Per Rule)' (1000000).
- Coloring:** Includes a checkbox for 'Enable Coloring' which is currently unchecked.

For detailed description of the fields of *Density Analysis* tab, see [Run Settings for Density Analysis](#) on page 53.

Running Density Analysis

Perform following steps to do density analysis:

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

The screenshot shows the 'Virtuoso IPVS Run Options' dialog box with the 'Density Analysis' tab selected. The dialog has three tabs: 'Verify Design', 'SignOff Fill', and 'Density Analysis'. The 'Density Analysis' tab is active and contains the following sections:

- Layer Visibility:** Includes radio buttons for 'Active Layer' (selected), 'Visible Layers', and 'Custom:'. There are 'Merge' checkboxes and 'Purpose' dropdown menus for each. The 'Active Layer' purpose is set to 'Selected' and 'Visible Layers' is set to 'Visible'.
- Density:** Includes input fields for 'Size X [Y]' (10) and 'Step X [Y]' (10). A 'Scale [%]' field is set to '0 to 100'. A 'Gradient' checkbox is checked with a value of '1e' and '0'.
- Viewer:** Includes input fields for 'Histogram Bins' (10), 'Marker Limit (Total)' (1000000), and 'Marker Limit (Per Rule)' (1000000).
- Coloring:** Includes a checkbox for 'Enable Coloring' which is unchecked.

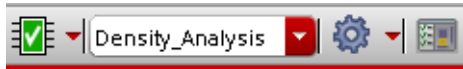
At the bottom of the dialog are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', 'Save To', 'Load From', and 'Help'.

