

# **IC Validator VUE**

## **User Guide**

---

Version M-2016.12-SP2, March 2017

**SYNOPSYS®**

# **Copyright Notice and Proprietary Information**

©2017 Synopsys, Inc. All rights reserved. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

## **Destination Control Statement**

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

## **Disclaimer**

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

## **Trademarks**

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at <http://www.synopsys.com/Company/Pages/Trademarks.aspx>.  
All other product or company names may be trademarks of their respective owners.

## **Third-Party Links**

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.  
690 E. Middlefield Road  
Mountain View, CA 94043  
[www.synopsys.com](http://www.synopsys.com)

# Contents

---

|  |      |
|--|------|
| About This User Guide .....                | viii |
| Customer Support.....                      | x    |
| <b>1. Introduction to IC Validator VUE</b> |      |
| VUE Setup .....                            | 1-2  |
| Licensing and Resource Requirements.....   | 1-2  |
| Required Files .....                       | 1-2  |
| Runset Requirements .....                  | 1-3  |
| VUE Execution .....                        | 1-3  |
| VUE Command-Line Options .....             | 1-3  |
| Bindkey Setting.....                       | 1-4  |
| VUE Preference System.....                 | 1-6  |
| VUE Menu Overview .....                    | 1-8  |
| File Menu.....                             | 1-8  |
| View Menu.....                             | 1-9  |
| Tools Menu .....                           | 1-9  |
| VUE Options Dialog Box .....               | 1-10 |
| General Options .....                      | 1-10 |
| DRC/Extraction Options .....               | 1-12 |
| LVS Options .....                          | 1-18 |
| Classification Menu.....                   | 1-19 |
| Windows Menu .....                         | 1-22 |
| Help Menu .....                            | 1-28 |

|  |      |
|--|------|
| VUE DRC Toolbar .....  | 1-28 |
| <b>2. VUE Communications Interface Guide</b>                         |      |
| Graphical User Interface .....                                       | 2-2  |
| Main VUE GUI .....   | 2-2  |
| Communication GUI .....  | 2-3  |
| Tools Menu .....   | 2-3  |
| IC WorkBench EV Plus .....   | 2-4  |
| IC WorkBench EV Plus Command Line .....                              | 2-4  |
| Within IC WorkBench EV Plus .....                                    | 2-4  |
| VUE Command-Line Options .....                                       | 2-5  |
| Communication Window .....   | 2-5  |
| Communicating with IC WorkBench EV Plus Sessions .....               | 2-6  |
| IC Compiler .....  | 2-6  |
| IC Compiler II .....   | 2-6  |
| Custom Compiler .....  | 2-6  |
| Laker Custom Layout System .....                                     | 2-7  |
| Cadence Virtuoso Layout Editor .....                                 | 2-7  |
| Running IC Validator VUE From Virtuoso .....                         | 2-8  |
| VUE Connections in Virtuoso .....                                    | 2-8  |
| Autopopulating VUE Input .....                                       | 2-10 |
| Exporting DFII or OpenAccess Capabilities and Usage Within VUE ..... | 2-14 |
| Exporting CDL Capabilities and Usage Within VUE .....                | 2-17 |
| Postprocessing CDL Netlists Within the IC Validator Run Stream ..... | 2-19 |
| Limitations .....  | 2-20 |
| <b>3. Running IC Validator Within VUE</b>                            |      |
| Executing IC Validator From VUE .....                                | 3-2  |
| Loading IC Validator Results Into VUE .....                          | 3-5  |
| Run Settings File .....  | 3-5  |
| Backward Compatibility .....   | 3-6  |
| Run Summary .....  | 3-6  |
| DRC Errors and Device Extraction Violations .....                    | 3-7  |
| Violation Browser Pane .....   | 3-9  |

|  |      |
|--|------|
| Error List Pane . . . . .                                  | 3-9  |
| Violation Detail Pane . . . . .                            | 3-9  |
| Error Classification Control . . . . .                     | 3-9  |
| Unmatched Error Classification Browser . . . . .           | 3-10 |
| DRC Ease-of-Use Support . . . . .                          | 3-11 |
| Predefined Keyboard Shortcuts . . . . .                    | 3-12 |
| Predefined Mouse Keys . . . . .                            | 3-12 |
| Filter Settings . . . . .                                  | 3-14 |
| Sorting the Error List . . . . .                           | 3-17 |
| Sorting the Violation Browser . . . . .                    | 3-18 |
| Copying the Error List . . . . .                           | 3-19 |
| Searching the Error List . . . . .                         | 3-19 |
| Selecting the Error List . . . . .                         | 3-20 |
| Adjusting Column Location . . . . .                        | 3-21 |
| Switching to Non-Detail Mode . . . . .                     | 3-22 |
| Hiding Function Violations . . . . .                       | 3-23 |
| Iterative Highlighting . . . . .                           | 3-23 |
| The soft_check Graphical Violation Report . . . . .        | 3-24 |
| LVS Errors . . . . .                                       | 3-29 |
| Device Grouping . . . . .                                  | 3-31 |
| Netlist Browser . . . . .                                  | 3-32 |
| Highlight List . . . . .                                   | 3-33 |
| Details Tab . . . . .                                      | 3-36 |
| Net Trace Tab . . . . .                                    | 3-37 |
| Select Port Net . . . . .                                  | 3-40 |
| Netlist Visualizer . . . . .                               | 3-41 |
| Netlist Visualizer Options . . . . .                       | 3-42 |
| Netlist Visualizer Interface . . . . .                     | 3-43 |
| Rendering Objects in the Graphical Canvas . . . . .        | 3-44 |
| Devices . . . . .  | 3-44 |
| Ports . . . . .  | 3-45 |
| Nets . . . . .   | 3-45 |
| Rendered Color for Matched and Unmatched Objects . . . . . | 3-46 |
| Selected Objects . . . . .                                 | 3-46 |
| Highlighted Objects . . . . .                              | 3-47 |
| Progress Bar . . . . .                                     | 3-49 |

|   |      |
|---|------|
| Adding Objects to the Graphical Canvas.....             | 3-50 |
| Adding Objects From the Text Window.....                | 3-50 |
| Adding Objects From the Graphical Canvas .....          | 3-53 |
| Adding Objects From the Equivalence List.....           | 3-55 |
| Interacting With Objects in the Graphical Canvas.....   | 3-55 |
| Pop-Up Menu .....                                       | 3-56 |
| Graphical Canvas Navigation.....                        | 3-57 |
| Bindkey Settings for Netlist Visualizer .....           | 3-58 |
| Layer Debugger .....                                    | 3-58 |
| Setup .....   | 3-59 |
| Layer Debugger Execution Options.....                   | 3-60 |
| Layer Debugger With Selected Debug Objects.....         | 3-61 |
| Running the Layer Debugger.....                         | 3-64 |
| Debugging the Run .....                                 | 3-64 |
| <b>4. Resolving a Short</b>                             |      |
| Short Static Output in Extraction Errors Tab .....      | 4-2  |
| Configuration of Short Static Output .....              | 4-2  |
| Loading Short Static Output.....                        | 4-2  |
| Interactive Short Finding in the Short Finder Tab ..... | 4-3  |
| Text-Based Short Finder .....                           | 4-4  |
| Configuration of the Text-Based Short Finder.....       | 4-7  |
| LVS-Based Short Finder .....                            | 4-21 |
| Configuration of the LVS-Based Short Finder.....        | 4-21 |
| Loading LVS-Based Short Finder Output .....             | 4-21 |

## **Index**

# Preface

---

This preface includes the following sections:

- [About This User Guide](#)
- [Customer Support](#)

---

## About This User Guide

This user guide assists the designer in analyzing and debugging with VUE.

---

### Audience

This user guide is designed to enhance the knowledge that both beginning and advanced users have of the VUE GUI. In addition to providing information about the VUE GUI, this guide discusses how to communicate with layout and schematic tools and how to analyze errors.

---

### Related Publications

For additional information about the IC Validator tool, see the documentation on the Synopsys SolvNet® online support site at the following address:

<https://solvnet.synopsys.com/DocsOnWeb>

You might also want to see the documentation for the following related Synopsys products:

- Custom Compiler™
  - IC Compiler™
  - IC Compiler II™
  - StarRC™
- 

### Release Notes

Information about new features, enhancements, changes, known limitations, and resolved Synopsys Technical Action Requests (STARs) is available in the *IC Validator Release Notes* on the SolvNet site.

To see the *IC Validator Release Notes*,

1. Go to the SolvNet Download Center located at the following address:

<https://solvnet.synopsys.com/DownloadCenter>

2. Select IC Validator, and then select a release in the list that appears.

---

## Conventions

The following conventions are used in Synopsys documentation.

| Convention            | Description  |
|-----------------------|--|
| Courier               | Indicates syntax, such as <code>write_file</code> .  |
| <i>Courier italic</i> | Indicates a user-defined value in syntax, such as <code>write_file design_list</code> .                      |
| <b>Courier bold</b>   | Indicates user input—text you type verbatim—in examples, such as<br><code>prompt&gt; write_file top</code>   |
| [ ]                   | Denotes optional arguments in syntax, such as <code>write_file [-format fmt]</code>                          |
| ...                   | Indicates that arguments can be repeated as many times as needed, such as<br><code>pin1 pin2 ... pinN</code> |
|                       | Indicates a choice among alternatives, such as<br><code>low   medium   high</code>                           |
| Ctrl+C                | Indicates a keyboard combination, such as holding down the Ctrl key and pressing C.                          |
| \                     | Indicates a continuation of a command line.  |
| /                     | Indicates levels of directory structure.   |
| Edit > Copy           | Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.                         |

---

---

## Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Center.

---

### Accessing SolvNet

The SolvNet site includes a knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. The SolvNet site also gives you access to a wide range of Synopsys online services including software downloads, documentation, and technical support.

To access the SolvNet site, go to the following address:

<https://solvnet.synopsys.com>

If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to sign up for an account.

If you need help using the SolvNet site, click HELP in the top-right menu bar.

---

### Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a support case to your local support center online by signing in to the SolvNet site at <https://solvnet.synopsys.com>, clicking Support, and then clicking “Open A Support Case.”
- Send an e-mail message to your local support center.
  - E-mail [support\\_center@synopsys.com](mailto:support_center@synopsys.com) from within North America.
  - Find other local support center e-mail addresses at  
<http://www.synopsys.com/Support/GlobalSupportCenters/Pages>
- Telephone your local support center.
  - Call (800) 245-8005 from within North America.
  - Find other local support center telephone numbers at  
<http://www.synopsys.com/Support/GlobalSupportCenters/Pages>

# 1

## Introduction to IC Validator VUE

---

*This chapter provides an introduction to IC Validator VUE usage, including runset requirements, available command-line options, and general VUE setup options.*

IC Validator VUE is a flow-based graphical tool that guides you through the entire physical verification flow. Within one interface, you can configure and execute a verification run, easily load the results, review a run summary, and debug the design by highlighting errors within most layout editor tools.

For more information, see the following sections:

- [VUE Setup](#)
- [VUE Execution](#)
- [VUE Command-Line Options](#)
- [Bindkey Setting](#)
- [VUE Preference System](#)
- [VUE Menu Overview](#)

---

## VUE Setup

This chapter contains the setup instructions for VUE. Make sure that you have incorporated the proper licensing information, that you have all of the correct directories and files, and that your environment accommodates VUE.

---

### Licensing and Resource Requirements

VUE is licensed separately from IC Validator functions. For more information about licensing software, native licensing, and resource requirements, see Chapter 2, “Licensing and Resource Requirements,” in the *IC Validator User Guide*.

When using native licensing, ICValidator2-Manager is the main license. With this license, IC Validator checks only for the presence of a valid license every time VUE is run.

The Netlist Visualizer features requires native licensing.

---

### Required Files

A number of files are required to run VUE and to connect with different interfaces. The installation package contains all of the required files to run VUE and to connect with different interfaces. It does not require any setup beyond running the Synopsys Common Installer.

Note:

The IC Validator and VUE release versions must be the same to ensure compatibility.

You should install the following VUE files in the install directory (ICV\_HOME\_DIR), shown in [Table 1-1](#):

*Table 1-1 Required VUE Files*

| VUE file                             | Description                                  |
|--------------------------------------|--|
| \$ICV_HOME_DIR/bin/systype/icv_vue   | VUE binary                                   |
| \$ICV_HOME_DIR/etc/VUE/*             | VUE command files                            |
| \$ICV_HOME_DIR/etc/tcl-u/IcwbVue.tcl | Tcl code for IC WorkBench EV Plus connection |
| \$ICV_HOME_DIR/lib-u/systype/VUE/*   | VUE connection filter files                  |

---

## Runset Requirements

To ensure that all data required by VUE is generated,

- Set the `create_vue_output` argument in the `error_options` function:

```
error_options (
    create_vue_output = true
);
```
- Set the `-vue` command-line option.

---

## VUE Execution

VUE can be started either on the shell command line or within the layout editor. For more information about starting VUE from within a layout editor, see [Chapter 2, “VUE Communications Interface Guide.”](#)

---

## VUE Command-Line Options

The VUE command-line syntax is

```
icv_vue [command-line-options] runsetfile
```

where `runsetfile` specifies the executable runset file.

The command-line options are described in [Table 1-2](#).

*Table 1-2 VUE Command-Line Options*

| Argument               | Description  |
|------------------------|--|
| <code>-64</code>       | Runs VUE with 64-bit coordinates support. The data to be viewed in VUE must be the result of an IC Validator run that used the <code>-64</code> command-line option. |
| <code>-cmdPrint</code> | Toggle the command print-out.  |
| <code>-hlic</code>     | Force shared licensing scheme.   |
| <code>-ilic</code>     | Force native licensing scheme.   |
| <code>-iconic</code>   | Start VUE iconified.   |

*Table 1-2 VUE Command-Line Options (Continued)*

| <b>Argument</b>                        | <b>Description</b>  |
|--|---|
| <code>-lay layTool</code>              | Layout tool on which to connect during startup. Supports Custom Compiler™, IC Compiler™, IC WorkBench EV Plus, Laker™ Custom Layout System, and Cadence® Virtuoso® Layout Editor.   |
| <code>-layArgs argName argValue</code> | Sets layout editor specific arguments required for connection on the command line, if needed. Details on allowable options for each layout tool is available in <a href="#">Chapter 2, “VUE Communications Interface Guide.”</a>  |
| <code>-ld</code>                       | Runs VUE in Layer Debugger mode.  |
| <code>-load vueFile</code>             | Autoload the information found in the <code>cell.vue</code> file. By default, no data file is loaded on VUE startup. For more information about where the <code>.vue</code> file is defined, see the “Run Settings File” section in <a href="#">Chapter 3, “Running IC Validator Within VUE.”</a> |
| <code>-schArgs argName argValue</code> | Assign arguments and their values that are needed when VUE connects to the schematic editor.  |
| <code>-sch schTool</code>              | Schematic tool on which to connect during startup. Supports Composer and CosmosSE™.   |
| <code>-usage   -h   -help</code>       | Display usage message and exit.   |
| <code>-v</code>                        | Print product version.  |
| <code>-Version</code>                  | Print product and library version.  |

---

## Bindkey Setting

The bindkey setting allows you to specify, or bind which keys are used to view the next or previous DRC (design rule checking) error in VUE. This capability is implemented as follows:

- Load the error output data from IC Validator into VUE. Click the first error in the Violations pane. VUE zooms to that error location in the layout editor.
- Fix the error in the layout editor.
- Press the next violation designated bindkey to enable VUE to zoom to the next error in the layout editor.

The Virtuoso and the Laker system interfaces support bindkeys. Set your own bindkeys in Virtuoso by following these steps:

- Create or modify the bindkey setting file, for example, `myBindKey.il`. The content of the skill file might look like

```
hiSetBindKey(window, key_or_combination, function)
hiSetBindKey("Layout", "Ctrl<Key>Down", "vueDrcNextViolation()")
hiSetBindKey("Layout", "Ctrl<Key>Up", "vueDrcPrevViolation()")
```

Modify only the `key_or_combination` portion of the file. Do not modify the window, function, or command portions of the file. In this case, press Ctrl+Down Arrow key to instruct `icv_vue` to select the next DRC violation listed in the `icv_vue` Violations pane. Press Ctrl+Up Arrow key to instruct `icv_vue` to select the previous DRC violation.

- Launch Virtuoso and load the bindkey skill file.
- Launch the `icv_vue` tool and load the results. After you select a cell from the `icv_vue` Layout Cell pane, use your own bindkey in Virtuoso to track the DRC violations.

Set your own bindkeys in the Laker system by following these steps:

- Create or modify the bindkey setting file, for example, `myBindKey.key`. The content of the file might look like

```
{
key_or_combination window function TCL_name / TCL_procedure
Ctrl<Key>Down * f.tcl LakerFilter ICV::VUEDrcNextViolation
Ctrl<Key>Up * f.tcl LakerFilter ICV::VUEDrcPrevViolation
}
```

Modify only the `key_or_combination` portion of the file. Do not modify the window, function, and command portions of the file. In this case, press Ctrl+Down Arrow key to instruct `icv_vue` to select the next DRC violation listed in the `icv_vue` Violations pane. Press Ctrl+Up Arrow key to instruct `icv_vue` to select the previous DRC violation.

- Create a .tcl file, for example, `vueBindKey.tcl`. The content of the .tcl file might look like  
`gtSetBindKey -windowType lakerDsgWnd -file ./myBindKey.key`
- Launch the Laker system and source the .tcl file. For example, source  
`./vueBindKey.tcl`.
- Launch `icv_vue` and load the results. After you select a cell from the `icv_vue` Layout Cell pane, use your own bindkey in the Laker system to track the DRC violations.

---

## VUE Preference System

All available user settings are categorized into three parts: application settings, run settings, and central enforcing settings.

- **Application Settings**

- Path: `~/.config/Synopsys/icv_vue.conf`

Like most of the windows applications, user-modified settings are retained and applied automatically to the next run. The application settings for the configuration of the GUI include the following:

- Window sizing
  - Docking widget layouts
  - Highlighting on/off
  - Color/fill selections

- Preconfiguration using `VUE_APP_SETTINGS_PATH` environment variable example:

```
setenv VUE_APP_SETTINGS_PATH ~/icv_vue.conf
```

- Saves and loads automatically

- **Run settings**

- Path:
    - `./icv_vue.runs` (same path as VUE startup)
    - `.icv_vue.runs` (same path as *block.vue*)

- Settings from those windows:

- Execution window shows all settings
    - DRC Errors window settings
      - > Filter violations setting
      - > DRM link setting
        - Rule Name Definition. Specifies the rule name used in the run settings file.

The `.icv_vue.runs` path refers to the violation comments to the left of the semicolon (`:`).

For example, in the violation comment,

```
R.Metal2.W : metal2 Minimum width must be > 0.22
```

the rule name is `R.Metal2.W` by default.

### - Syntax

```
[DRM_Description]
Rule_Name =DRM_File
[DRM_Diagram]
Rule_Name =DRM_File
[DRM_Custom]
Rule_Name =DRM_File
```

### - Example

```
[DRM_Description]
M1.DN.2 = ../doc/abc.pdf#page=7
M1.DN.1 = ../doc/abc.pdf#page=9
try.Tox_W=../doc/abc.pdf#page=11

[DRM_Diagram]
try.Metall.W=../doc/abc.pdf#page=1

[DRM_Custom]
ntry.Tox.W=../doc/abc.pdf#page=5
```

#### Note:

To use the web-based DRM file or browser, set the following environment variable:

```
setenv VUE_DRM_USE_BROWSER
```

The web-based DRM file format is as follows:

M1.DN.1= <https://xxx.xxx.xxx/xxx.pdf#page=xx>

- LVS Errors window shows layer visibility
- Saves manually:
  - DRC Errors window > right-click the Violation Browser > Filter Settings> Save
  - VUE File menu > Save Run Settings
- Loads automatically or manually:
  - Loads automatically. The order is ./icv\_vue.runs, then .icv\_vue.runs
  - Load manually. DRC Errors window > right-click the Violation Browser > Filter Settings > Load
  - VUE File menu > Load Run Settings
- **Central enforcing settings:**
  - Path: \$ICV\_HOME\_DIR/etc/VUE/.icv\_vuer
  - Disables certain VUE functionalities:

- Usually edited by CAD or IT to disable certain VUE functionalities from third-party users.
- Overrides incrementally the same settings by previous setting files.
- Currently supports enabling and disabling of DRC classification.
  - Edits and saves manually.
  - Loads automatically

---

## VUE Menu Overview

Information about the main VUE menu is provided in following sections.

---

### File Menu

The File menu contains several commonly used administrative commands. You can

- Load Run Settings
- Save Run Settings
- Close the current project
- Exit VUE

[Table 1-3](#) describes these file menu commands.