4. Click *OK* to close the *Virtuoso IPVS Run Options* form.

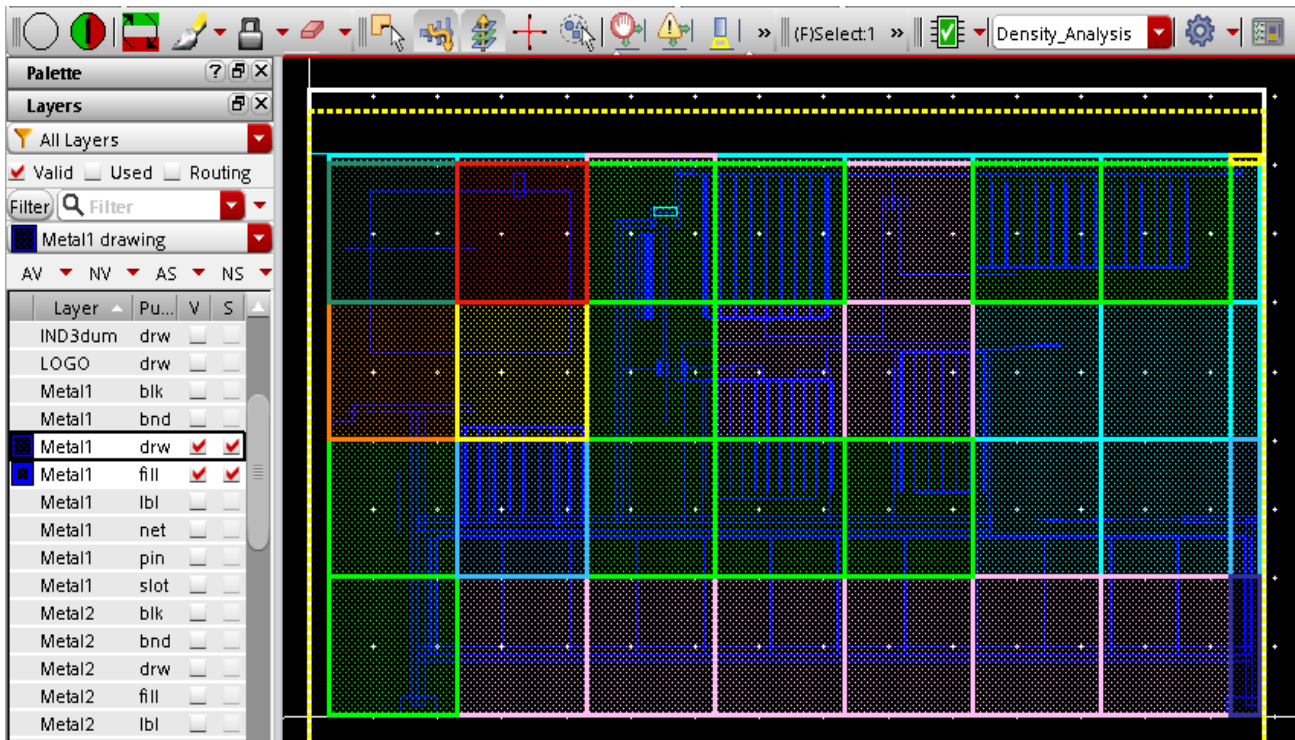
Virtuoso IPVS User Guide

Running Density Analysis

5. Select *Density Analysis* from the *VIPVS* toolbar.



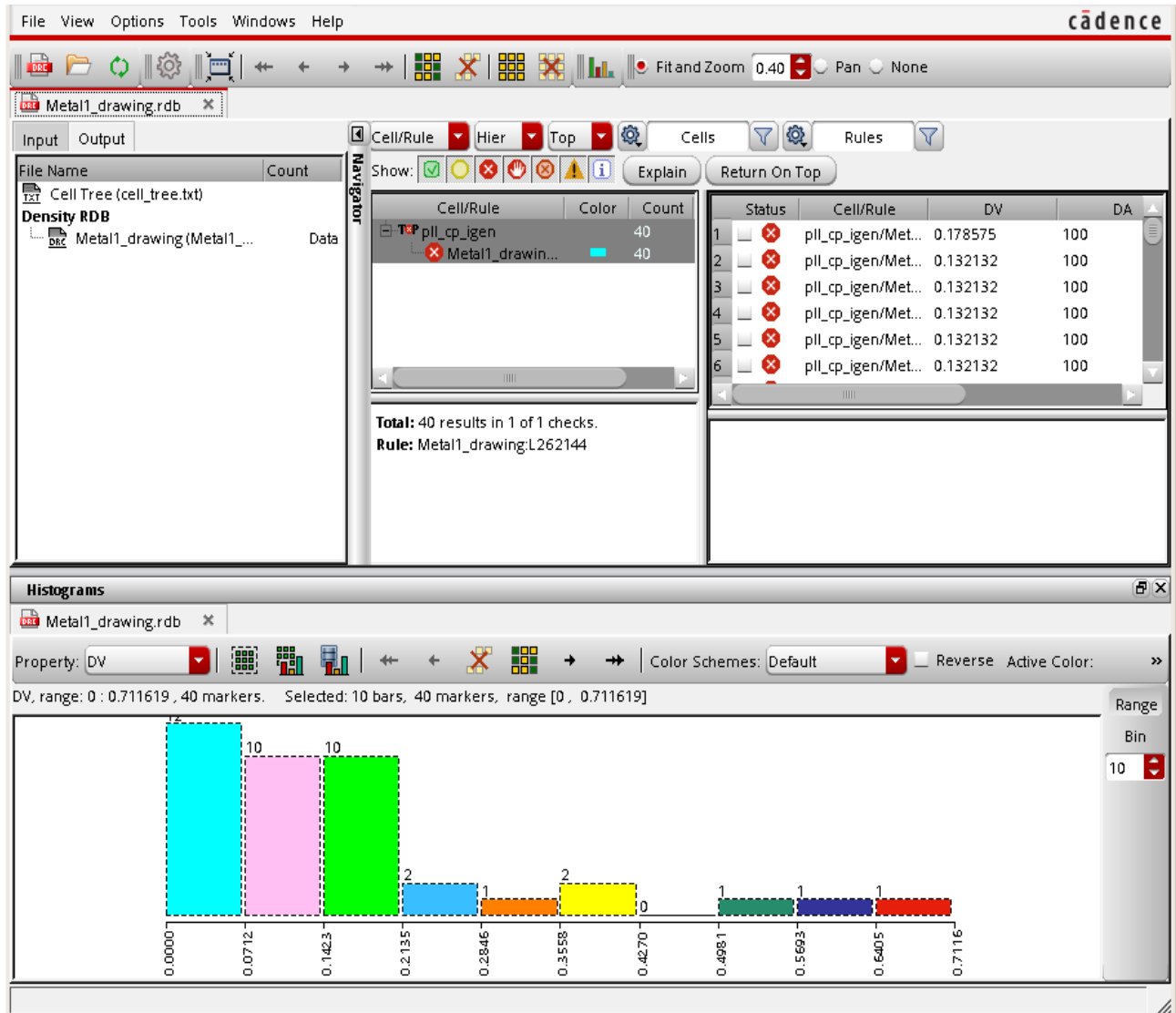
6. Click the *Run VIPVS Verify Design* icon. On completion of the run, the density heatmap is shown on the layout canvas:



Virtuoso IPVS User Guide

Running Density Analysis

The PVS Results Viewer also opens showing the generated histogram:

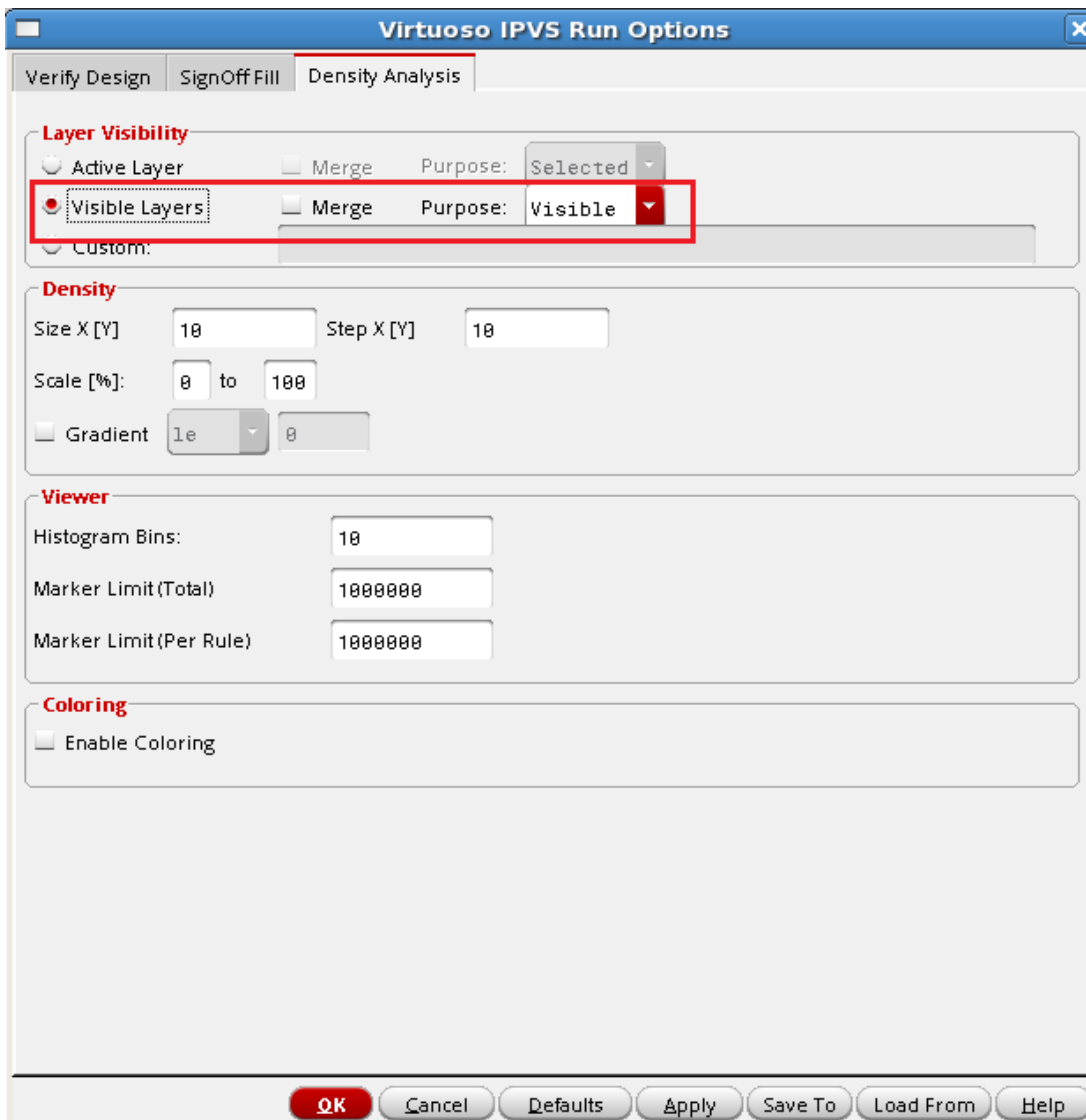


In the steps above the density analysis was run on the active layer. Similarly you can run density analysis on the Visible and Custom Defined layers.