*Table 1-3 File Menu Commands*

| File menu command | Description  |
|-------------------|--|
| Load Run Settings | Loads a valid VUE run settings file. The default is <code>.icv_vue.runs</code> . See the " <a href="#">VUE Preference System</a> " on page 1-6 for more information about the run settings                           |
| Save Run Settings | Creates a VUE run settings file in the directory you specify. The default is <code>.icv_vue.runs</code> . See the " <a href="#">VUE Preference System</a> " on page 1-6 for more information about the run settings. |
| Close             | Unloads all run data from VUE, removing any necessary tabs. The Execution and Load tabs remain.  |
| Exit              | Unloads all run data and VUE shuts down.   |

---

## View Menu

The View menu allows you to view the run, layout errors, and LVS errors summaries.

[Table 1-4](#) describes the View menu commands.

*Table 1-4 View Menu Commands*

| View menu command     | Description                      |
|-----------------------|----------------------------------|
| Run Result            | <i>cell.RESULTS</i> file         |
| Run Summary           | <i>run_details/cell.sum</i> file |
| Layout Errors Summary | <i>cell.LAYOUT_ERRORS</i> file   |
| LVS Errors Summary    | <i>cell.LVS_ERRORS</i> file      |

---

## Tools Menu

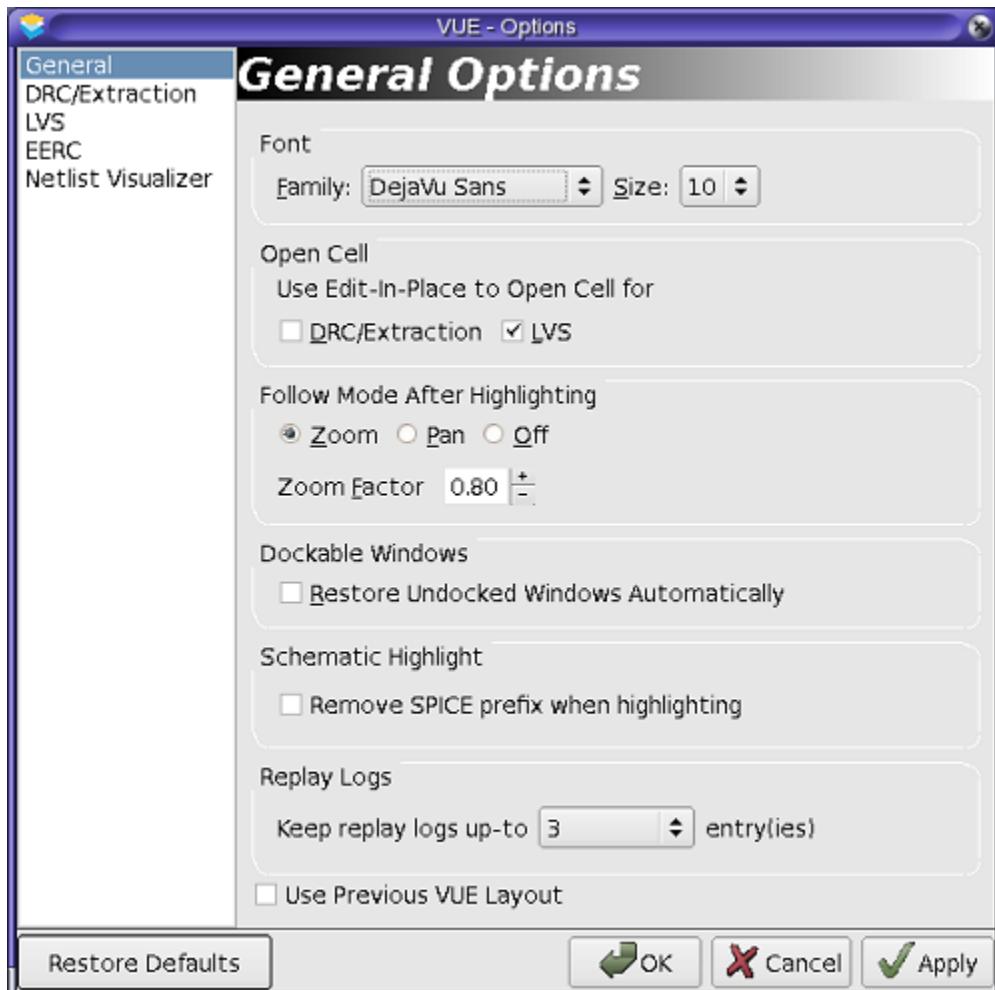
The Tools menu allows you to change viewing options or set the layout connection. Instructions for setting up the connection to layout tools are discussed earlier in this chapter. For more information about the Communication window, see [Chapter 2, “VUE Communications Interface Guide.”](#)

## VUE Options Dialog Box

As shown in [Figure 1-1](#), the VUE Options dialog box allows you to set how VUE displays errors or highlights objects.

### General Options

*Figure 1-1 General Options*

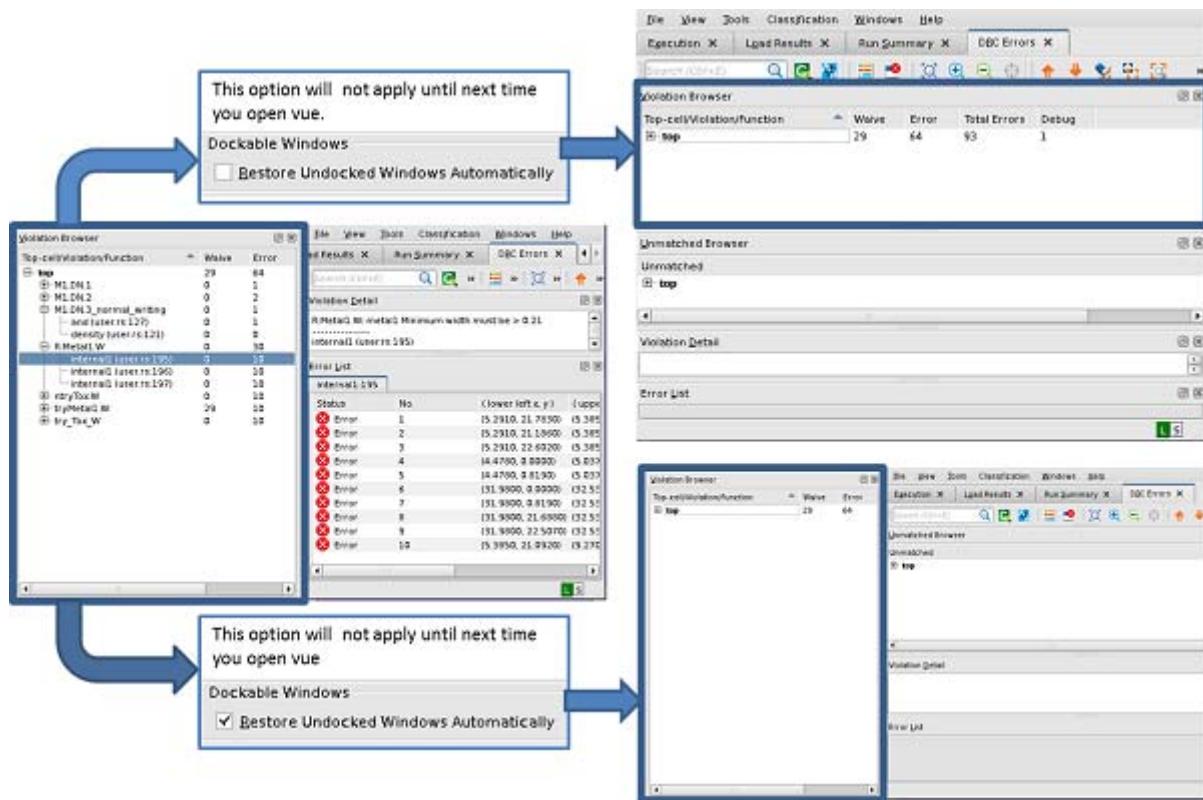


- **Font.** Specifies the font of all text displayed by VUE. The default font type is DejaVu Sans, size 10.
- **Open Cell.** Specifies whether cells are opened in Edit-In-Place mode for DRC and LVS (layout versus schematic) errors. The Edit-In-Place mode opens a cell within the context of its parent. If this option is disabled, VUE opens only the cell.

- **Follow Mode.** Specifies how VUE controls the layout window when highlighting objects. There are three options available:
  - **Zoom.** Specifies that VUE should center the highlight in the window with the specified zoom factor. The closer the zoom factor is to one, the closer VUE zooms to the error. The default zoom factor is 0.8.
  - **Pan.** The layout editor pans to the center of the error marker, but keeps the zoom level at its current setting in the layout editor.
  - **Off.** Neither the Pan nor the Zoom of the layout editor is affected by selecting a new error marker. With this setting, it is possible that the error marker is drawn outside of the currently viewable area of the layout editor.
- **Dockable Windows.** Specifies whether the undocked windows are automatically restored. The default is off.

[Figure 1-2](#) shows the Dockable Windows feature in the General Options window.

*Figure 1-2 General Options: Example of Dockable Windows*



- **Schematic Highlight.** Specifies whether to remove the SPICE prefix when highlighting in a third-party schematic editor. The default is off.
- **Replay logs.** Specifies how many replay files are recorded. The default is 3.

- **Use previous VUE layout.** Specifies whether you revert to the previous VUE layout.

## DRC/Extraction Options

The DRC/Extraction Options page is shown in [Figure 1-3](#) and [Figure 1-4](#).

*Figure 1-3 DRC/Extraction Options Page*

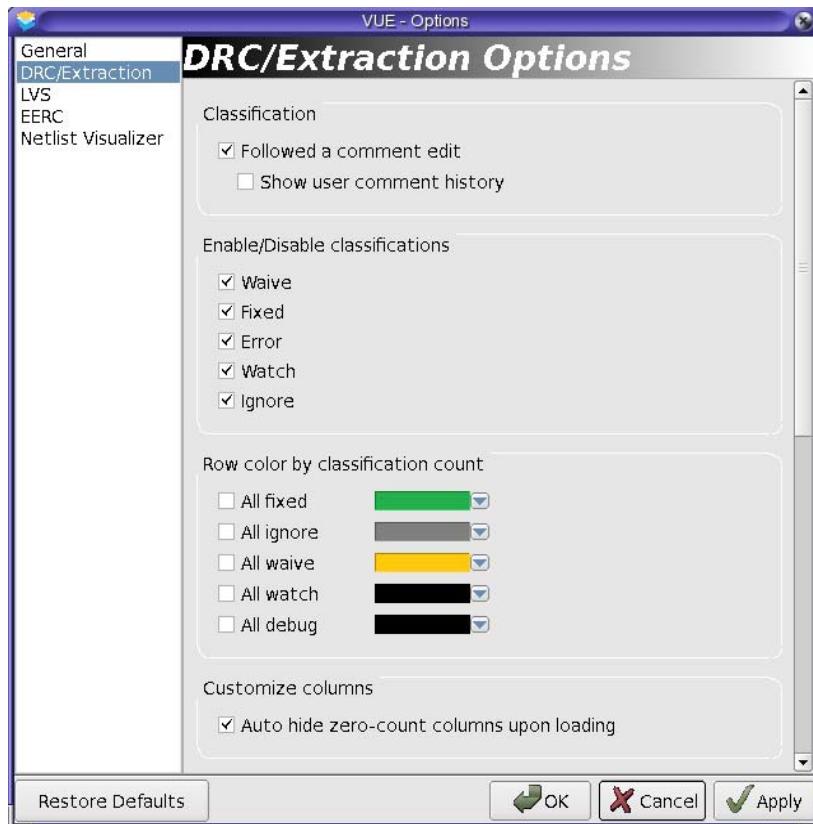
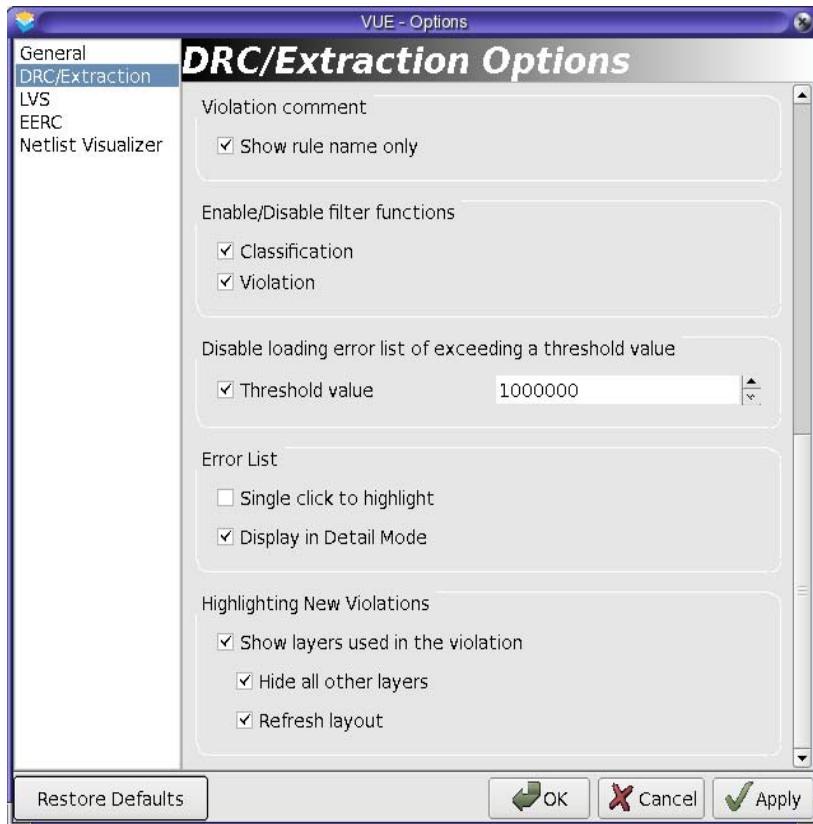


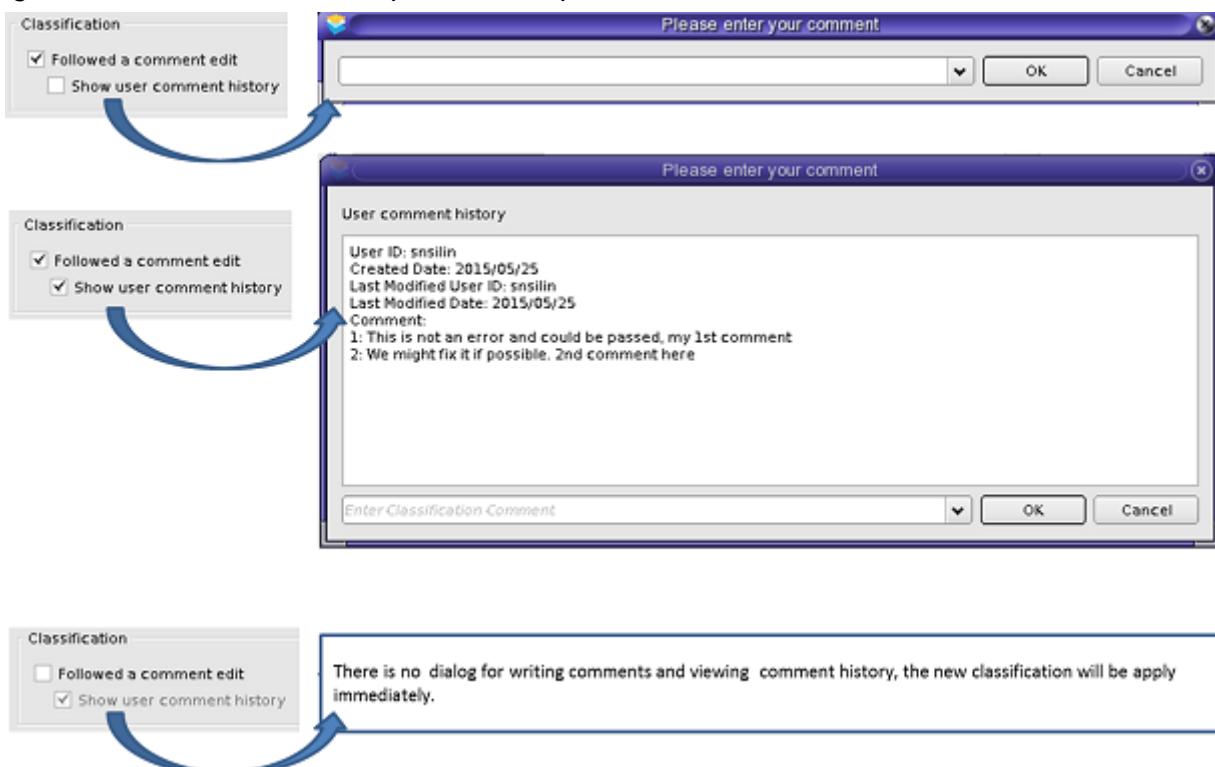
Figure 1-4 DRC/Extraction Options Page Part II



- **Classification.** Specifies whether comments of the selected error in the edit pane follow from the classification and show the history of comments. The default is “Followed a comment edit,” but does not show the user comment history.

[Figure 1-5](#) shows an example of the classification selections.

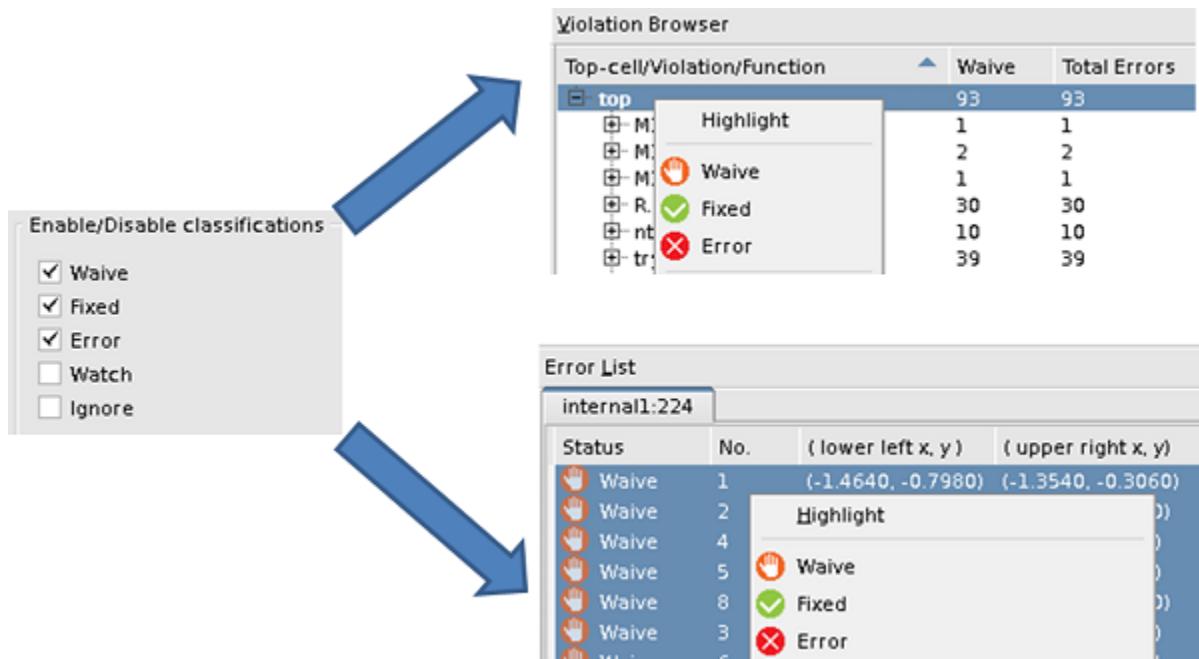
Figure 1-5 DRC/ Extraction Options: Example of Classification



- **Enable/Disable classifications.** Specifies whether the classification types are available during classification. The default is to set all classification types to enabled to be classified.

Figure 1-6 shows the Enable/Disable classifications of the DRC/Extraction options.