Running Density Analysis on Visible Layers

Perform following steps to run density analysis for visible layers:

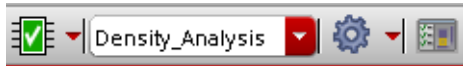
1. Set the visibility of the layers to on for which you want to run the density analysis. In this example, two layers `metal1_drawing` and `metal1_fill` are set to visible.
2. Click *VIPVS Run Options* icon on the *VIPVS* toolbar. The *Virtuoso Run Options* form opens.
3. Select the *Density Analysis*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



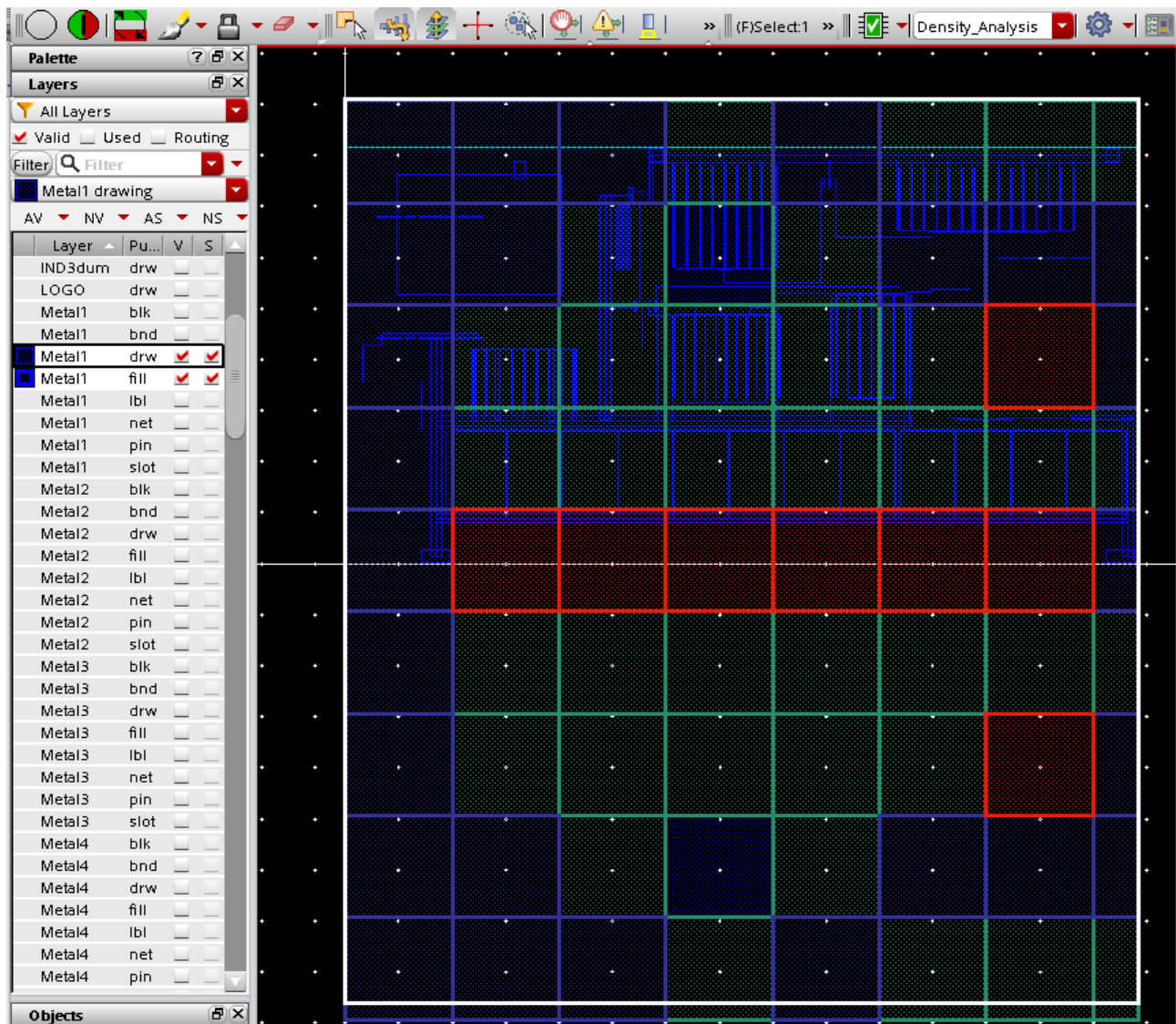
Virtuoso IPVS User Guide

Running Density Analysis

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



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

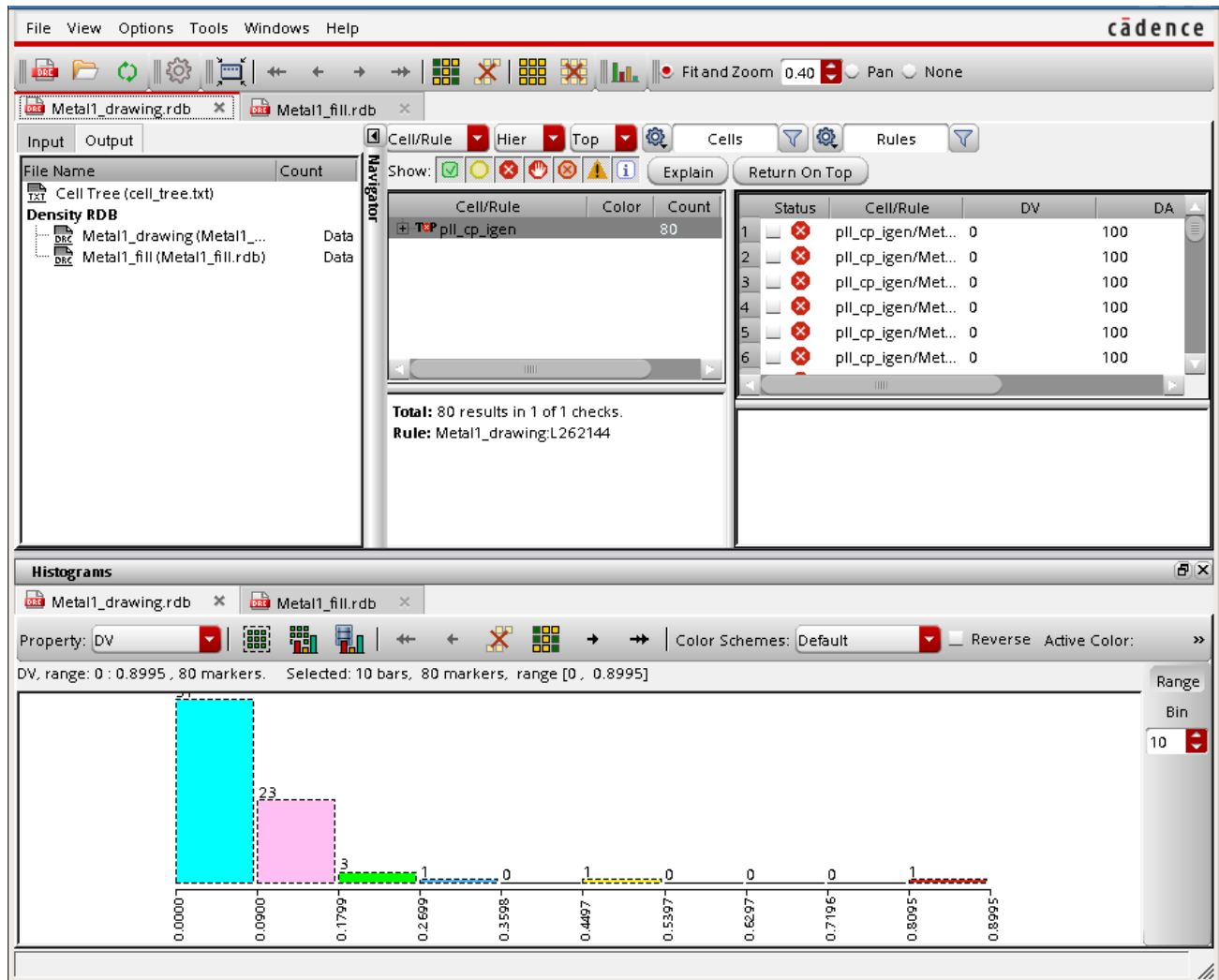


The PVS Results Viewer also opens showing the generated histograms. The PVS Results Viewer shows two tabs for each layer showing two histograms:

Virtuoso IPVS User Guide

Running Density Analysis

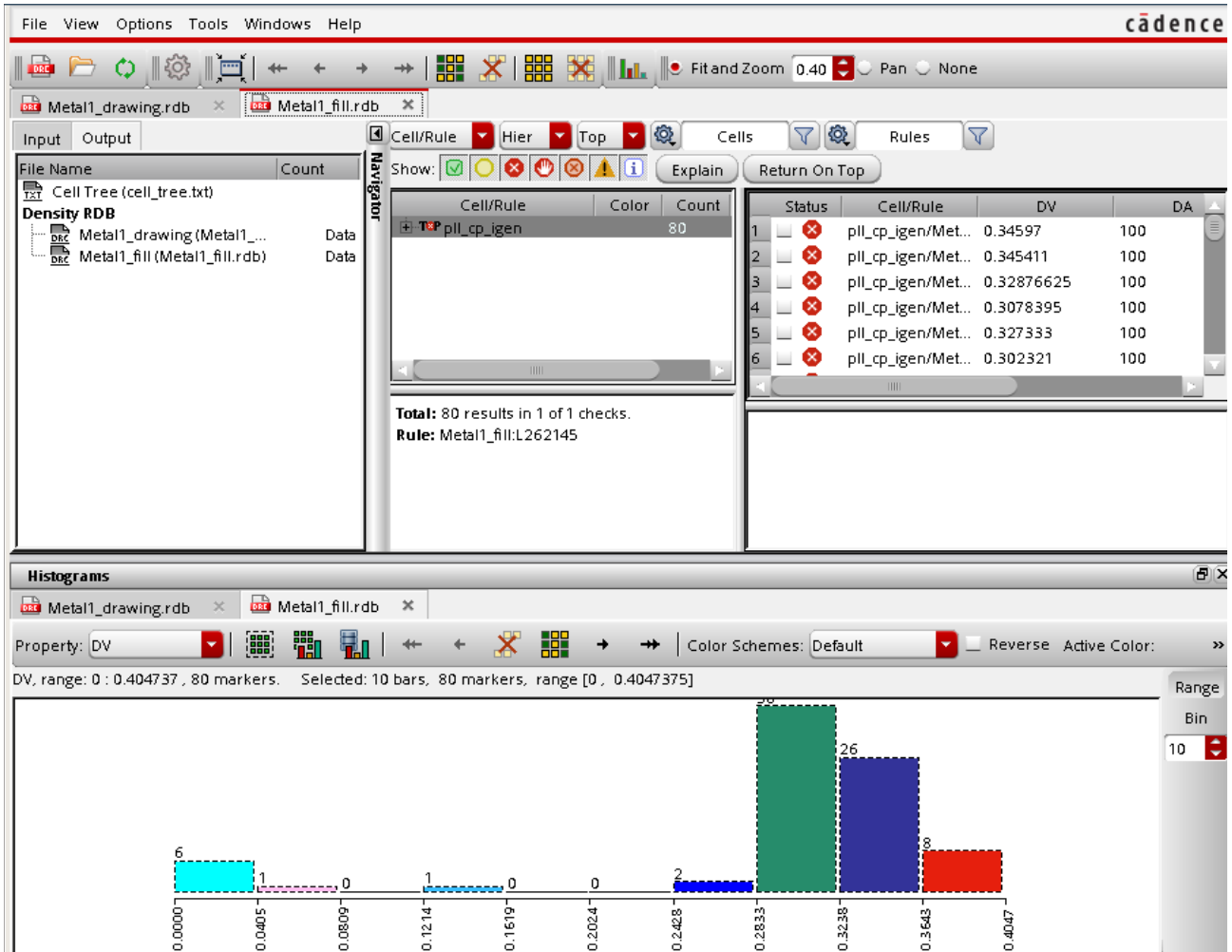
Figure 7-2 Histogram 1 for Metal1_drawing layer