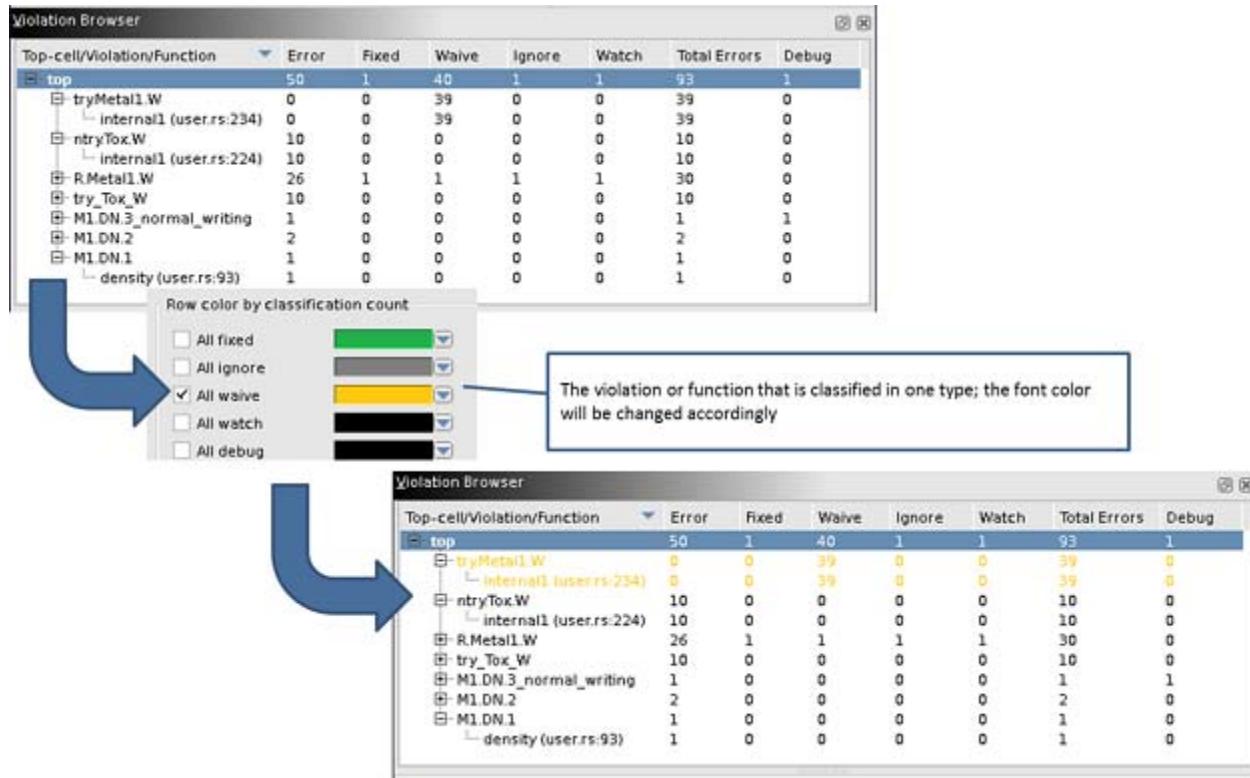
*Figure 1-6 DRC/ Extraction Options: Example of Enable/Disable Classifications*



- Row color by classification count. Specifies whether the color for the entire violation is to be classified as one type. The default is that no colors are set.

[Figure 1-7](#) shows the “Row color by classification count” selection.

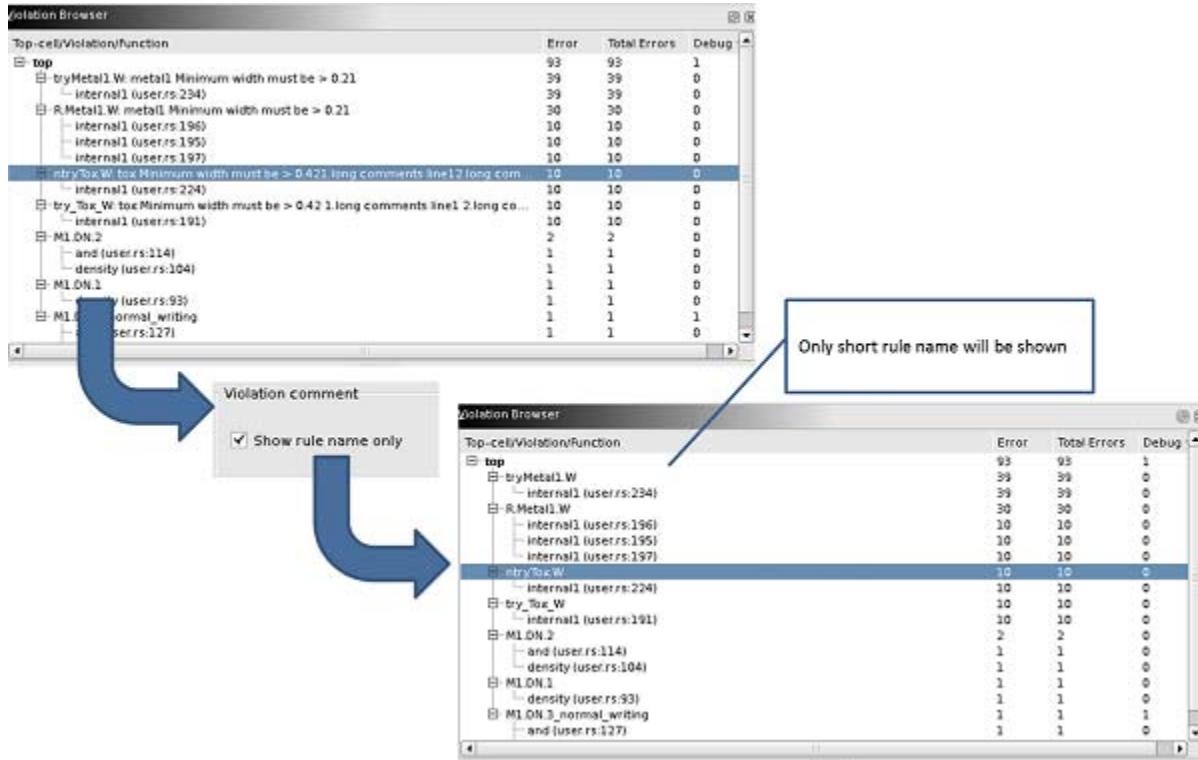
Figure 1-7 DRC/ Extraction: Row color by classification count



- Customize columns.** Specifies whether the zero-count columns are displayed upon loading. The default is enable.
- Violation comment.** Specifies whether only the rule name is shown. The default is off.

Figure 1-8 shows the “Show rule name only” selection.

Figure 1-8 DRC/ Extraction: Show rule name only



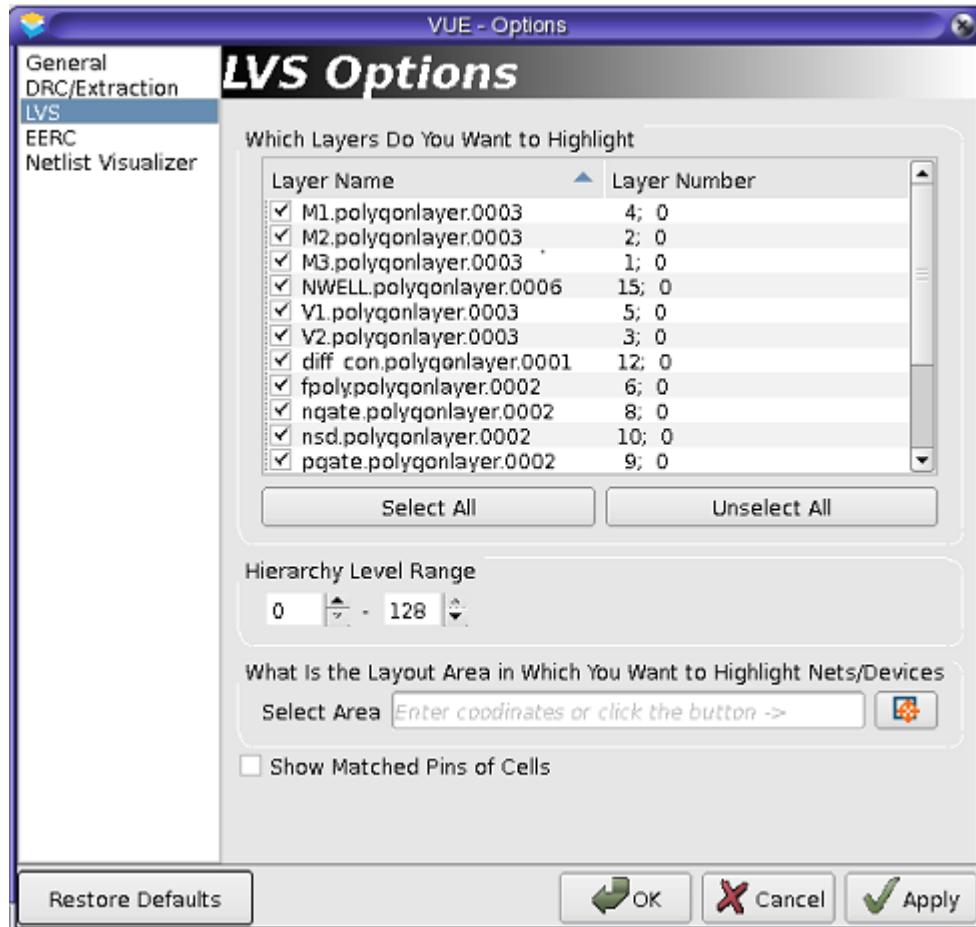
- **Enable/Disable filter functions.** Specifies whether to enable the filter function in the violation browser. The default is to enable both the classification and violation in the violation browser.
- **Disable loading error list of exceeding a threshold value.** Specifies whether enabling the loading for a given violation error depends on the threshold value. The default is to enable the threshold with equal to or more than 1000000 errors.
- **Error list.**
  - **Single click to highlight.** Specifies whether single select is highlighting. The default is off, which means you right-click or highlight the previous/next function.
  - **Display in Detail Mode.** Specifies whether to show the default information in the error list. The default is on.
- **Show Layers used in the violation.** The default is off. By checking this option, as shown in [Figure 1-4](#), you can highlight DRC violations (from within the DRC violation browser or error list), and corresponding layers used by the highlighted violation are turned on in the layout editor (for example, in Virtuoso).
- **Hide all other layers.** The default is off. Specifies whether to hide irrelevant layers that are not used by the highlighted violation.

- **Refresh layout.** The default is on after checking the “Show Layers used in the violation option”. Specifies whether to refresh the layout viewer after turning on or off the layout editor layers.

## LVS Options

The LVS Options page is shown in [Figure 1-9](#).

*Figure 1-9 LVS Options Page*



- **Which Layers Do You Want to Highlight.** Specifies which layers of a net are highlighted. This is useful when you want to view the routing layer of a net to see some connections better. All of the layers in the device matrix connect sequence are listed here.
- **Hierarchy Level Range.** Limits the net highlight to a specific range of hierarchical levels. This improves highlighting time.

- **Select Area.** Enter lower-left and upper-right coordinates of a region to highlight. Alternately, you can define the region in the layout editor by using the button to the right of the text box.
- **Show matched pins of cells.** Controls the level of detail printed for diagnostic messages. By default, VUE shows only the pins that connect to unmatched nets. When you select this option, VUE shows all pin connections.

## Classification Menu

The Classification menu allows you to

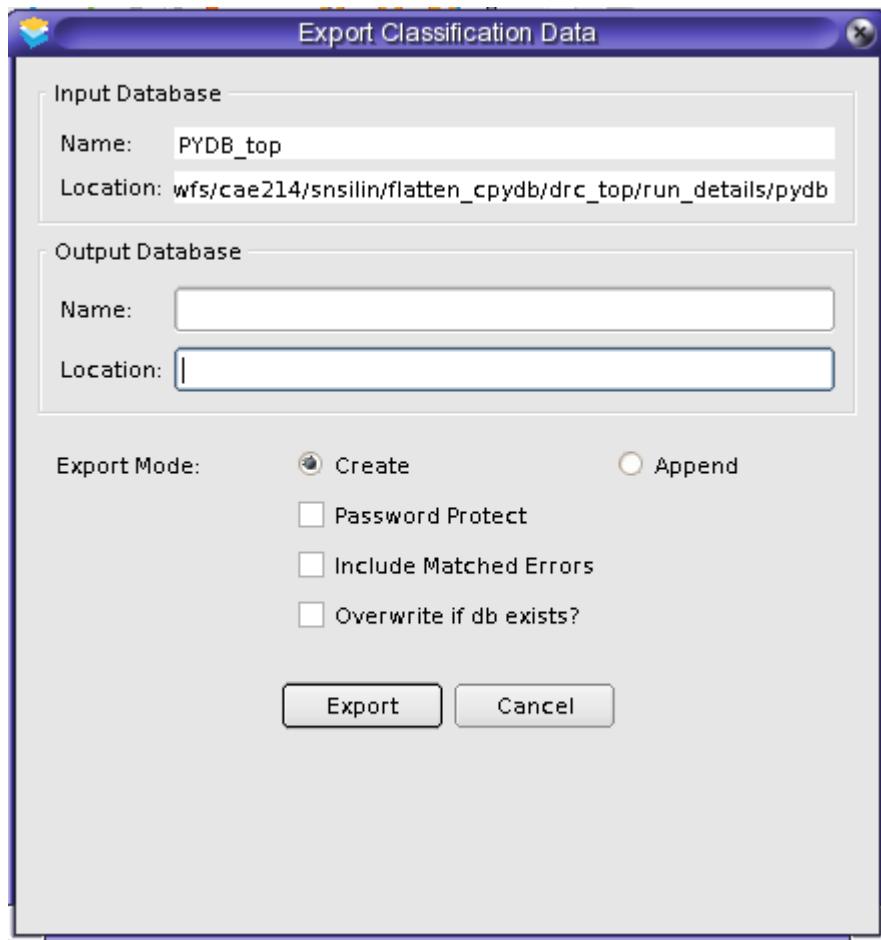
- Export classification data for use in subsequent IC Validator runs.
- Generate classification reports to review the current status of errors in the design.

See Chapter 10, “DRC Error Classification,” in the *IC Validator User Guide* for more information about using an exported classification database in IC Validator to retain classification information between IC Validator runs.

### Export Classification Data

This menu item launches the dialog box, such as the one provided in [Figure 1-10](#), that specifies the creation of an error classification database. This database can be used by IC Validator in subsequent runs to maintain classification information, such as status or user comments, between runs.

Figure 1-10 Export Classification Data



The IC Validator error database currently loaded in VUE DRC is listed at the top of the dialog box. Enter the output name and location, using a name that is different from the input library name. (Note that the default is the current working directory.)

The Create and Append export modes are output methods for exporting classification data.

- **Create.** This mode creates a new error classification database.
- **Password Protect.** A newly created classification database can be password protected. For another user to alter or append to the created database, the correct password must be entered.
- **Include Matched Errors.** By default, when exporting to a new classification database, errors matched from a previous IC Validator run are not included in the new database. This is done in to reduce the amount of duplicated data as these errors exist in a separate classification database. To export all errors, despite being matched, check the Include Matched Errors check box.

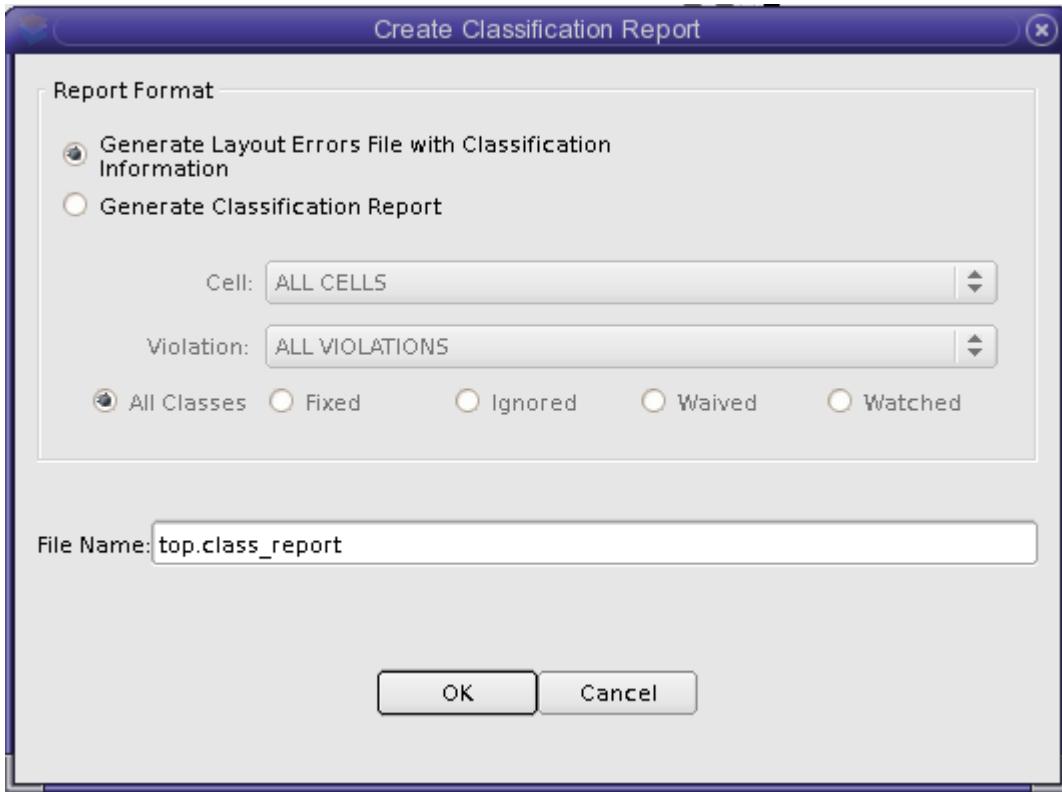
- **Overwrite if db exists?**. If checked, VUE DRC removes the previous classification database with the same name and creates a new one. All data from the previous database is lost. If you are overwriting an existing database that is password protected, you are required to enter the correct password.
- **Append**. If a classification database exists, you can append the changes to it with the Append export mode. If the database is password protected, a password must be verified before the export is successful.

See Chapter 8, DRC Error Classification, of the *IC Validator User Guide* for more information about how to use the classification database in subsequent IC Validator runs.

### Generate Classification Report

Two types of classification reports can be generated: an annotated LAYOUT\_ERRORS file and a standalone classification report. VUE DRC does not display these reports when they are created. [Figure 1-11](#) is an example of a classification report.

*Figure 1-11 Generated Classification Report*



The annotated LAYOUT\_ERRORS file is in the same format as the LAYOUT\_ERRORS file generated during the IC Validator run, and it includes additional information regarding all

classification data that has been sent to the error database. This includes all classified and unclassified errors in the database.

A standalone classification report can be generated that displays only the error classification data. This report can be customized to report certain cells, violations, and classification types by using the options available within the window.

See Chapter 8, “DRC Error Classification,” in the *IC Validator User Guide* for examples of the report.

---

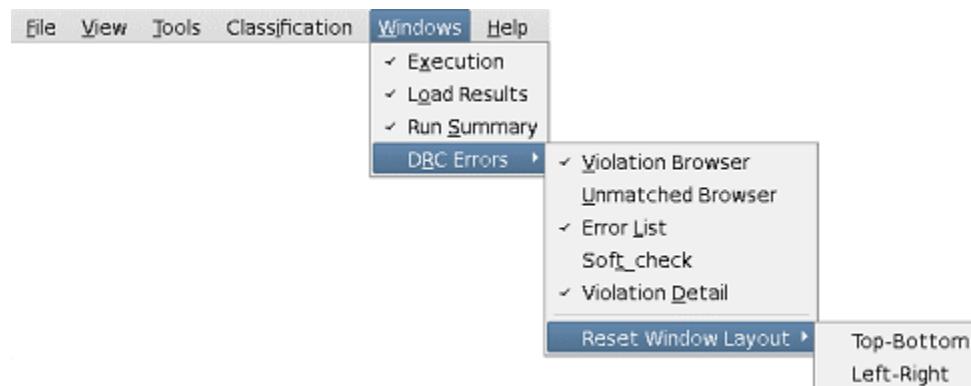
## Windows Menu

The Windows menu lists the windows and widgets that are shown in VUE interface. A customized work environment allows you to:

- Dock and undock a widget
- Remove or recover a window
- Remove or recover a widget
- Merge multiple widgets to create one window
- Move the Violation Detail selection from the bottom to the middle of the list.

[Figure 1-12](#) shows the DRC errors in the Windows menu.