Virtuoso IPVS User Guide

Running Density Analysis

Figure 7-3 Histogram 2 Metal1_fill layer



Merging Density Analysis Results

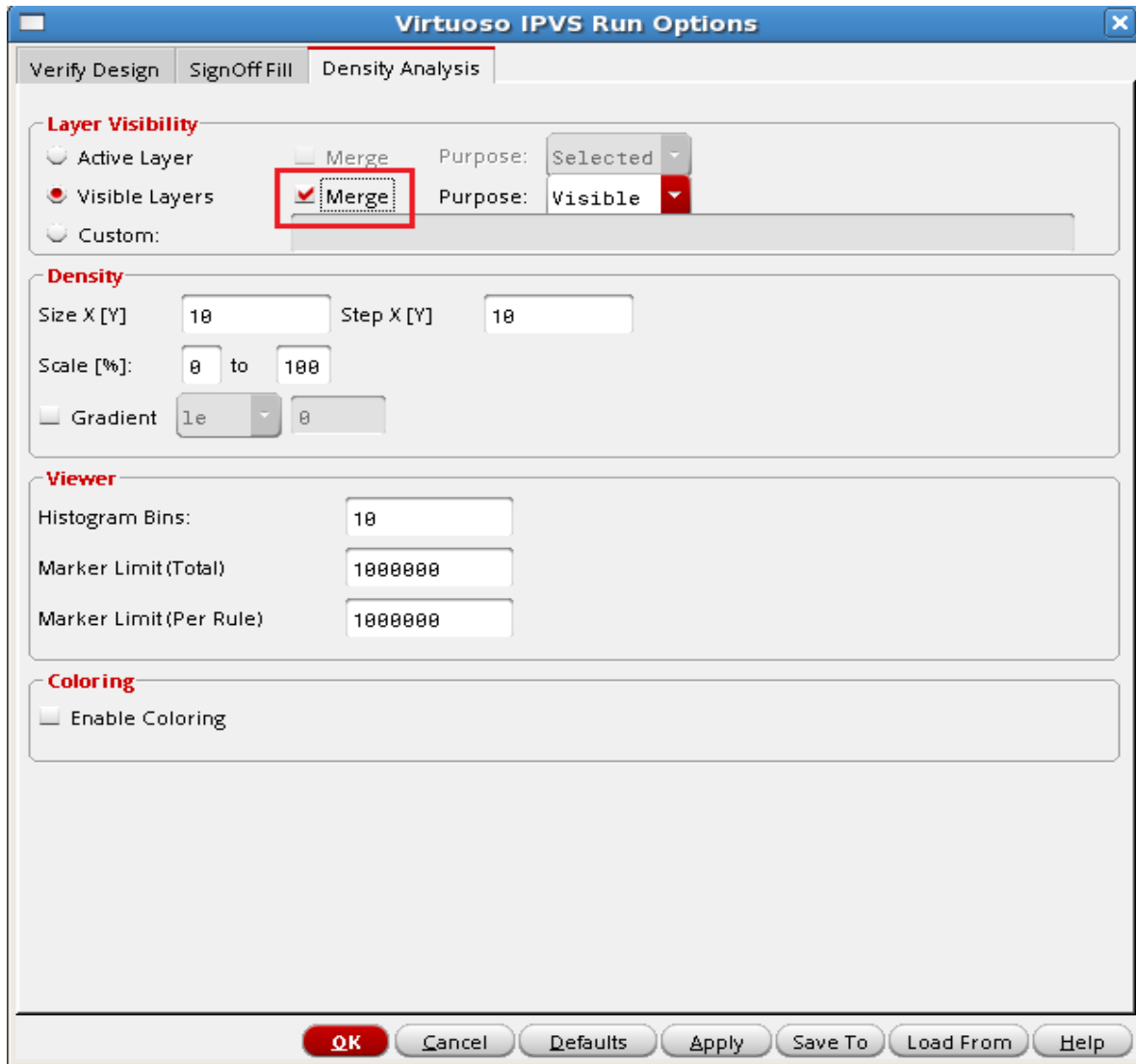
You can merge the run results of the multiple layers using the *Merge* option in the *Density Analysis* tab. In this case, the tabs are merged into one and PVS Results Viewer and histogram show result under one tab only.

1. Set the visibility of the layers to on for which you want to run the density analysis. In this example, two layers `metal1_drawing` and `metal1_fill` are set to visible.
2. Click *VIPVS Run Options* icon on the *VIPVS* toolbar. The *Virtuoso Run Options* form opens.
3. Select the *Density Analysis* tab.

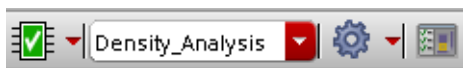
Virtuoso IPVS User Guide

Running Density Analysis

4. Select *Visible Layers* radio button in the *Layer Visibility* section and also select the *Merge* check box.



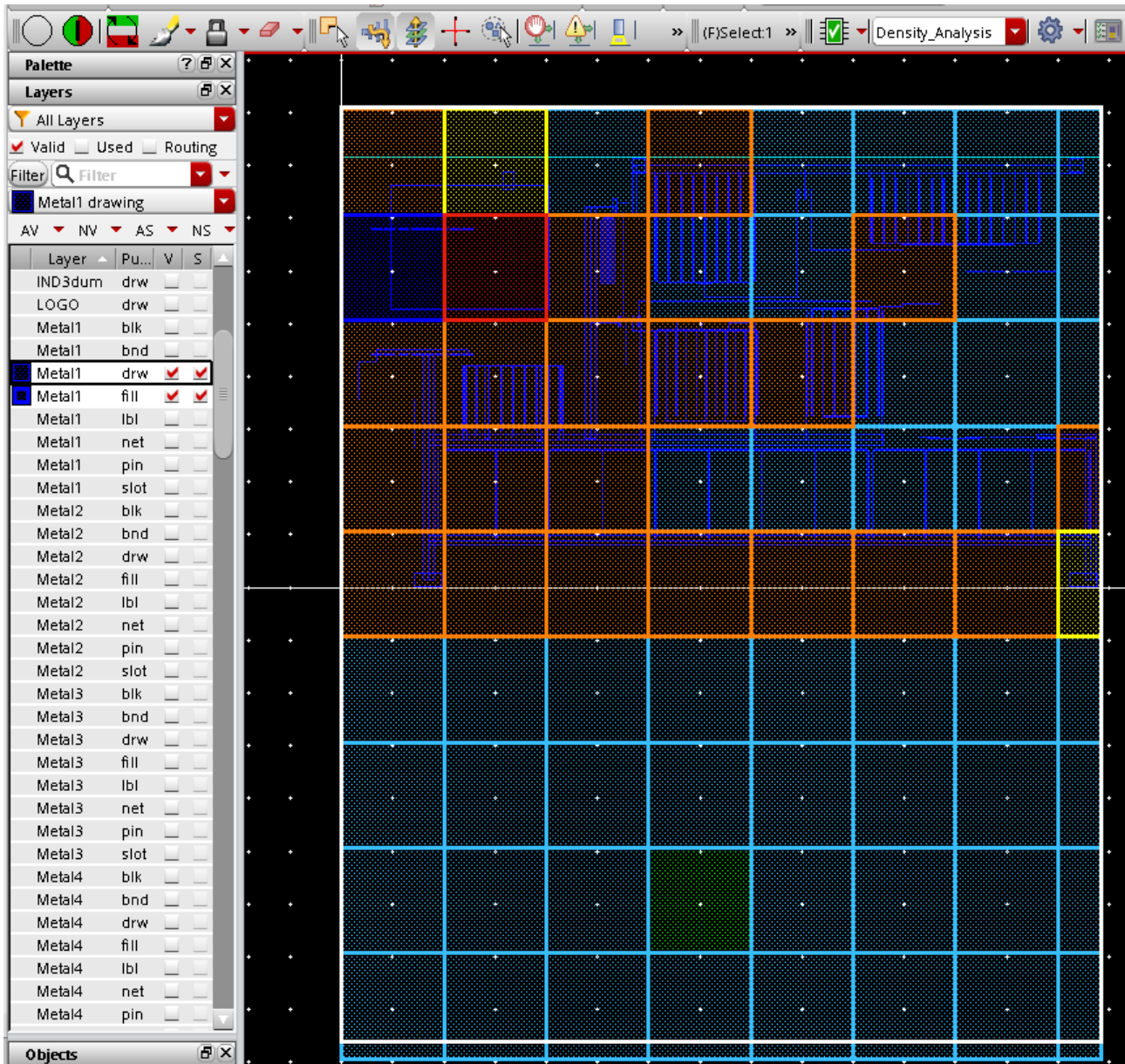
5. Click OK to the *Virtuoso Run Options* form.
6. Select *Density Analysis* from the *VIPVS* toolbar.