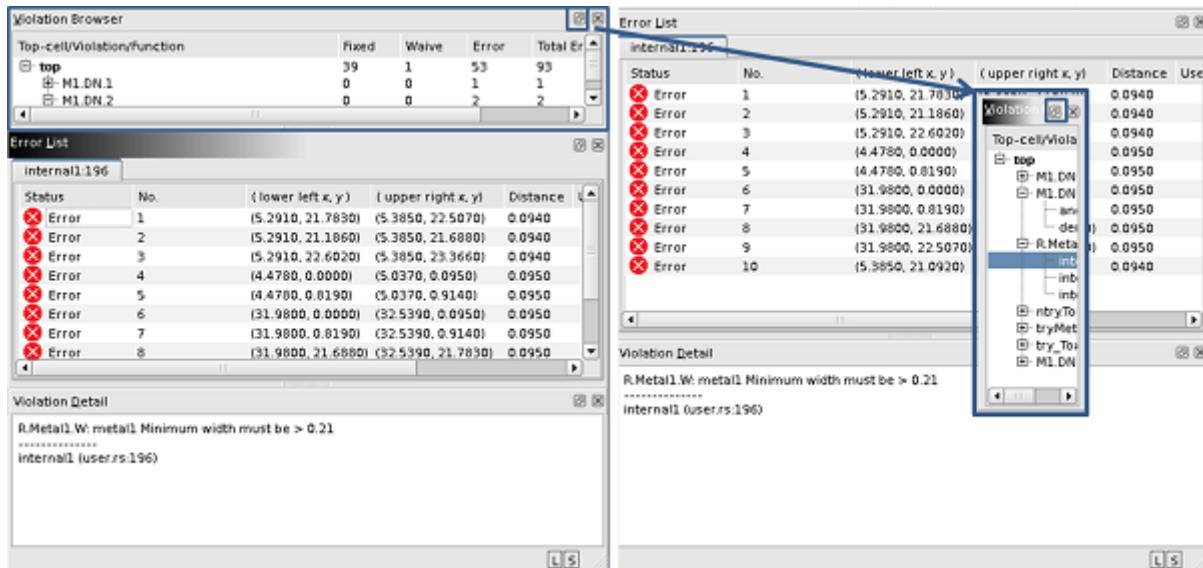
*Figure 1-12 Windows Menu*



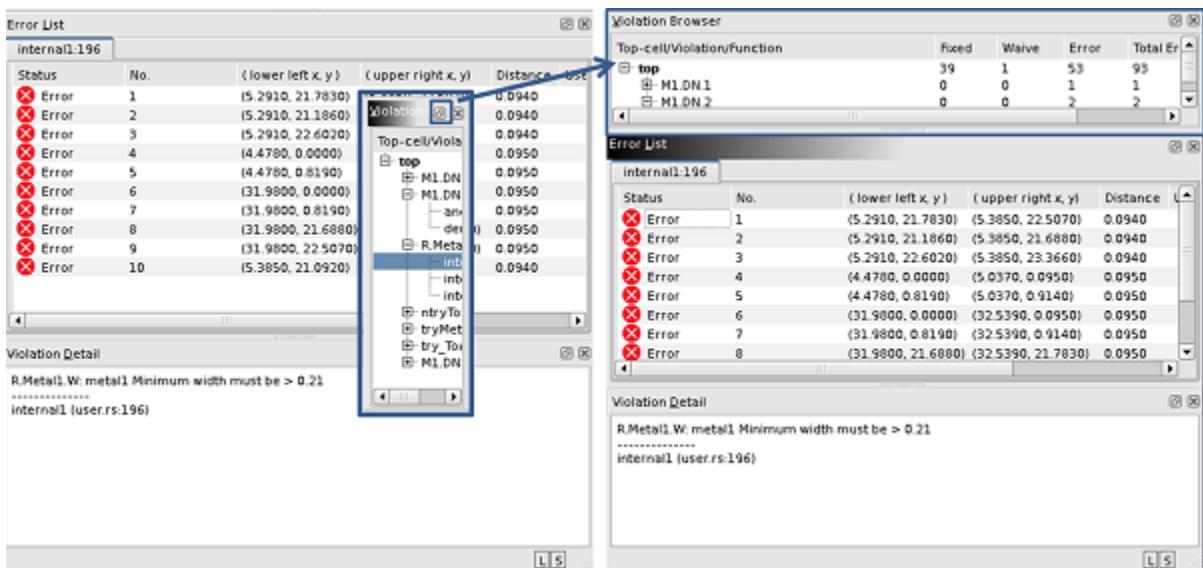
The dockable windows feature enables you to dock and undock certain windows from the VUE interface. The windows are identified by an Undock button in the right corner of each pane.

[Figure 1-13](#) and [Figure 1-14](#) show examples of docking and undocking a widget.

**Figure 1-13 Windows Menu: Undocking a Widget**



**Figure 1-14 Windows Menu: Docking a Widget**



The removable windows feature enables you to remove and recover certain windows from the VUE interface. [Figure 1-15](#) and [Figure 1-16](#) show examples of removing and recovering a window.

Figure 1-15 Windows Menu: Removing a Window

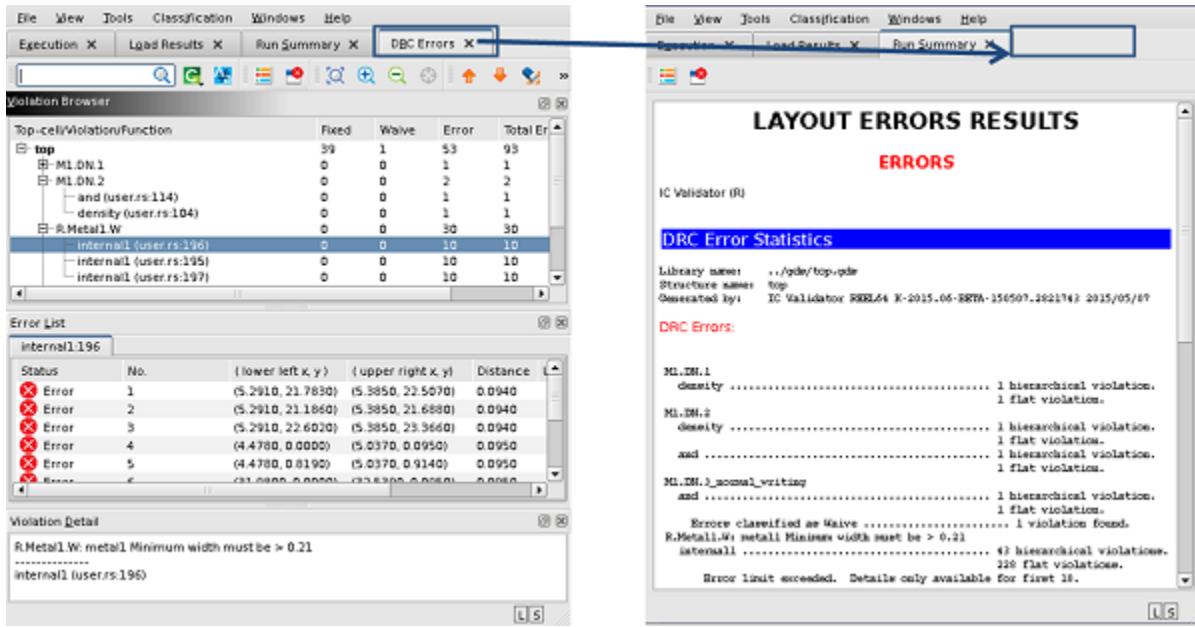
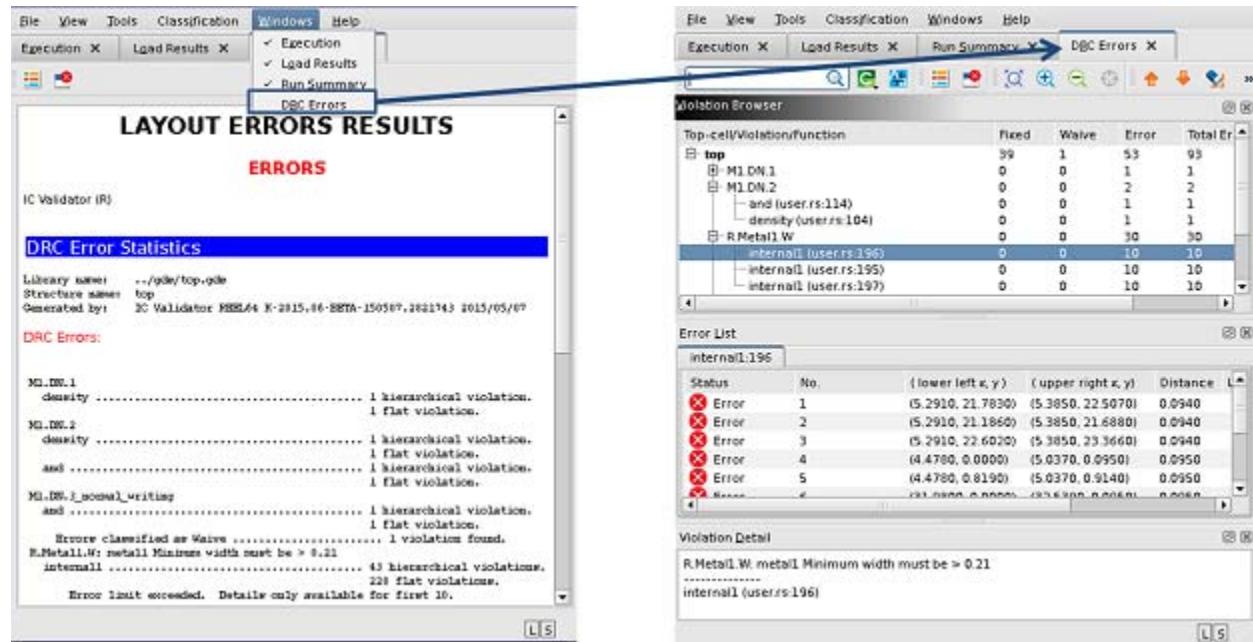


Figure 1-16 Windows Menu: Recovering a Window



The removable windows feature also enables you to remove and recover certain widgets from the VUE DRC interface. [Figure 1-17](#) and [Figure 1-18](#) show examples of removing and recovering a widget.

Figure 1-17 Windows Menu: Removing a Widget

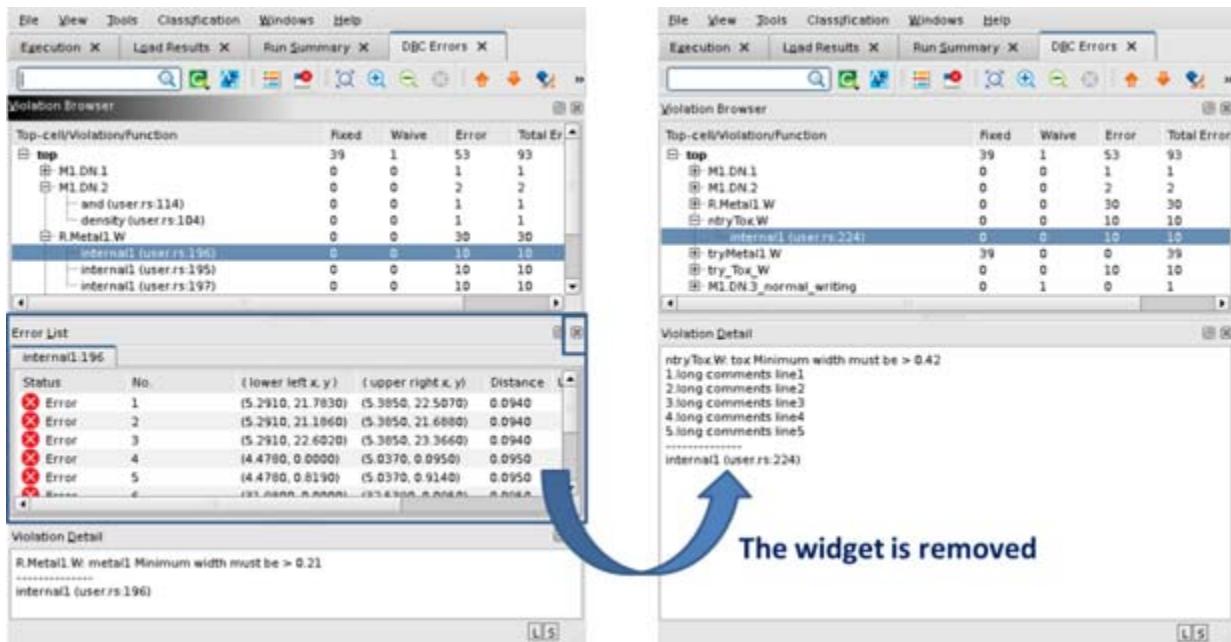
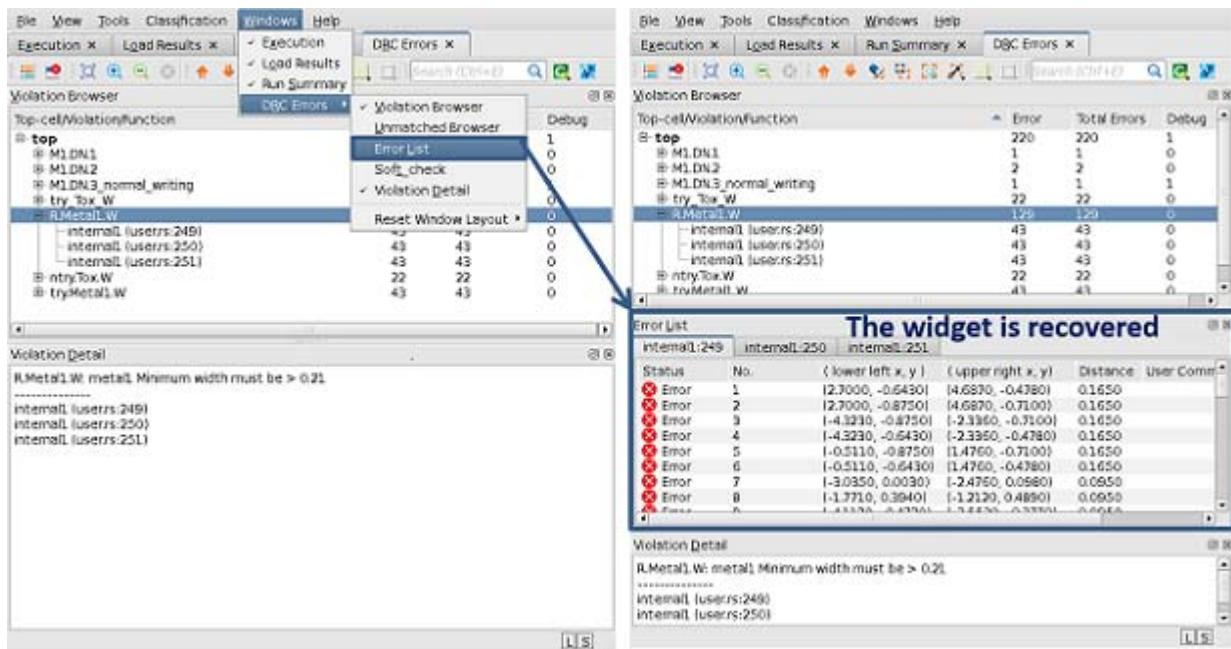


Figure 1-18 Windows Menu: Recovering a Widget



[Figure 1-19](#) and [Figure 1-20](#) show examples of merging multiple widgets into one table and splitting widgets from the table.

Figure 1-19 Windows Menu: Merging Multiple Widgets Into One Table

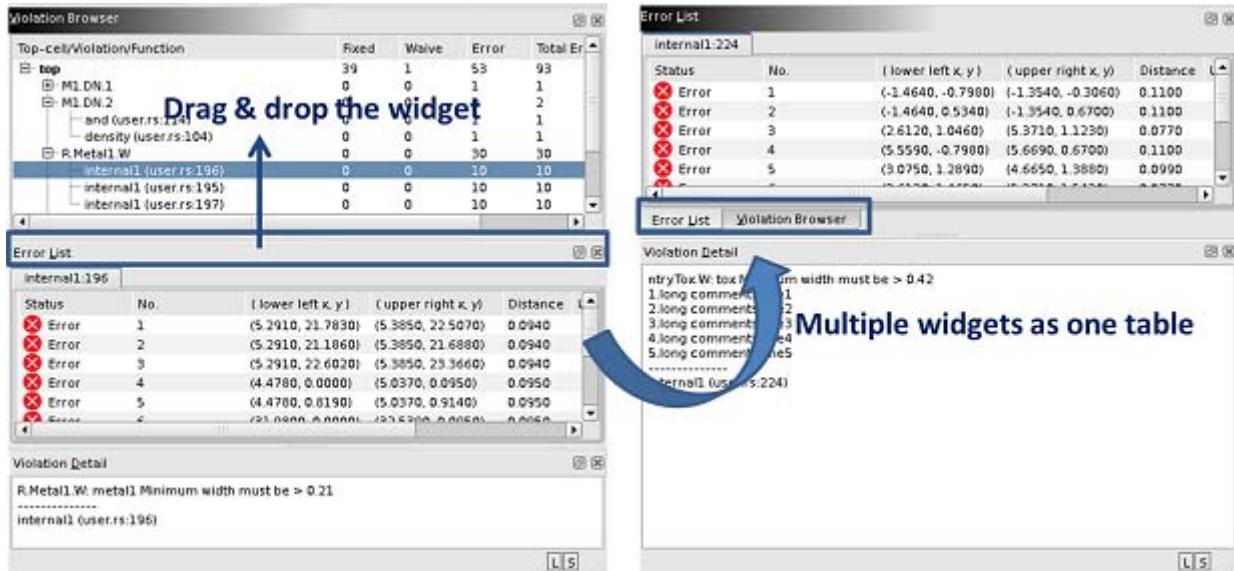
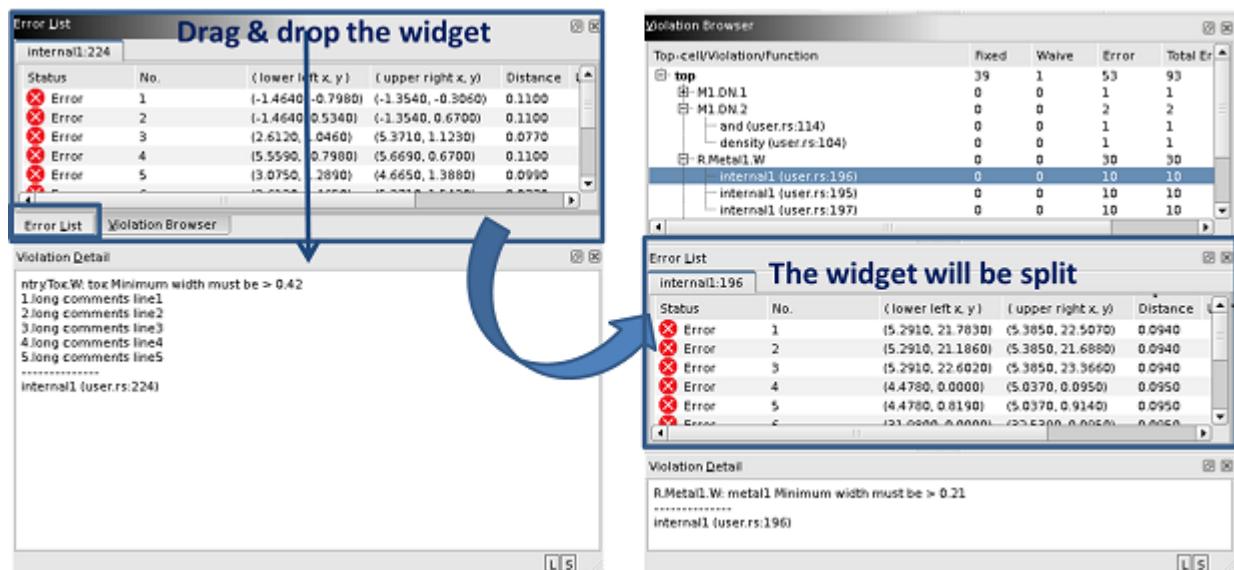
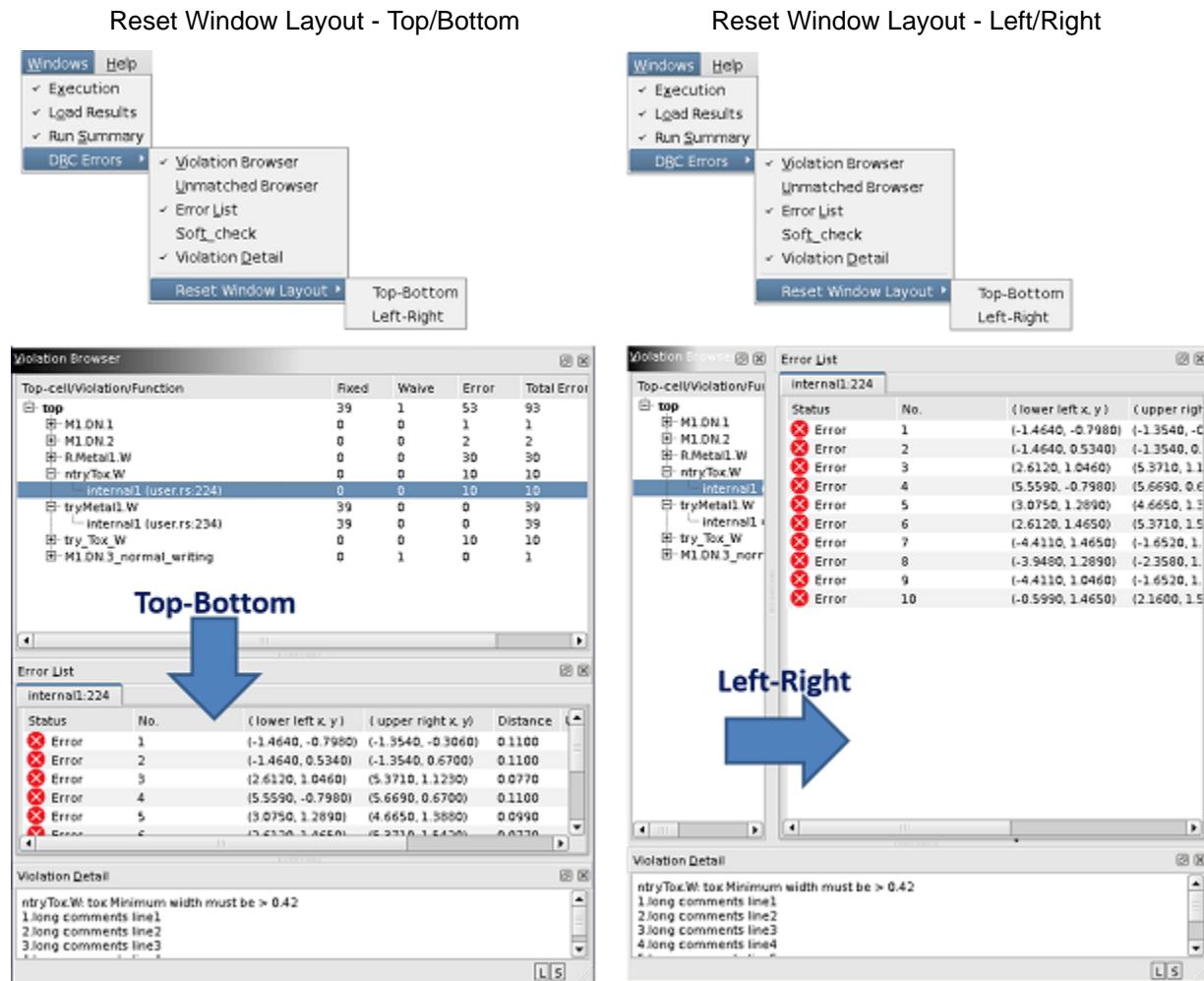