Virtuoso IPVS User Guide

Running Density Analysis

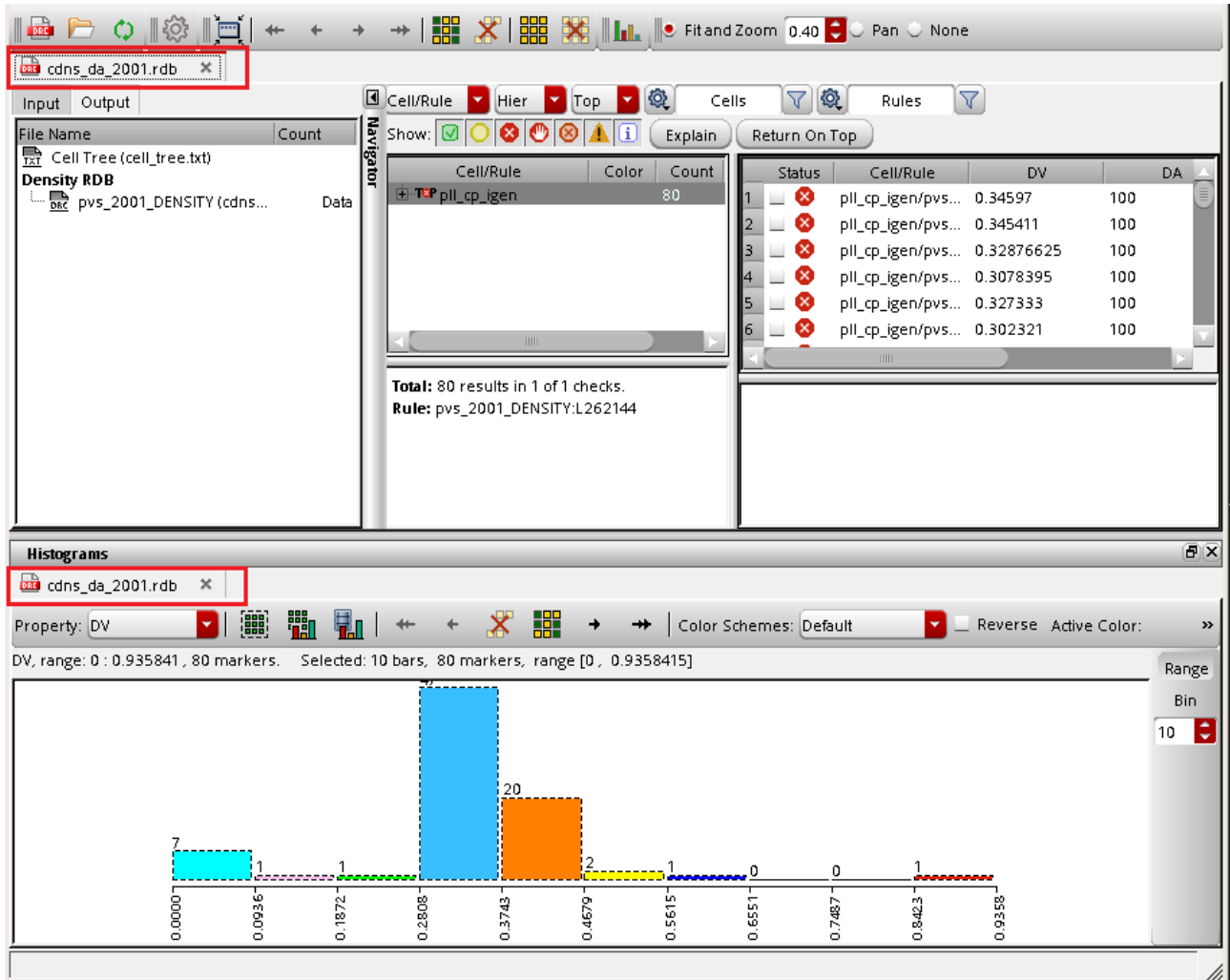
7. Click the *Run VIPVS Verify Design* icon. On completion of the run, the density heatmap is shown on the layout canvas:



Virtuoso IPVS User Guide

Running Density Analysis

The PVS Results Viewer opens showing the generated merged histogram:



Virtuoso IPVS User Guide

Running Density Analysis

Error Viewing

VIPVS supports three type of browsers: one is the Virtuoso native supported Annotation Browser, second is PVS sign-off DRC Debug Environment (DRCDE) and third is PVS Results Viewer. DRCDE is the default browser for VIPVS. This chapter describes the basic setup and customization of the browsers. For details of each browser usage, refer to:

- Annotation Browser: Virtuoso Layout Suite XL User Guide
- DRC Debug Environment: Cadence Physical Verification System User Guide
- PVS Results Viewer: Cadence Physical Verification System User Guide

The content of this chapter is organized as follows:

- DRC Design Environment on page 146
 - Quick Start: PV Color Scheme Setup in Virtuoso on page 146
 - Customization on page 146
- Annotation Browser on page 147
- PVS Results Browser on page 148

DRC Design Environment

Quick Start: PV Color Scheme Setup in Virtuoso

Since PVS sign-off DRC Debug Environment's default color display package is packaged within PVS installation, its color scheme setup is mandatory for DRCDE to operate within Virtuoso correctly.

For details of DRCDE color scheme setup, refer to *Setting Colors for DRC Debug Environment* in *Cadence Physical Verification System User Guide*.