Figure 1-20 Windows Menu: Splitting Table

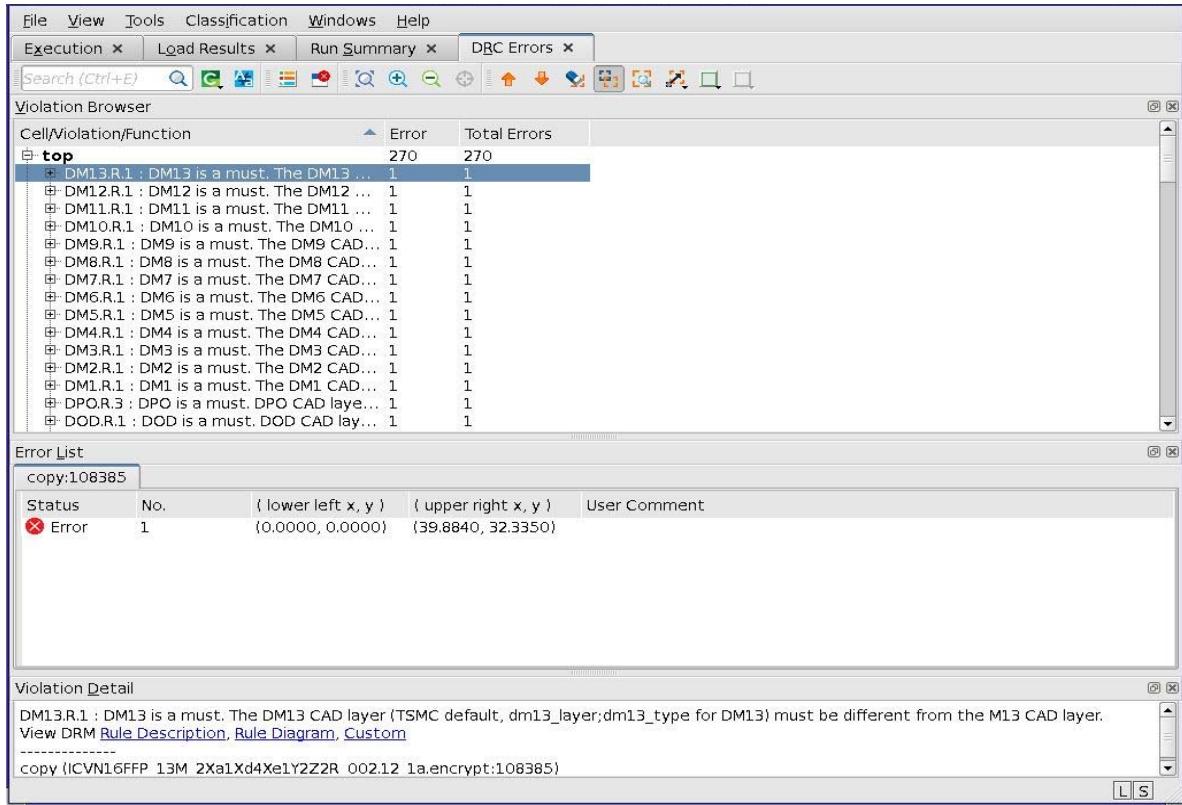


The Reset Window Layout feature in Windows menu > DRC Errors> Reset Window Layout, provides an easy way to relocate the Violation Browser, Error List, and Violation Detail panes. Figure 1-21 shows an example of the Top/Bottom and Left/Right window layouts.

**Figure 1-21 Windows Menu: Resetting the Window Layout**



**Figure 1-22** shows the Violation Detail pane with: text View DRM Rule Description, Rule Diagram, Custom. When you click these hyperlinks, a DRM document appears. You can search for the rule that identified the DRC violation.

**Figure 1-22 Violation Details**

For more information about the DRM link setting, see the “[VUE Preference System](#)” on [page 1-6](#).

---

## Help Menu

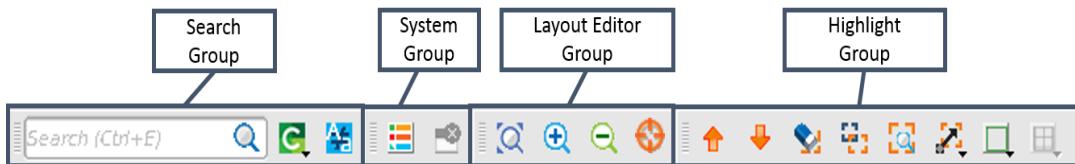
The *IC Validator VUE User Guide*, *IC Validator User Guide*, *IC Validator LVS User Guide*, and *IC Validator Reference Manual*, are now being distributed exclusively from the Documentation tab on SolvNet at the address <https://solvnet.synopsys.com/DocsOnWeb>. To enable the IC Validator VUE-based help system, download the `icvVUEug.pdf`, `icvug1.pdf`, `icvlvsug1.pdf`, and `icvrefman.pdf` files from SolvNet and place them into `$ICV_HOME_DIR/doc/icv/pdf/`.

---

## VUE DRC Toolbar

As shown in [Figure 1-23](#), there are four groups in the toolbar that you use to change the existing viewing capabilities. They are: Search, System, Layout Editor, and Highlight. Each group can be relocated within another group, shown in [Figure 1-24](#).

**Figure 1-23 VUE DRC Toolbar With Four Groups**

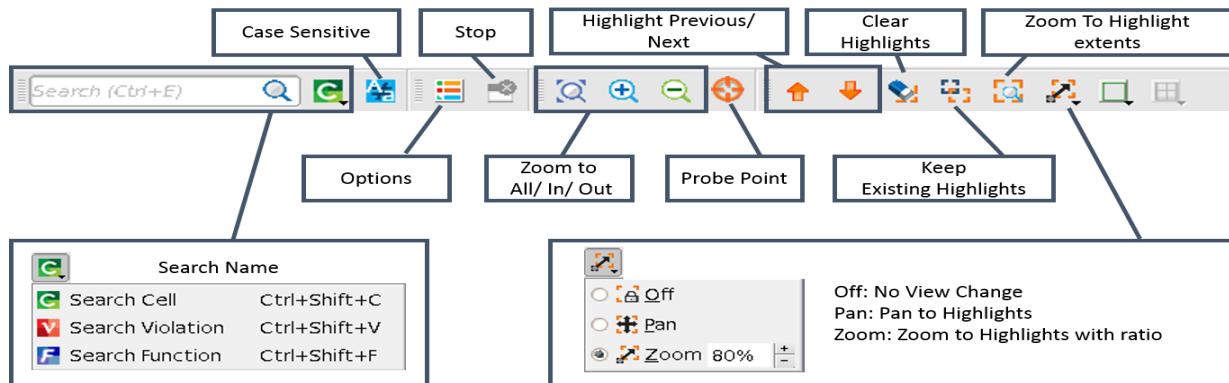


**Figure 1-24 Relocating Toolbar Groups Within Other Groups**



Figure 1-25, Figure 1-26, and Table 1-5 show the toolbar functions for each of the four groups.

**Figure 1-25 VUE DRC Toolbar**



Only those violations that contain the matched specified cell/violation/function names are displayed (others are hidden or temporarily filtered out). For example, you can switch to the Search function and type “text\_merge”, “\_rename”, and so on.

Figure 1-26 VUE DRC Highlight Color and Fill

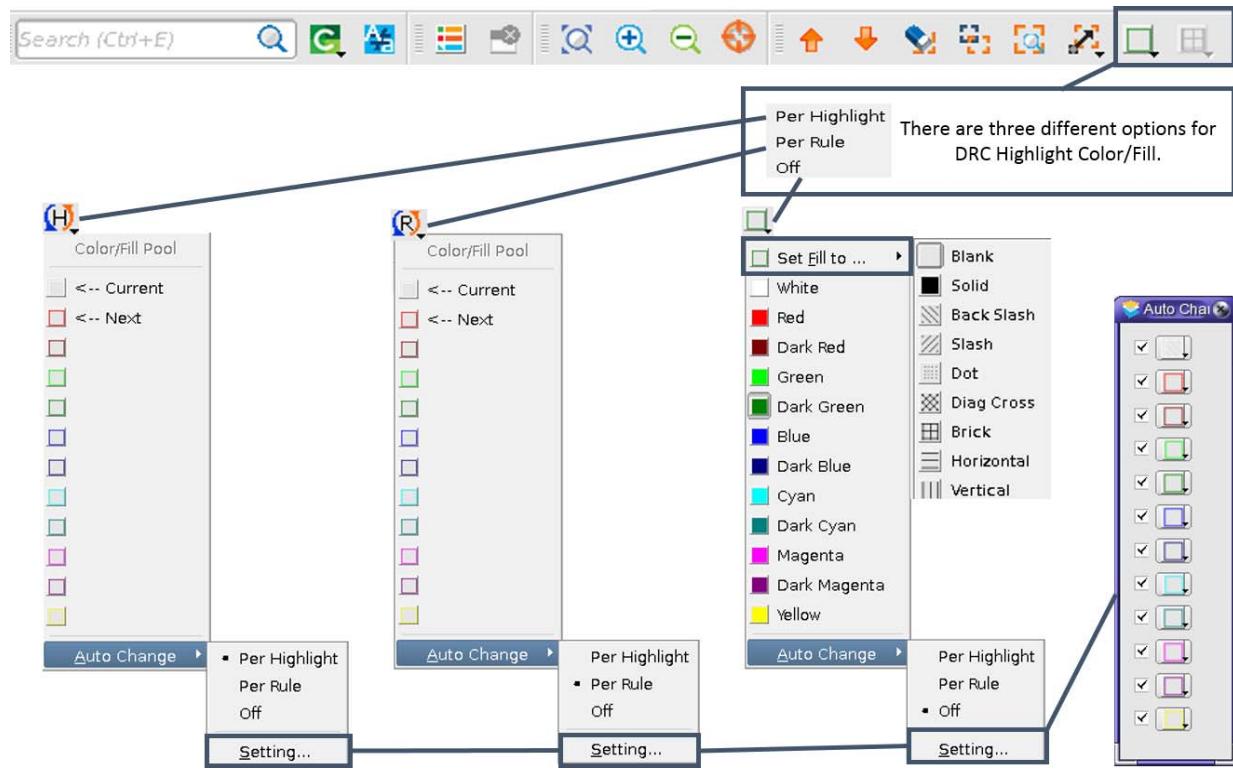


Table 1-5 defines the VUE DRC options.

Table 1-5 VUE DRC Toolbar Options

| Icon | Group  | Option                | Description  |
|------|--------|-----------------------|--|
|      | Search | Search                | Search name text box.  |
|      | Search | Search Cell Name      | Search name with cell name.                                      |
|      | Search | Search Violation Name | Search name with violation name.                                 |
|      | Search | Search Function Name  | Search name with function name.                                  |
|      | Search | Case-sensitive        | Words differ in meaning based on use of uppercase and lowercase. |
|      | System | Options Window        | Opens the VUE DRC Options window.                                |

**Table 1-5 VUE DRC Toolbar Options (Continued)**

| <b>Icon</b> | <b>Group</b>          | <b>Option</b>             | <b>Description</b>  |
|-------------|-----------------------|---------------------------|---|
|             | System                | Stop                      | Opens a task window listing all of the currently running processes.   |
|             | Layout Editor Control | Zoom to All               | Zooms to fit the entire open block within the layout window.  |
|             | Layout Editor Control | Zoom In                   | Zoom in by a factor of 0.5. Highlighted polygons are not considered   |
|             | Layout Editor Control | Zoom Out                  | Zoom out by a factor of 2. After you click this button, VUE waits for you to select a point in the layout editor. |
|             | Highlight             | Probe Point               | Used for LVS errors.  |
|             | Highlight             | Highlight Previous        | Highlights the previous error list item.  |
|             | Highlight             | Highlight Next            | Highlights the next error list item.  |
|             | Highlight             | Clear Highlights          | Removes all of the highlights that VUE DRC draws in the layout.   |
|             | Highlight             | Keep Existing Highlights  | On. The default.<br>Off. Removes all of the previous highlights before processing to the next one.                |
|             | Highlight             | Zoom to Highlight Extents | Zooms to fit the extents of the highlighted items.  |
|             | Highlight             | Off                       | Highlight zoom settings:<br>Off: No view change.  |
|             | Highlight             | Pan                       | Highlight zoom settings:<br>Pan: Pan to highlights.   |
|             | Highlight             | Zoom ratio                | Highlight zoom settings:<br>Zoom: Zoom to highlight with ratio.   |

**Table 1-5 VUE DRC Toolbar Options (Continued)**

| <b>Icon</b> | <b>Group</b> | <b>Option</b>                | <b>Description</b>   |
|-------------|--------------|------------------------------|--|
| N/A         | Highlight    | Check Highlight              | Change the color and fill pattern for highlighting primary DRC Errors if currently in the DRC/Extraction Errors tab. The error must be highlighted again to update the color and fill pattern.   |
| N/A         | Highlight    | Marker Highlight             | Change the color and fill pattern for highlighting secondary DRC Errors if currently in the DRC/Extraction Errors tab. The error must be highlighted again to update the color and fill pattern. |
|             | Highlight    | Auto Change -> Per Highlight | Highlight color/fill changes automatically per demanded highlight.   |
|             | Highlight    | Auto Change -> Per Rule      | Highlight color/fill changes automatically per different rule.   |
| N/A         | Highlight    | Auto Change -> Off           | Highlight color/fill is not changed.   |
| N/A         | Highlight    | Auto Change -> Setting       | Customize color/ fill loop with a user setting.  |

# 2

## VUE Communications Interface Guide

*This chapter provides basic information about how to interface with layout and schematic tools.*

VUE supports interfaces to several layout and schematic capture tools. These interfaces allow VUE to display error information, highlight nets, shorts, and devices, and to cross-probe.

The supported layout interfaces are:

- Graphical User Interface
- IC WorkBench EV Plus
- IC Compiler
- IC Compiler II
- Custom Compiler
- Laker Custom Layout System
- Cadence Virtuoso Layout Editor

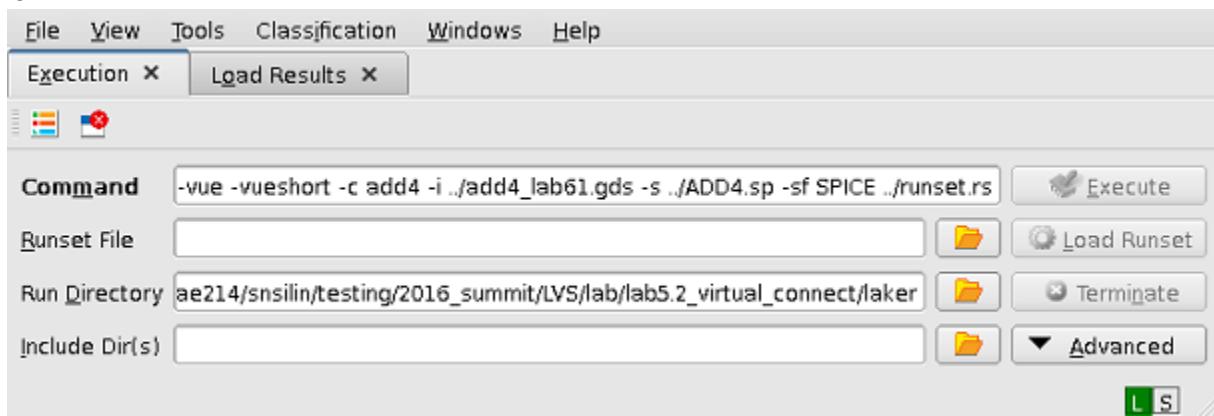
## Graphical User Interface

Information about the communications interface is shown in two sections of the GUI.

### Main VUE GUI

The lower-right corner of the main VUE GUI window displays the communication status of the layout and schematic connections, shown in [Figure 2-1](#). The names of the connected layout and schematic tools are displayed with a tool tip.

*Figure 2-1 Main VUE Window*



Note:

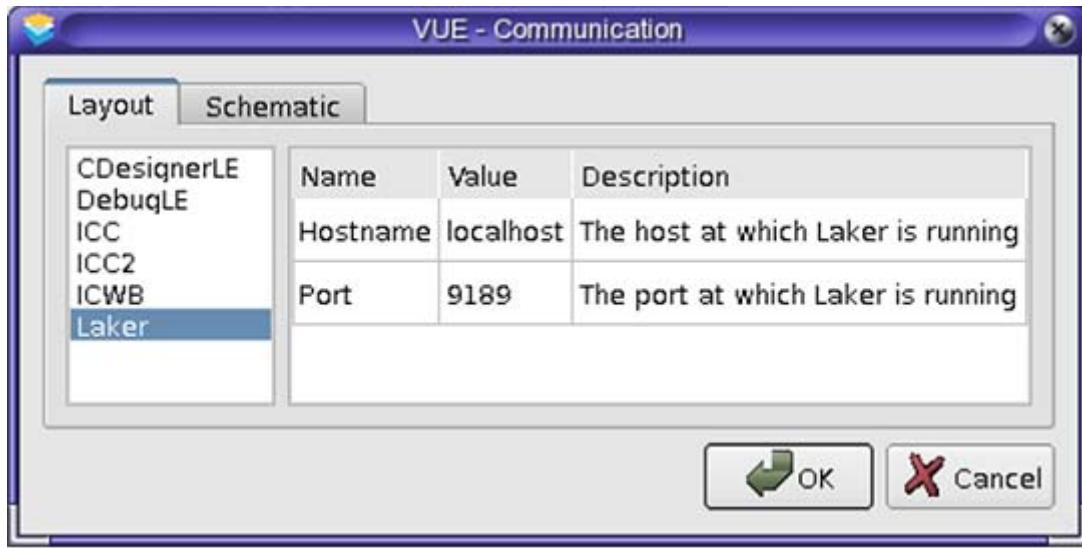
The `icv_vue -64` command-line option runs VUE with 64-bit coordinates support.

---

## Communication GUI

The Communication GUI is started from Tools > Communication. The VUE Communication dialog box allows you to connect to different layout tools and to change interface-specific parameters. The supported layout tools are shown in [Figure 2-2](#).

*Figure 2-2 Schematic and Layout Tools Lists*



---

## Tools Menu

From the Tools menu, you see

- Communication window. Allows you to change viewing options or set the layout connection.
  - Communication status. The communication status of the layout and schematic connection is shown in the lower-right corner of the main VUE GUI window, shown in [Figure 2-1](#). The names of the connected layout and schematic tools are displayed with a tool tip.
  - [Table 2-1](#) shows the layout and schematic communication status.

See [Chapter 2, “VUE Communications Interface Guide”](#) for more information about the Communication window.

*Table 2-1 Communication Status*

| Status | Description  |
|--------|--|
|        | Layout is not connected; schematic is not connected. |
|        | Layout is connected; schematic is not connected.     |
|        | Layout is not connected; schematic is connected.     |
|        | Layout is connected; schematic is connected.         |

## IC WorkBench EV Plus

VUE communicates with IC WorkBench EV Plus through a socket or port connection. This socket must be opened by IC WorkBench EV Plus. You can open the communication socket manually upon IC WorkBench EV Plus startup, or within the IC WorkBench EV Plus session. The socket can be any number between 1000-65535 that is not already being used. The `netstat` UNIX command shows which port numbers are available.

### IC WorkBench EV Plus Command Line

To open a socket on the command line, start IC WorkBench EV Plus using the following command-line argument:

```
icwbev -socket number
```

### Within IC WorkBench EV Plus

To open a socket after IC WorkBench EV Plus has already been started, type the following within the IC WorkBench EV Plus command window:

```
user_socket open 0
```

This command returns a number. This is the open socket that IC WorkBench EV Plus established.

IC WorkBench EV Plus connects to VUE automatically when you set the `ICWBEV_USER` environment variable:

- For newer versions of IC WorkBench EV Plus:

```
setenv ICWBEV_USER ICV
```

A telescope button is added to the toolbar. Click this button to start VUE and establish a connection.

---

## VUE Command-Line Options

You can start IC WorkBench EV Plus and VUE from the UNIX command prompt and connect directly to the existing IC WorkBench EV Plus session using the following command:

```
icv_vue -lay icwb -layArgs Port number Host hostname
```

**Note:**

The host name variable specifies the machine name to use when you want to display IC WorkBench EV Plus and VUE. You do not need to change this variable if you are running and displaying the GUI on the same machine. If you are displaying the GUI on a different machine, you must set this variable.

---

## Communication Window

Start VUE from a UNIX command line and connect to a version of IC WorkBench EV Plus that is already running by following these steps:

1. Choose Tools, and then choose Communication to display the Communication dialog box.
2. Select the IC WorkBench entry.

Enter the values for the variables directly into the Variables dialog box in the Communication window.

1. Choose Port, then enter the socket value and click OK.
2. Select Hostname, then enter the host name (or IP) and click OK.

**Note:**

The host name variable specifies the machine name on which you want to display IC WorkBench EV Plus and VUE. You do not need to change this variable if you are run and display on the same machine. If you display the GUI on a different machine, you must set this variable.

---

## Communicating with IC WorkBench EV Plus Sessions

To communicate with another IC WorkBench EV Plus session, open the Communication dialog box and choose IC WorkBench EV Plus from the list on the left. Then, set the host name and port variables, and click OK.

---

## IC Compiler

To use VUE within IC Compiler, set the following environment variables to run the tool:

```
setenv ICV_HOME_DIR path_to_icv_build
setenv SYNOPSYS_SYSTYPE LINUX.64
```

Source the IccMenu.tcl file, which is located in the install directory:

```
$ICV_HOME_DIR/etc/tcl-u/IccMenu.tcl
```

To load the IccMenu.tcl file automatically, execute the following command in the IC Compiler console:

```
source $env(ICV_HOME_DIR)/etc/tcl-u/IccMenu.tcl
```

A menu is added to IC Compiler that starts VUE and establishes a connection.

---

## IC Compiler II

To use VUE within IC Compiler II, set the following environment variables to run the tool:

```
setenv ICV_HOME_DIR path_to_icv_build
setenv SYNOPSYS_SYSTYPE LINUX.64
```

The Icc2Menu.tcl file is located in the install directory. To source this file, execute the following command in the IC Compiler II console:

```
source $env(ICV_HOME_DIR)/etc/tcl-u/Icc2Menu.tcl
```

A menu is added to the IC Compiler tool that starts VUE and establishes a connection.

---

## Custom Compiler

To use VUE with the Custom Compiler™ tool, choose DRC, LVS, or Extract from the Verification menu. Each of those selections launches another menu with the “setup and run”, “run”, “debug”, and “View Output” selections.

For more information about using VUE with Custom Compiler, see the *See the Custom Compiler™ Design Checking and Physical Verification User Guide*, available on SolvNet.

---

## Laker Custom Layout System

To use VUE with the Laker Custom Layout System,

1. Set the following environment variables to run the tool:

```
setenv ICV_HOME_DIR path_to_icv_build  
setenv SYNOPSYS_SYSTYPE LINUX.64
```

2. Invoke the Laker system with the startup Tcl file.

```
% laker -play $ICV_HOME_DIR/etc/tcl-u/LakerVue.tcl
```

3. Connect VUE and the Laker system using one of these two methods:

- Select the Laker system from the tool list in the VUE Communication dialog box and connect it to port 9189.
- Open a design window within the Laker system and click IC Validator > Start IC Validator VUE.

---

## Cadence Virtuoso Layout Editor

The VUE Cadence® Virtuoso® Layout Editor interface was developed in SKILL v6.10 and tested with Cadence ICFB v5.0. (The Cadence Virtuoso Layout Editor was formerly known as Composer.)

To use VUE with the Cadence tools, set the following environment variables to run the tool:

```
setenv ICV_HOME_DIR path_to_icv_build  
setenv SYNOPSYS_SYSTYPE LINUX.64
```

Load the SkillVueMenu.il file here.

```
$ICV_HOME_DIR/etc/VUE/SkillVueMenu.il
```

You can load the SkillVueMenu.il file automatically by executing the following command in the Command Interface Window.

```
load( strcat( getShellEnvVar("ICV_HOME_DIR") \\\n" /etc/VUE/SkillVueMenu.il" ))
```

Add this command to the cdsinit file so that the SkillVueMenu.il file is loaded automatically every time you start the Cadence environment.

Loading this file adds the IC Validator menu to all new Virtuoso Layout and Schematic Editor windows.

---

## Running IC Validator VUE From Virtuoso

From the IC Validator menu in the Virtuoso editor, select Run VUE. This opens the VUE menu. The L (maskLayoutL), XL (maskLayoutXL), and GXL (maskLayoutGXL) environments in the IC Validator menu are supported in the SkillVueMenuHF.il and SkillVueMenu.il files in Virtuoso.

Note:

To debug a cell using DRC/Extraction and LVS simultaneously, set the Edit-In-Place mode to On for DRC/Extraction and Off for LVS. By default, DRC/Extraction does not use Edit-In-Place, while LVS does.

---

## VUE Connections in Virtuoso

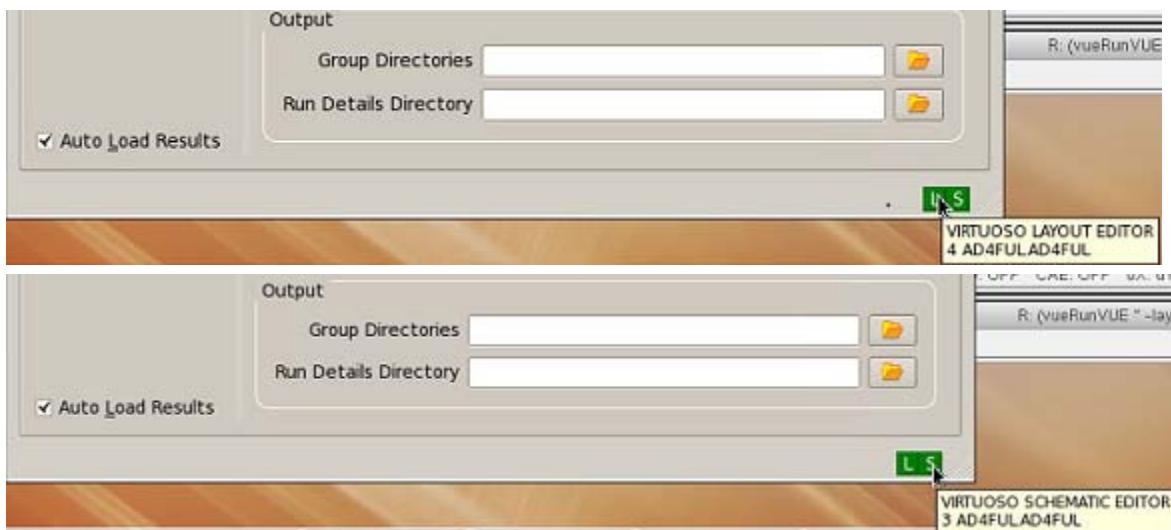
If the Layout window of a cell is open and VUE is launched from it, VUE connects only to the Layout window. Similarly, if a Schematic window of a cell is open and VUE is launched from it, VUE connects only to the Schematic window.

If both the VUE and Layout and Schematic views of the same cell are open and VUE is launched from one of the two windows, VUE connects to both windows. The order of the search to make the connection is as follows:

- Search for an open window of another view of the same cell in the same library.
- If not found, search for an open window of another view of the same cell in a different library.

In [Figure 2-3](#), the status of the connection is displayed by tool tip, which contains the editor type, window ID, library name, and cell name.

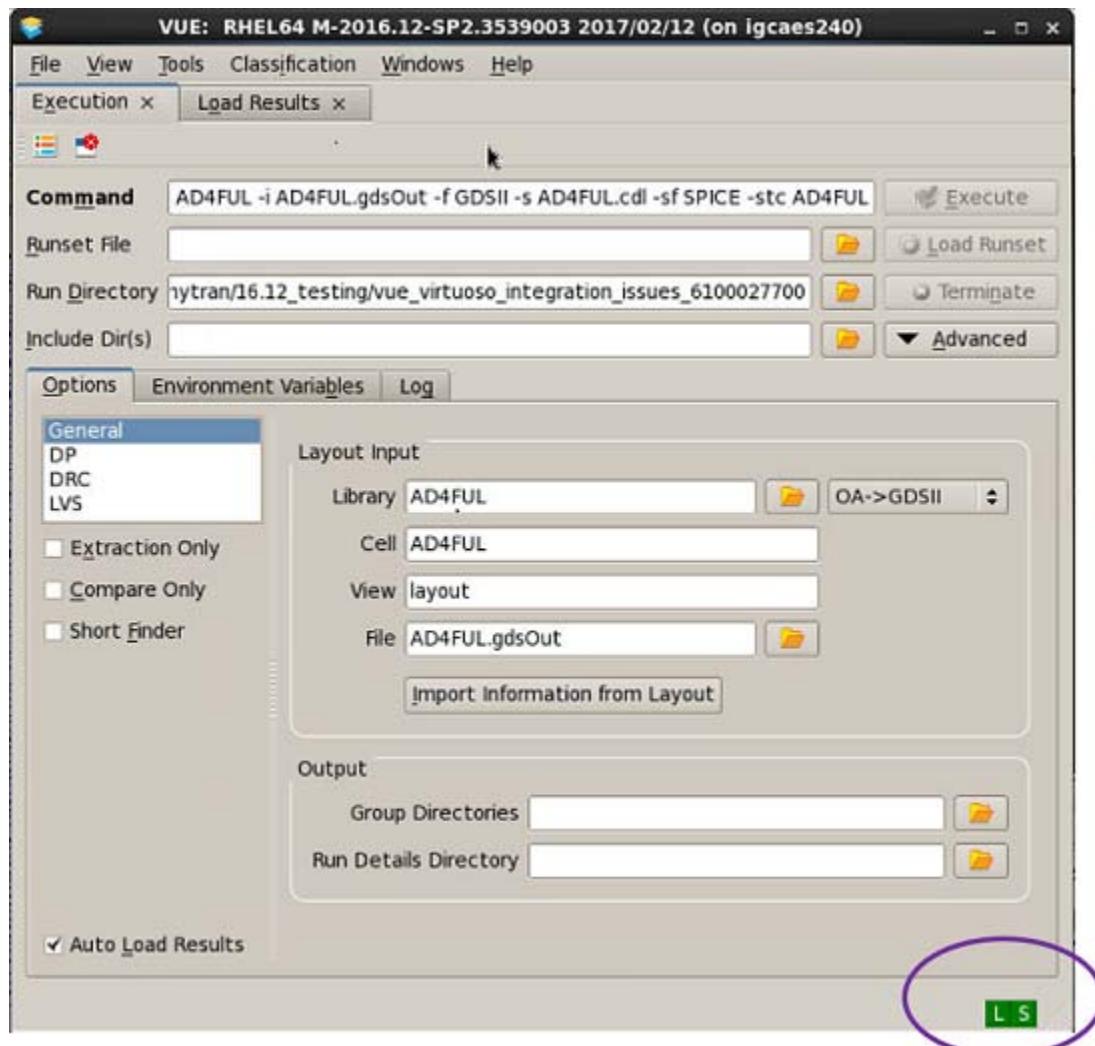
Figure 2-3 Status of the Connected Window



## Autopopulating VUE Input

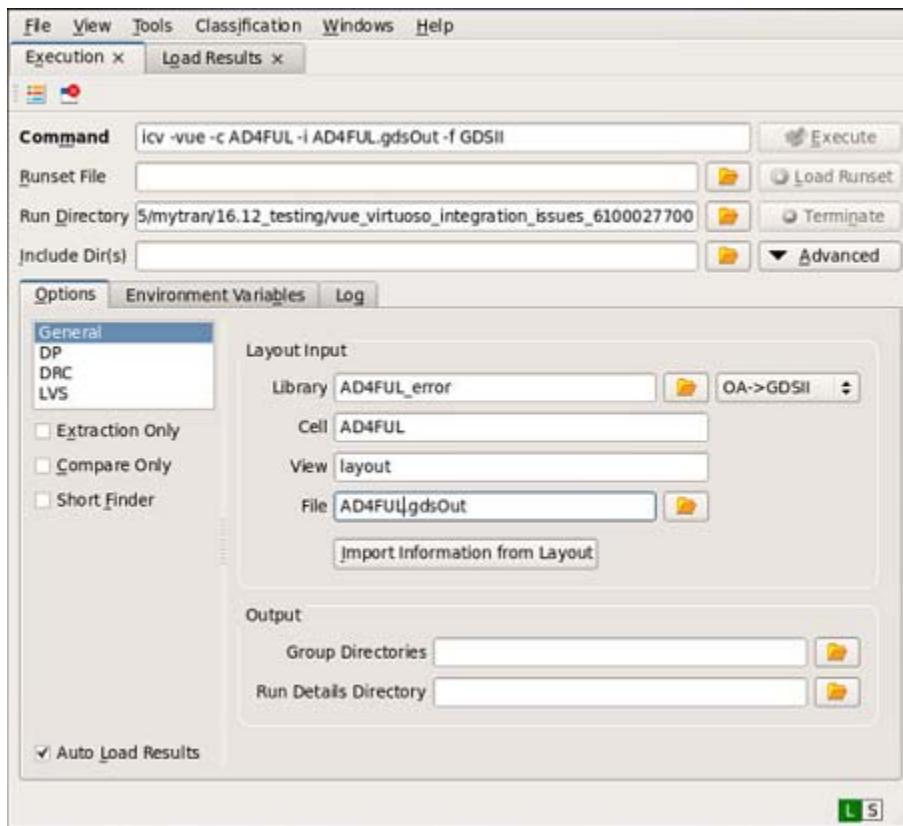
Autopopulating the input boxes occurs when the information is provided within the IC Validator runset.rs file library function or when the Layout or Schematic is connected. In [Figure 2-4](#), both the Layout and Schematic are connected.

*Figure 2-4 Layout and Schematic Connected*



If the information in the Layout Input fields is not correct in the connected Layout window, you can load the input information from the open and connected windows by clicking the Import Information from Layout button, shown in [Figure 2-5](#).

*Figure 2-5 Import Information From Layout*



To make connections to other open windows versus the window where VUE was invoked, choose Tools > Communication, as shown in [Figure 2-6](#).

Choose from a list of available windows for connection or disconnection and click OK. Click the Refresh button to update the list of open windows for selection, as shown in [Figure 2-7](#) and [Figure 2-8](#).

*Figure 2-6 Communication*

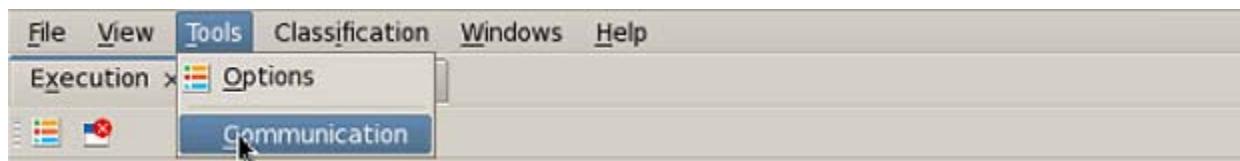


Figure 2-7 VUE Communication - Layout Tool and Window Configuration

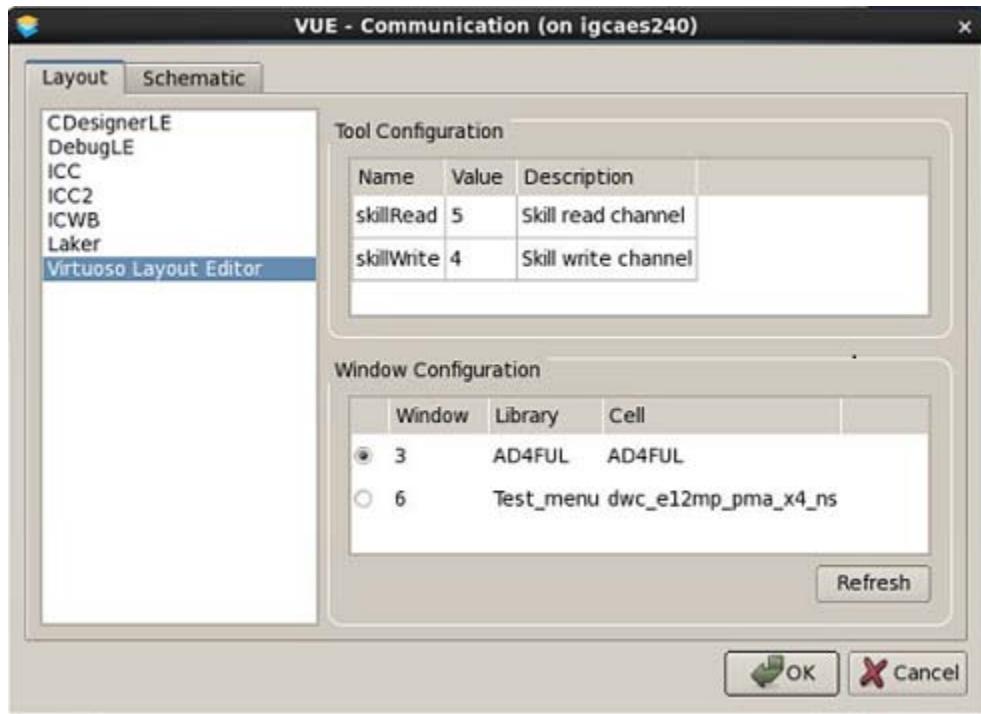
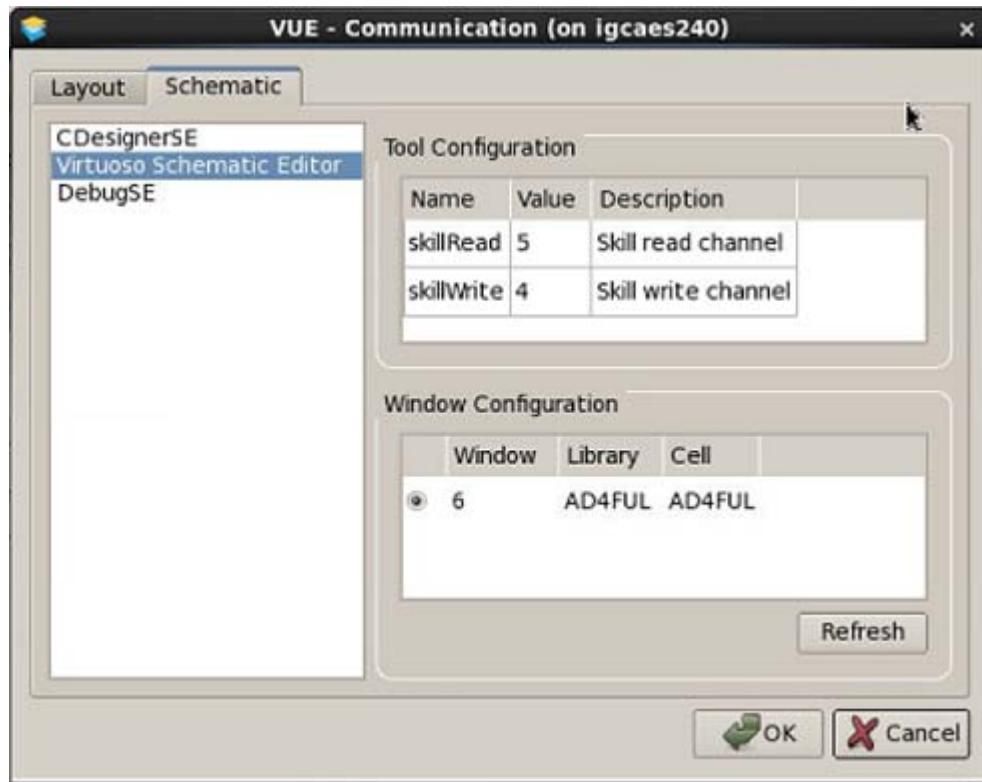
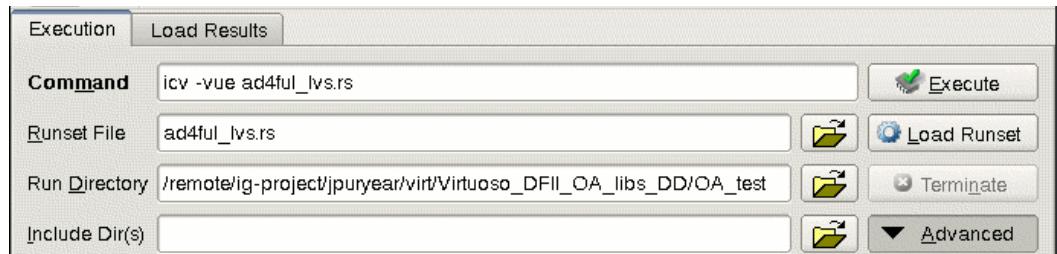


Figure 2-8 VUE Communication - Schematic Tool and Window Configuration



You can fill in or load runset information from the browse button. After the information is populated, click Load Runset, as shown in [Figure 2-9](#).