Customization

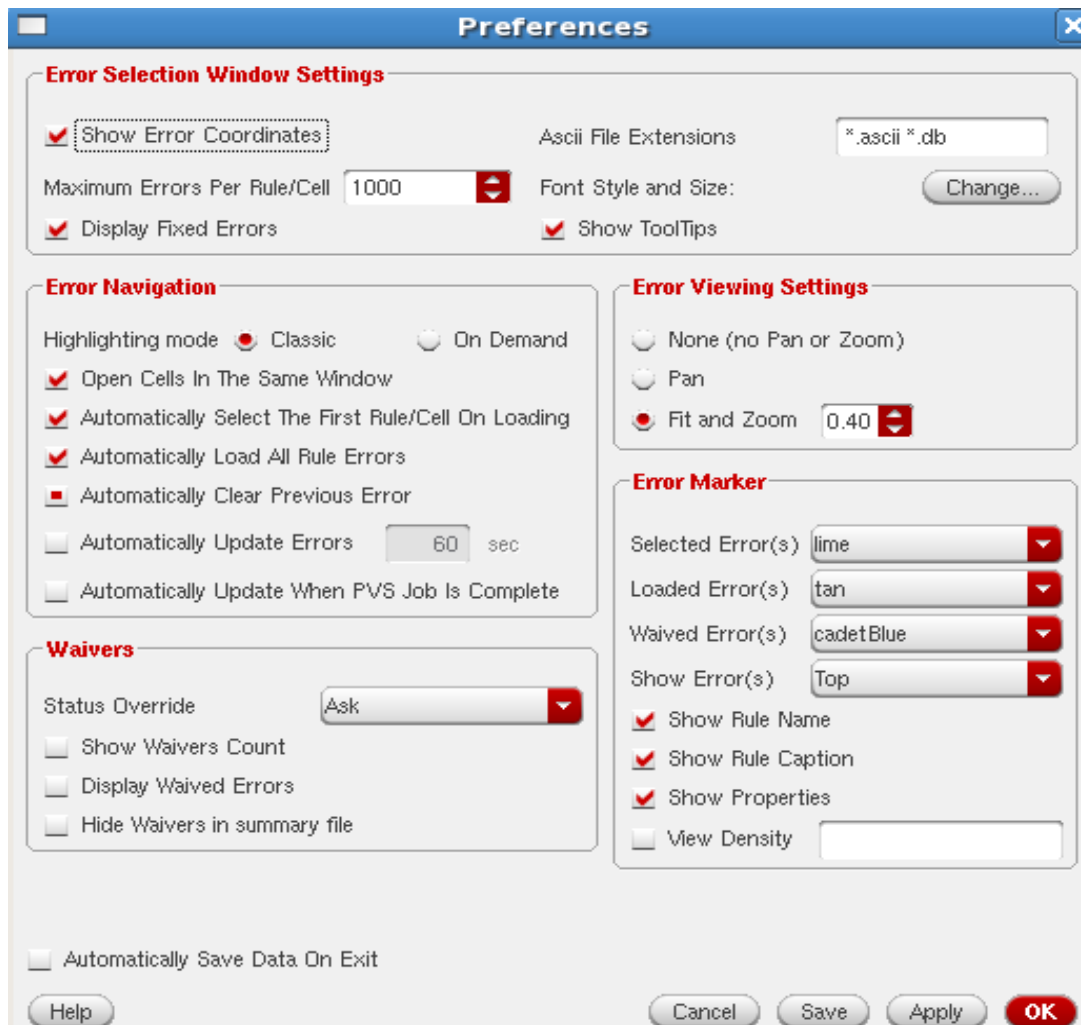
You can customize DRCDE setting through the *Preferences* form. You can save this setting by clicking *Save* in the *Preferences* form. By default, it is saved under `~/ .cds_pvsui` if `PVSUI_SETTING` is not set.

For more details on DRCDE customization, refer to the *Debugging DRC* chapter in *Cadence Physical Verification System User Guide*.

Virtuoso IPVS User Guide

Error Viewing

Figure 8-1 Customize DRCDE



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 cellview. However, browser will not be invoked like DRCDE after each run. You have to invoke it through *Window > Assistants > Annotation Browser* in the layout. Once it is invoked, VIPVS violations are populated under the *DRC/DFM* tab.

Similar to DRCDE, 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*.

PVS Results Browser

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

Batch Commands

This chapter provides the syntax, descriptions, and examples for the batch utility that can be used in conjunction with the Virtuoso® Integrated Physical Verification System (VIPVS) editing feature.

ipvsGenSnapshot

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 PVS and can be used directly for PVS sign-off technology enablement. By default, this utility generates a preset file that can be loaded into *Virtuoso IPVS Snapshot Creator*.

Usage:

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

To generate a snapshot via the command line:

```
ipvsGenSnapshot -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:

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

Virtuoso IPVS User Guide

Batch Commands

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 VIPVS is referenced to. If `Virtuoso_IPVS_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 VIPVS is referenced to. Ignored if `-preset` argument is specified.
- `-control FilePath` Optional keyword to specify the control file containing PVS 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 `Virtuoso_IPVS_defaultSNPRuleDeck` is set, this option is not required. However, explicit declaration has priority over this shell variable. Ignored if `-preset` argument is specified.

Important

VIPVS 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      ""
```

Virtuoso IPVS User Guide

Batch Commands

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

```
[ERROR] PVS 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 *Virtuoso IPVS 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 this message and exits.
- `-v, -V, -version, --version` Prints PVS version and exits.

```
setenv PVS_SNAP_SHOTS_DIRS $PWD/CAD_SNP_BatchDir
setenv Virtuoso_IPVS_DefaultSNPRuleDeck $RULEFILE
setenv Virtuoso_IPVS_DefaultSNPLayerMap $LAYERMAP

#ipvs -- Basic without control file
ipvsGenSnapshot -name CDN20_Default -dir $PVS_SNAP_SHOTS_DIRS -log ./cdn20Default.snpGen.log

#ipvs -- switch on and off via control file
ipvsGenSnapshot -name CDN20_Colorless -dir $PVS_SNAP_SHOTS_DIRS -control ../TECHDIR/PVS/batchSnpGenSetup/Colorless.ctl -log ./CDN20_Colorless.snpGen.log