Figure 2-9 Runset File



## Exporting DFII or OpenAccess Capabilities and Usage Within VUE

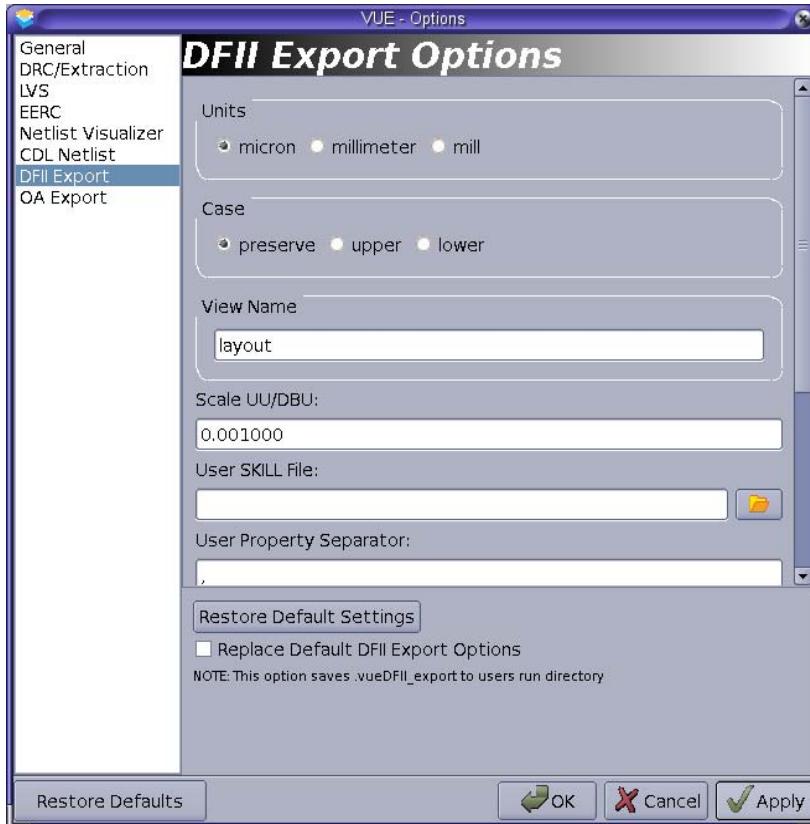
To access the DFII export options or OpenAccess export options, choose Tools > Options, shown in [Figure 2-10](#).

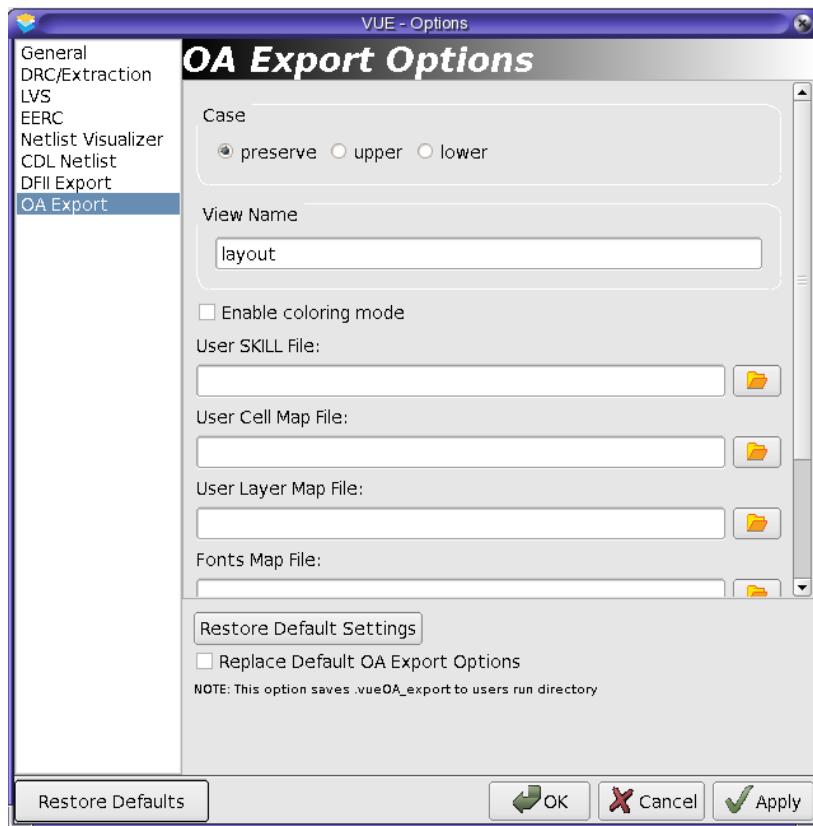
*Figure 2-10 DFII Export Options*



The DFII Export Options and OA Export Options pages in the VUE Options dialog box cover the majority of selectable options that are provided within the Cadence Virtuoso Export stream, shown in [Figure 2-11](#) and [Figure 2-12](#).

*Figure 2-11 DFII Export Options*



*Figure 2-12 OA Export Options*

The OA Export Options page allows you to specify the SKILL file, cell map file, layer map file, fonts map file, property map file, and coloring mode.

**Note:**

The Enable coloring mode check box appears only when you connect with Virtuoso Advanced Node.

As shown in [Figure 2-11](#), the Replace Default DFII Export Options check box allows you to override the default settings packaged within the IC Validator tool. Check this box and click OK or Apply to generate the .vueDFII\_export file in the working directory.

As shown in [Figure 2-12](#), the Replace Default OA Export Options check box allow you to override the default settings packaged within the IC Validator tool. Select this check box and click OK or Apply to generate the .vueOA\_export file in the working directory.

To ensure that your defaults are loaded for every run, make sure the export file resides within the directory from which VUE was launched.

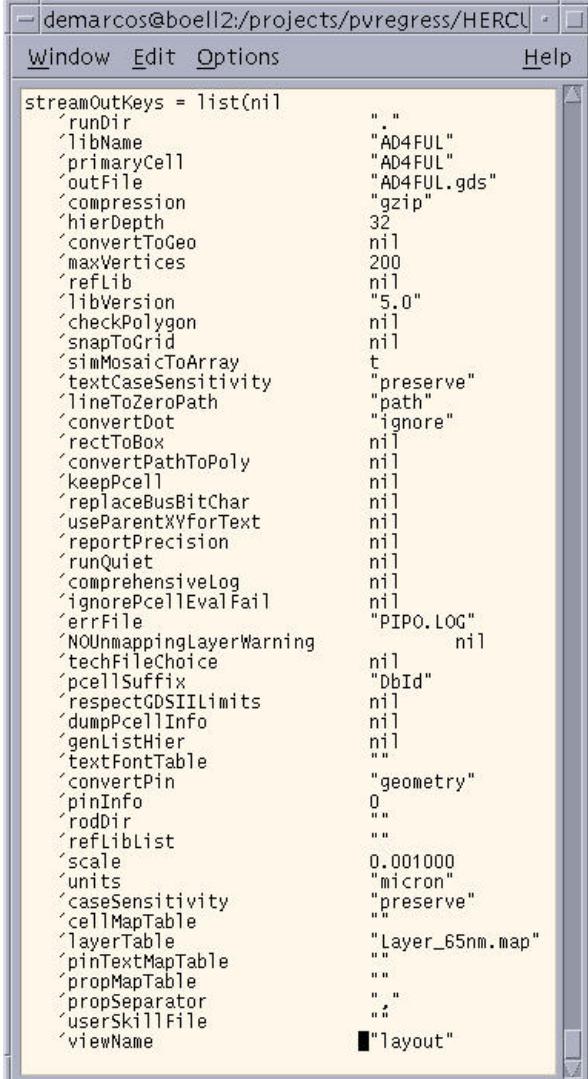
**Note:**

If you saved your defaults to an export file and you want to restore the IC Validator default settings,

- Click the Restore Default Settings button. Then, select the Replace Default DFII Export Options or Replace Default OA Export Options check box and click OK or Apply.
- Manually remove the export file from the working directory.

[Figure 2-13](#) shows an example of the .vueDFII\_export file. This file can be changed manually and is automatically loaded with VUE.

*Figure 2-13 .vueDFII\_export File*



```

demarcos@boell2:/projects/pvregress/HERCU>
Window Edit Options Help
streamOutKeys = list(nil
  'runDir           "."
  'libName          "AD4FUL"
  'primaryCell      "AD4FUL"
  'outFile          "AD4FUL.gds"
  'compression      "gzip"
  'hierDepth        32
  'convertToGeo     nil
  'maxVertices      200
  'refLib           nil
  'libVersion       "5.0"
  'checkPolygon     nil
  'snapToGrid        nil
  'simMosaicToArray t
  'textCaseSensitivity "preserve"
  'lineToZeroPath   "path"
  'convertDot       "ignore"
  'rectToBox         nil
  'convertPathToPoly nil
  'keepPcell         nil
  'replaceBusBitChar nil
  'useParentXYforText nil
  'reportPrecision   nil
  'runQuiet          nil
  'comprehensiveLog nil
  'ignorePcellEvalFail nil
  'errFile           "PIPO.LOG"
  'NOUnmappingLayerWarning nil
  'techfileChoice    nil
  'pCellSuffix       "DbId"
  'respectGDSIILimits nil
  'dumpPcellInfo     nil
  'genListHier       nil
  'textFontTable     ""
  'convertPin        "geometry"
  'pinInfo           0
  'rodDir            ""
  'reflibList         nil
  'scale              0.001000
  'units              "micron"
  'caseSensitivity    "preserve"
  'cellMapTable      ""
  'layerTable         "Layer_65nm.map"
  'pinTextMapTable   ""
  'propMapTable       ""
  'propSeparator      ""
  'userSkillFile     ""
  'viewName          "layout"
)

```

**Note:**

The .vueDFII\_export is prepackaged with IC Validator to provide factory defaults. This file is located in the following directory:

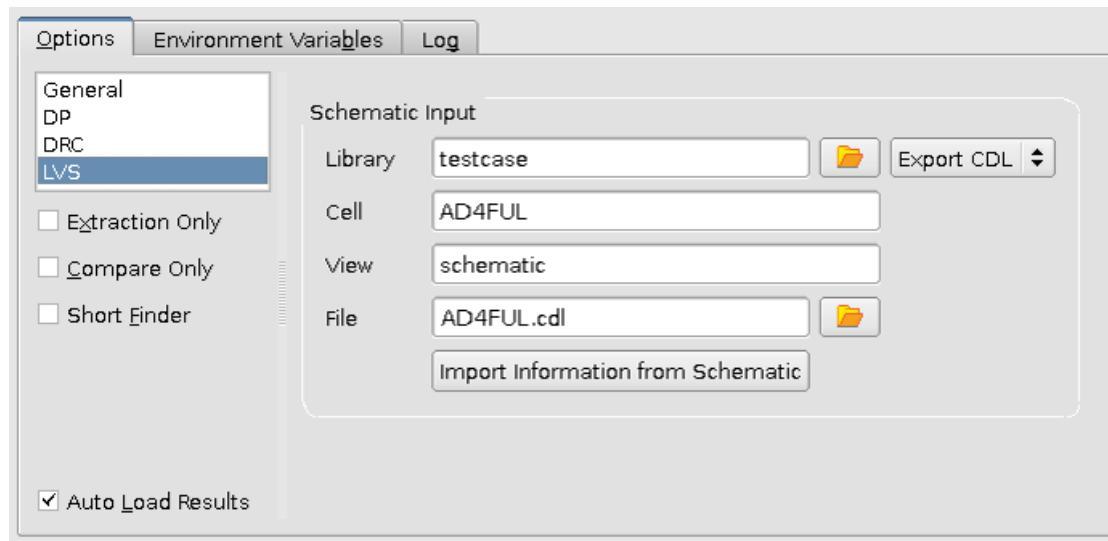
ICV\_HOME\_DIR/etc/

## Exporting CDL Capabilities and Usage Within VUE

You can import schematic library information from the LVS Options just as you can import layout information for an IC Validator run using the Import Information from Schematic.

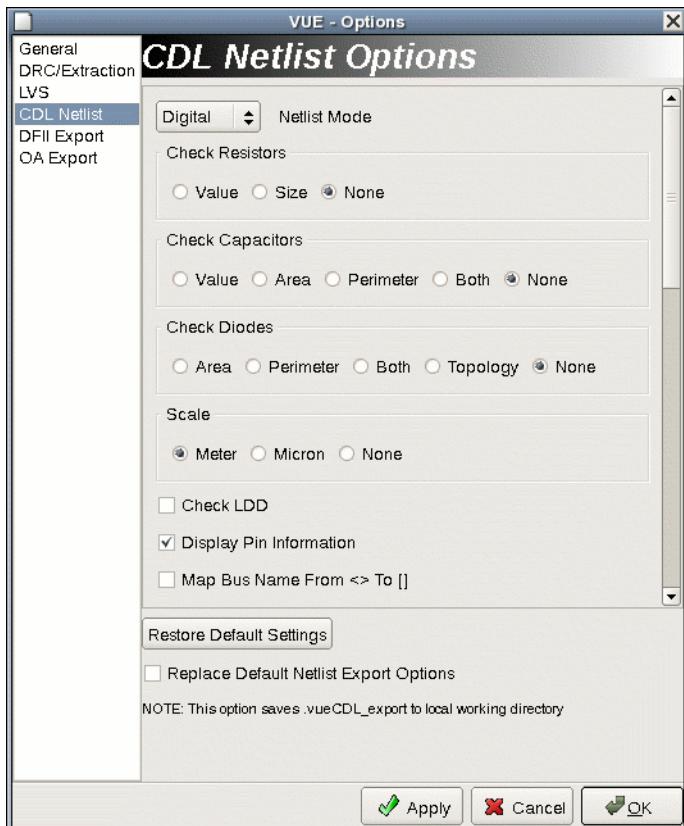
[Figure 2-14](#) shows the Schematic Library Information dialog box.

*Figure 2-14 Schematic Library Information*



The schematic library is automatically exported to a CDL netlist upon execution. The CDL process options can be controlled from the CDL Netlist Options dialog box, shown in [Figure 2-15](#).

*Figure 2-15 CDL Netlist Options*



The Replace Default Netlist Export Options check box allows you to override the default settings packaged within IC Validator. Check this box and click OK or Apply to see a new .vueCDL\_export file in the working directory.

To ensure that your defaults are loaded for every run, make sure the .vueCDL\_export file resides within the directory from which VUE was launched.

**Note:**

If you have previously saved defaults to the .vueCDL\_export file and you want to restore the factory default settings, you must first click the Restore Default Settings button, and then either check the Replace Default Netlist Export Options box and click OK or Apply, or remove the .vueCDL\_export file from the working directory.

An example of the .vueDFII\_export file is shown in [Figure 2-13](#). This file can be changed manually and is automatically loaded with VUE.

## Postprocessing CDL Netlists Within the IC Validator Run Stream

Flows that require CDL netlists to be postprocessed benefit from the IC Validator Netlist Hook script. One of the reasons CDL netlists might need to be postprocessed is that inherent CDL format limitations do not allow 3-terminal resistors. VUE allows you to use these scripts within the IC Validator netlisting flow. The benefit of this flow allows you to streamline your physical verification process so that external processes can be handled in a single step to maintain flow integrity.

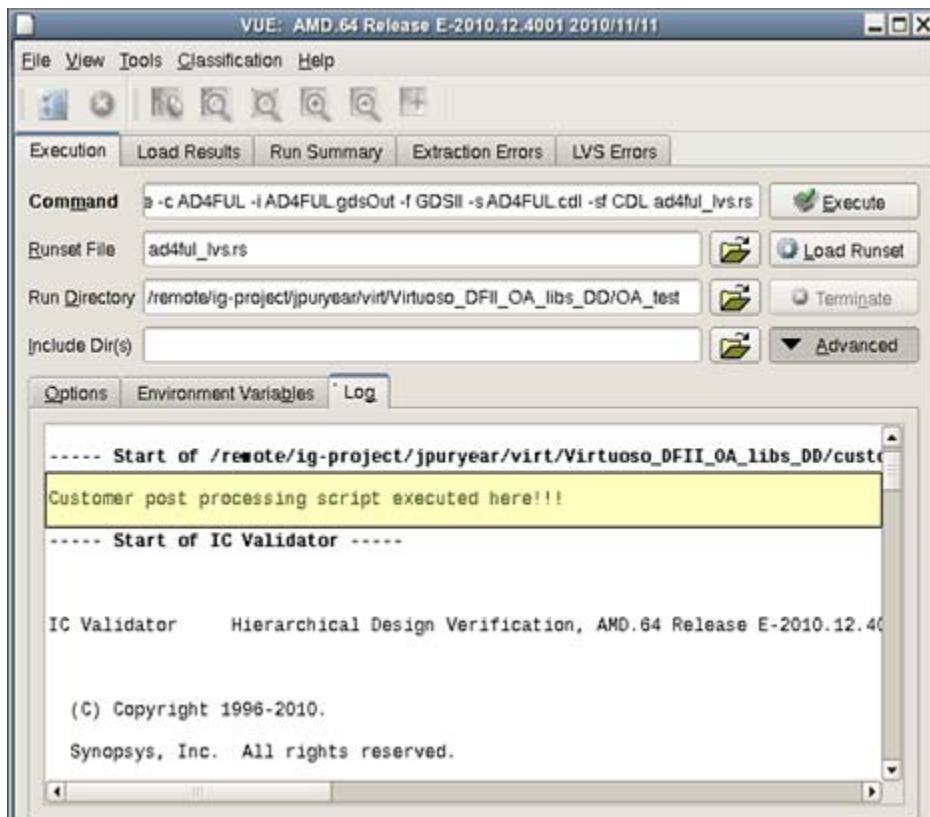
To use the IC Validator Netlist Hook script, execute the following command:

```
% setenv CDL_HOOK_SCRIPT_NAME \
customer_cdl_post_processing_script.pl
```

The script can be any name you choose.

Click Execute to observe where the user scripts are invoked within the flow. See the highlighted section in [Figure 2-16](#).

*Figure 2-16 Hook Script Details*



Ensure that the required environment variable is set before launching ICFB Virtuoso and VUE. The output postprocessed CDL file name generated from the user scripts must be the same name as the input for correct NetTran program execution.

This flow does not influence the NetTran program execution nor its settings.

**Note:**

The end user is responsible for any scripts used within this flow and any effects these scripts might cause.

---

## Limitations

VUE must be started from the menu entry in Virtuoso Layout Editor or Virtuoso Schematic Editor. VUE cannot connect to these tools if it is started directly from a UNIX command prompt.

Traversing between Virtuoso and Virtuoso XL layout tools might deregister the IC Validator SKILL filter files, causing the IC Validator menu to disappear. Furthermore, some third party SKILL files might inadvertently deregister IC Validator menus upon loading.

Reload the IC Validator SKILL files in the former case, and ensure that the SKILL files loading sequence in the latter case does not invoke a conflict.

**Note:**

When a filter is deregistered, you might need to reload it. Another vendor's SKILL files or switching between Virtuoso and XL might cause filters to be deregistered.

# 3

## Running IC Validator Within VUE

---

*This chapter explains how to run VUE and view and analyze errors.*

VUE is described in the following sections:

- [Executing IC Validator From VUE](#)
- [Loading IC Validator Results Into VUE](#)
- [DRC Errors and Device Extraction Violations](#)
- [The soft\\_check Graphical Violation Report](#)
- [LVS Errors](#)
- [Netlist Visualizer](#)
- [Layer Debugger](#)

---

## Executing IC Validator From VUE

The Execution tab allows you to set up and execute IC Validator. The runset file name, including its path if the file is not in the current working directory, must be entered to start IC Validator successfully.