#ipvs --- select | unselect rules via wildcard through control file
ipvsGenSnapshot -name ColorDenOnly -dir $PVS_SNAP_SHOTS_DIRS -control ../TECHDIR/PVS/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; ?DMO*
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;
```

Virtuoso IPVS User Guide

Batch Commands

Environment Variables

This chapter provides information on Virtuoso IPVS 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:

- [VIPVS Shell Environmental Variables](#) on page 154
- [Virtuoso CDS Environmental Variables](#) on page 158

VIPVS Shell Environmental Variables

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

- [Table 10-1](#) lists shell variables that allow you to customize the display and content of fields available in Virtuoso IPVS Run Option.
- [Table 10-2](#) lists shell variables that allow you to declare default technology setup for VIPVS Snapshot Creator GUI.
- [Table 10-3](#) lists shell variables that allow you to customize the default VIPVS toolbar icons display.
- [Table 10-4](#) lists shell variables that enable VIPVS in debug environment.

Table 10-1 Virtuoso IPVS Run Option GUI Customization

Name	Description	Type	Default
Virtuoso_IPVS_markerBrowser	Specifies the default error browser as PVSDRCDE or AnnotationBrowser.	string	PVSDRCDE
Virtuoso_IPVS_enableSNPCreation	When set to YES, the <i>Create</i> tab will be displayed under the <i>Snapshots</i> field.	boolean	NO
Virtuoso_IPVS_enableSNPManagement	When set to YES, the <i>Manage</i> tab will be displayed under the <i>Snapshots</i> field.	boolean	NO
Virtuoso_IPVS_maintenance	When set to YES, the <i>Upgrade Snapshot</i> button is displayed under the <i>Create</i> tab.	boolean	NO
Virtuoso_IPVS_excludeCellFile	Specifies the default path of exclude cell file.	string	NIL

Virtuoso IPVS User Guide

Environment Variables

Name	Description	Type	Default
Virtuoso_IPVS_data	Specifies global path to VIPVS 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 10-2 Technology Enablement

Name	Description	Type	Default
PVS_SNAP_SHOTS_DIRS	Specifies global snapshot directories as <code><path1>:<path2></code> :.... If declared directory does not exist, VIPVS creates the directory. If it is not specified, snapshot directory is pointed to <code><\$PWD>/ .cadence/ pvs/<\$userid>/ presets/vipvs/ snapshot</code> . Path input can be relative path, absolute path or shell variables.	string	NIL
Virtuoso_IPVS_defaultSNPConfigFile	Specifies the default configurator file that is loaded under the <i>Configurator</i> tab in the <i>VIPVS Snapshot Creator</i> form.	string	NIL

Virtuoso IPVS User Guide

Environment Variables

Name	Description	Type	Default
Virtuoso_IPVS_defaultSNPLayerMap	Specifies the default layermap file that is pre-populated in <i>Layer Map List</i> in the <i>Input</i> tab of the <i>VIPVS Snapshot Creator</i> form.	string	NIL
Virtuoso_IPVS_defaultSNPObjectMap	Specifies the default objectmap file that is pre-populated in <i>Object Name Table</i> in the <i>Input</i> tab of the <i>VIPVS Snapshot Creator</i> form.	string	NIL
Virtuoso_IPVS_defaultSNPRuleDeck	Specifies the default rule deck{s} that is pre-loaded in <i>Rules</i> under the <i>Tech& Rules</i> tab in the <i>VIPVS Snapshot Creator</i> form.	string	NIL

Table 10-3 VIPVS Toolbar Customization

Name	Description	Type	Default
Virtuoso_IPVS_Info	When set to NO, <i>Snapshot Info inquiry</i> triggers are not populated as <i>VIPVS Run Option</i> icon drop-down items.	boolean	YES

Virtuoso IPVS User Guide

Environment Variables

Name	Description	Type	Default
Virtuoso_IPVS_initSnapshot	Specifies the default active snapshot populated in the <i>Snapshot</i> combo field. If it is not found, first item from the available snapshot list is set to active snapshot.	string	NIL
Virtuoso_IPVS_initArea	Specifies the default Verify-Design Area checking as CELL (Whole CellView), VIEWPORT (Visible Area) or AREA (Changed Area).	string	CELL
Virtuoso_IPVS_onlyVD	When set to YES, only the <i>Verify-Design</i> icon is available in the VIPVS toolbar. Otherwise, both <i>Post-Edit</i> and <i>Verify-Design</i> icons are available.	string	YES

Table 10-4 VIPVS Debug Utility

Name	Description	Type	Default
Virtuoso_IPVS_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/PVS_RUNDIR_VD

Virtuoso CDS Environmental Variables

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

- [Table 10-5](#) lists of variables to control error markers counts
- [Table 10-6](#) lists of variables to customize VIPVS verification setting

Table 10-5 Error Markers Filtering

Name	Description	Type	Default
<code>drdBatchVioLimit</code>	Specifies maximum total number of markers reported by VIPVS per run. It is applicable to both of Post-Edit and Verify-Design modes.	int	5000
<code>drdVioLimitPerRule</code>	Specifies maximum number of markers per rules reported by VIPVS. It is applicable to both of Post-Edit and Verify-Design modes.	int	5000

Table 10-6 VIPVS Verification Setting

Name	Description	Type	Default
<code>drdPVSMInPostEditBloat</code>	Specifies minimum area of checking expansion in x and y direction that is larger than the default bloat value stored in the active snapshot. It applies to both Post-Edit and Verify-Design modes.	float	0
<code>drdPostEditClipToViewportOnly</code>	When set to <code>t</code> , Post-Edit view mode is set to <i>Errors Visible in Window</i> , otherwise, it is set to <i>All Errors</i> .	boolean	NIL

Virtuoso IPVS User Guide

Environment Variables

Name	Description	Type	Default
<code>drdPostEditClipToViewportOnlyBloatByValue</code>	It needs to be paired up with <code>drdPostEditClipToViewportOnly</code> . When both of them are set to <code>t</code> , Post-Edit view mode is set to <i>Errors Visible in Window + window halo</i> .	boolean	NIL
<code>drdPostEditClipToViewportOnlyBloatPercent</code>	Specifies halo by percentage of visible window and it needs to be paired up with both <code>drdPostEditClipToViewportOnly</code> and <code>drdPostEditClipToViewportOnlyBloatByValue</code> .	int	0
<code>drdPostEditClipToViewportOnlyBloatValue</code>	Specifies halo by absolute value and it needs to be paired up with both of <code>drdPostEditClipToViewportOnly</code> and <code>drdPostEditClipToViewportOnlyBloatByValue</code> .	float	0
<code>drdPVSEFunctionalPresets</code>	Specifies the default dynamic rule filtering to any of following: <ul style="list-style-type: none"> ■ <i>Default</i> ■ <i>Any Visible</i> ■ <i>Any Visible nodensity</i> ■ <i>Any Visible noconnect</i> ■ <i>Any Visible noconnect nodensity</i> 	string	Default
<code>drdPVSErrorDisplayMaster</code>	When set to <code>t</code> , markers are generated and displayed per instances. In DRCDE, markers are grouped per cell name when it is set to <code>t</code> .	boolean	NIL Note: Starting from ICADV122 / 617 ISR1, the default is set to <code>t</code> .

Virtuoso IPVS User Guide

Environment Variables

Appendix A - Preset to Snapshot Batch Migration

To convert several preset files generated by VIPVS prior to PVS15.14 to its snapshot equivalents, you can run `ipvsGenSnapshot` with `-preset` argument.

Example:

```
ipvsGenSnapshot -name Default -dir . -log default.log -preset
Default.PVS1513.vipvs.preset
```

`ipvsGenSnapshot` will perform conversion of batch preset file `Default.PVS1513.vipvs.preset` to Default snapshot.

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

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

```
ipvs -vipvsSnapshot
```

Or invoke this form with a specific preset file loaded:

```
ipvs -vipvsSnapshot -preset <inputPresetFilePath>
```

To convert several preset files generated by VIPVS prior to PVS15.14 to its snapshot equivalents, you can run `ipvs` 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 `PVS_SNAP_SHOTS_DIRS`

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

```
ipvs -vipvsSnapshot -preset <PresetPath> -any_flow -nograph
```

Virtuoso IPVS User Guide

Appendix A - Preset to Snapshot Batch Migration

Following example shows batch snapshot creation using three presets created prior to PVS15.14:

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 PVS_SNAP_SHOTS_DIRS

- `ipvs -vipvsSnapshot -nograph` execution commands

```
% cat runBatchPrep2Snp
setenv PVS_SNAP_SHOTS_DIRS $PWD/CAD_PRS2SNP_BatchDir
ipvs -vipvsSnapshot -preset ./PRESETS/ColorlessRules.preset -any_flow -nograph
ipvs -vipvsSnapshot -preset ./PRESETS/FDRY1ColorChk_Mx.preset -any_flow -
nograph
ipvs -vipvsSnapshot -preset ./PRESETS/Mx_DecompRules.preset -any_flow -nograph
```