On startup, there are four text boxes in VUE.

- **Command.** Specifies the IC Validator command-line option which is executed and defaults to `icv -vue`. It can also be used to execute a script instead of calling IC Validator directly.

Note:

If you are in 64-bit mode VUE, the command line changes to automatically adjust to run the 64-bit version of IC Validator.

- **Runset File.** Defines the runset that is executed. The runset can be loaded in VUE by clicking the Load Runset button. This populates the configuration boxes with values set in the runset.
- **Run Directory.** Specifies the directory where IC Validator is run and defaults to the current working directory.
- **Include Dir(s).** If the runset requires the use of an Include directory for additional runset pieces, the location can be specified here.

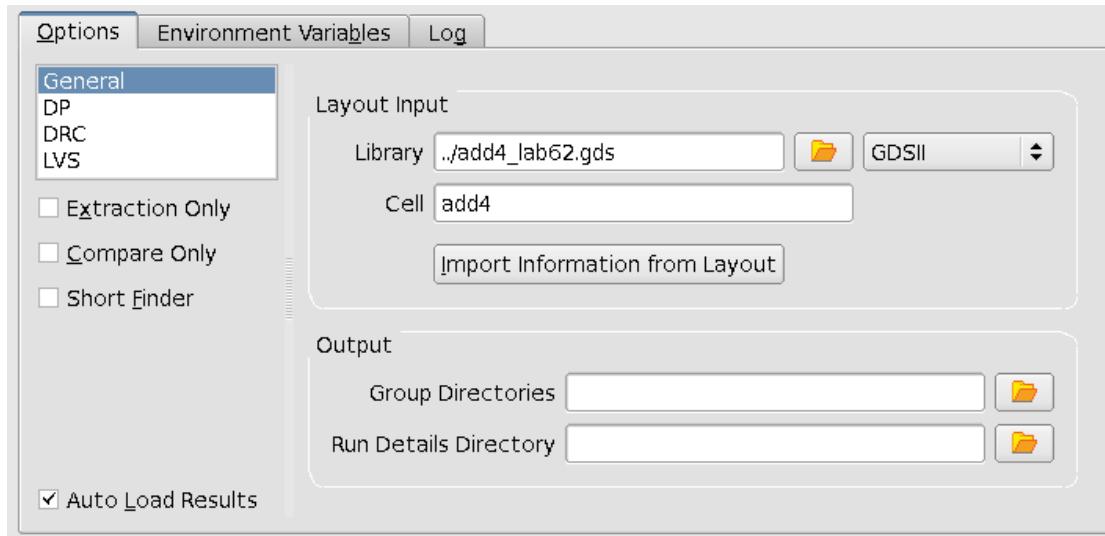
The Advanced button displays additional controls for the IC Validator run. For example, the Environment Variables tab appears after you click the Advanced button. Select this tab to display all of the environment variables used in the runset. The environment variables are displayed after you load an error-free runset. You can change the values of the environment variables before executing the run.

The Execute button starts the IC Validator run as it is currently configured within VUE. A log window opens in VUE to help track the run.

You can control four categories of options using the Options window, shown in [Figure 3-1](#).

The general options control the input and output of an IC Validator run, such as block, library name, and library format (GDSII, Milkyway, OASIS, or OpenAccess).

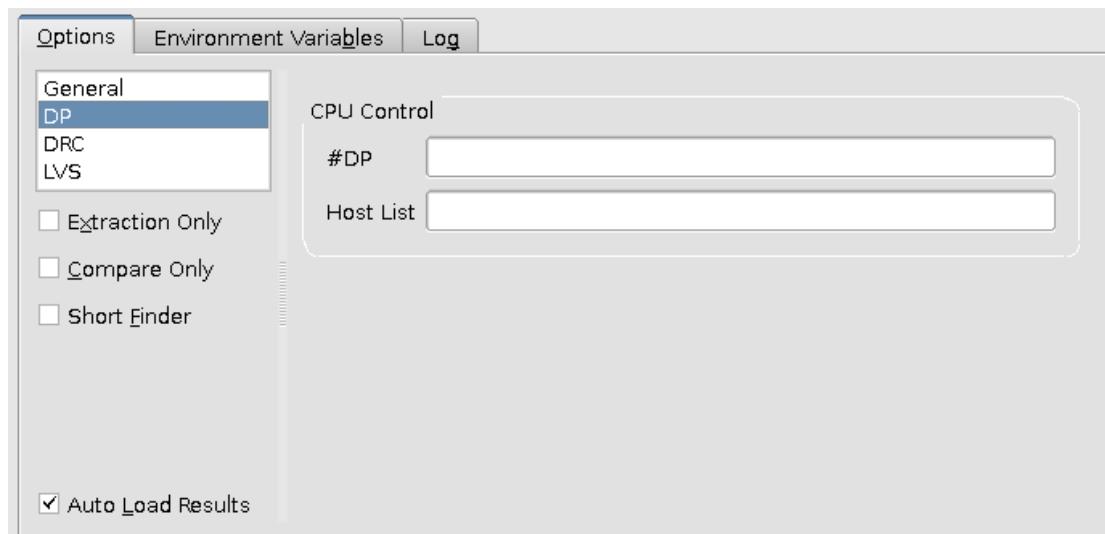
Figure 3-1 Options Tab With General Options



When you click the Import Information from Layout button, VUE loads the Cell, Library, and format information into the input pane from the connected layout editor. In [Figure 3-1](#), the Block information is add4; the Library information, add4\_lab62.gds; and the format information, GDSII.

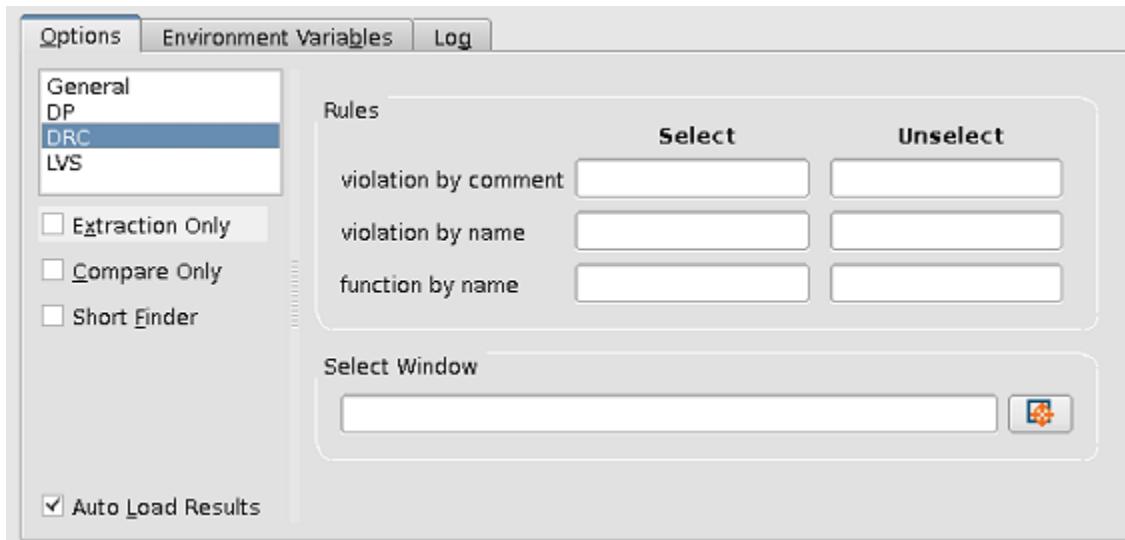
DP Options control distributed processing and allow you to select the number of distributed processes and threads, and specify the host machines for the run. [Figure 3-2](#) shows the log window when DP is selected.

Figure 3-2 Options Tab With DP Options



DRC options allow you to select or unselect rules and specify a select window for your run. If specified, IC Validator runs on only the defined extents. [Figure 3-3](#) shows the log window when DRC is selected.

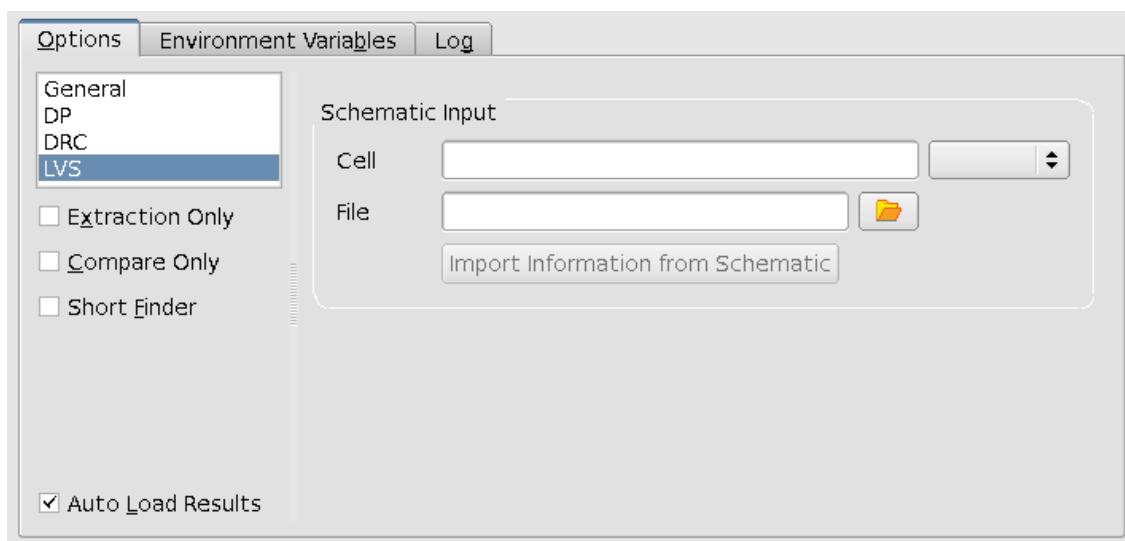
*Figure 3-3 Options Tab With DRC Options*



See Chapter 1, IC Validator Basics, of the *IC Validator User Guide* for more information about the select and unselect command-line options.

LVS options allow you to specify a schematic netlist or schematic library for use in the IC Validator run. [Figure 3-4](#) shows the log window when LVS is selected.

*Figure 3-4 Options Tab With LVS Options*



---

## Loading IC Validator Results Into VUE

VUE loads the results if there are no IC Validator execution errors with an exit status of zero. If an intermediary script or executable is used to launch IC Validator, it must return with a status that accurately reflects the IC Validator return code. Occasionally, system network problems cause error messages to appear in the Messages window when executing a runset from the IC Validator interface. If this occurs, execute the runset from the UNIX command line.

After IC Validator successfully completes the run, VUE, by default, loads the error output. This is controlled by an Auto Load Results check box in the Execution Configuration tab.

DRC results can be loaded during the run. When loaded, VUE displays the DRC errors present in the error database. To load violations found after the initial loading, you must manually reload the results. All error classifications done before reloading the database are retained.

---

## Run Settings File

The run settings file (*cell.vue*) is an ASCII file that contains all the information that VUE needs to display the results correctly. It is automatically generated by IC Validator in the working directory, if the `create_vue_output` argument is `true` in either the `error_options` function or the `-vue` command-line option. Within the Load Results tab, the Run Settings File box is automatically populated if the *cell.vue* file exists in the same directory from which VUE was started. Otherwise, you can manually browse for *cell.vue*. [Figure 3-5](#) shows an example of the Load Results tab where the Run Settings File is automatically populated.

*Figure 3-5 Load Results Tab*



VUE is designed to load information correctly even when it is moved from the original working directory. All of the information in the *cell.vue* file is relative to the run directory. As long as all of the data is kept in the same location relative to the run directory when it is moved, the data is loaded automatically. If data is moved relative to the run directory, you can edit the appropriate line in the *cell.vue* file to retrieve the information.

If necessary, you can browse for the *cell.vue* file, and then load. The browser displays only directories and files with the .vue extension.

A load status bar in the lower right corner is displayed when loading data into VUE.

The version of IC Validator that generates the run settings file, and the version of VUE used to view the data must be from the same major release. For example, version H-2013.06 is compatible with version H-2013.06-SP1, but not with version J-2014.06.

---

## Backward Compatibility

VUE supports backward compatibility beginning with version J-2014.12-SP1. Support for a compatible previous version begins with version J-2014.06.

For example, version K-2015.12 loads the results of versions K-2015.06, J-2014.12, and J-2014.06, but it does not load the results of version I-2013.12.

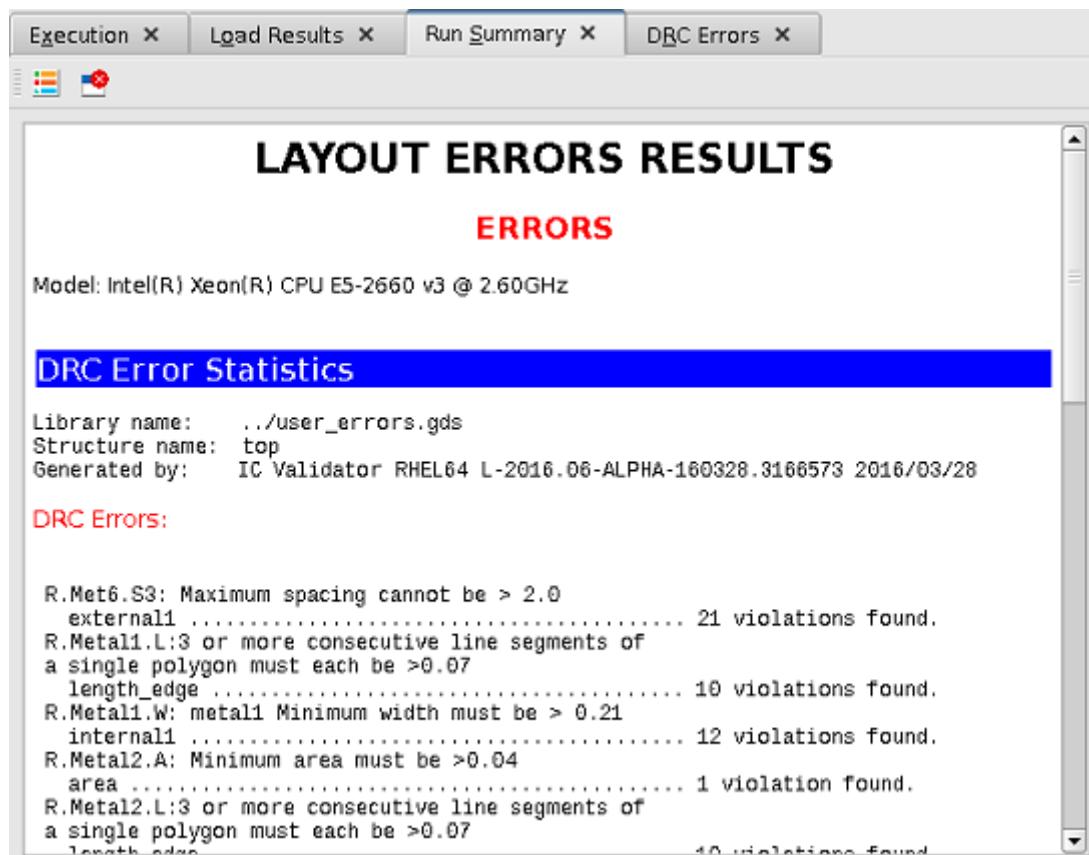
---

## Run Summary

After data is loaded into VUE, additional tabs are added to the debugging flow. The first of these is the Run Summary tab, which provides a high-level summary of the IC Validator run. The first information you see is the top-level result of the run.

If DRC errors exist, a summary of each error and count are given. If LVS errors exist, a count of passed and failed equivalence points is given. [Figure 3-6](#) is an example of the Run Summary tab.

*Figure 3-6 Run Summary Tab With DRC Errors*

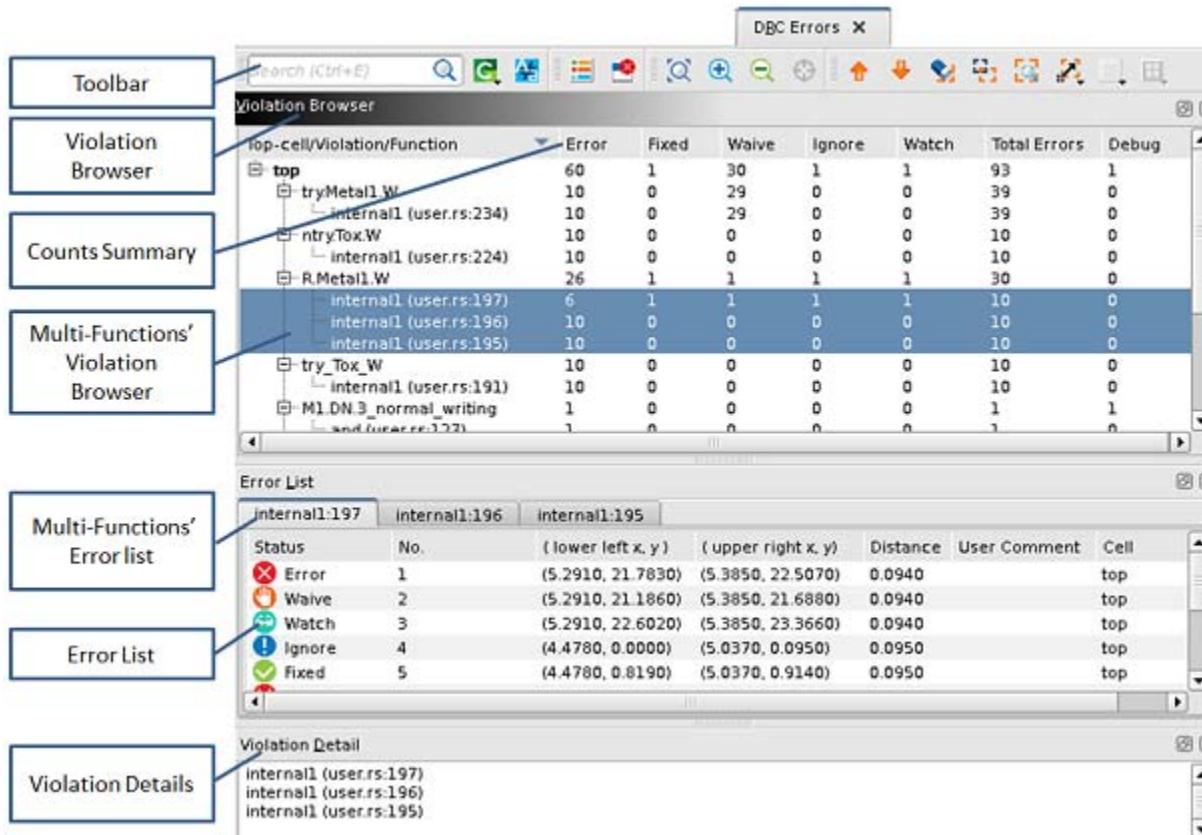


## DRC Errors and Device Extraction Violations

The DRC Errors tab provides an organized tree list of layout cells and DRC and device extraction violations. This section includes the Violation Browser, Error List, Violation Detail,

classification and unmatched error browser. The name of the tab reflects the type of runset used. [Figure 3-7](#) is an example of the DRC Errors tab.

*Figure 3-7 DRC Errors Tab*



---

## Violation Browser Pane

The Violation Browser pane in the DRC Errors tab displays tree-style browsing for cells, violations and functions. The associated counts are also provided on the right for a quick summary view.

You can quickly switch to a different arrangement order, like topcell/violation/function by right-clicking the violation browser cell/violation/function title. Give sort mode or filter functions are also provided in the right-click menu.

You can perform highlighting, classification, or filtering in the browser by right-clicking the cell, violation or function.

---

## Error List Pane

The Error List pane allows you to quickly view all individual errors within one or multiple violations. Individual errors can be highlighted by clicking. Multiple errors can be highlighted simultaneously by Shift-clicking for multiple concurrent errors within the list or Ctrl-clicking separate errors. To highlight all errors quickly within the list, press Ctrl+A. For more information about individual errors when all are highlighted, select one error; VUE zooms to that highlight extent while other highlights remain.

VUE provides complete control following the selected highlights. The Zoom list has three available options that match the Follow mode. For more information about the Zoom, Pan, and None options for the Follow mode, see [Chapter 1, “Introduction to IC Validator VUE.”](#)

---

## Violation Detail Pane

This pane shows the command that generated the violation, the line number, and other information depending on the error type.

---

## Error Classification Control

For every error displayed, you can assign an error classification and user comments by right-clicking. The classification information is written to the IC Validator error database and can be exported for use in a subsequent IC Validator run. The classification comment is written to the User Comment column in the Error List. See Chapter 8, DRC Error Classification, of the *IC Validator User Guide* for more information about this flow.

**Table 3-1** includes descriptions of each error classification.

*Table 3-1 Error Classification*

| Icon | Classification | Description  |
|------|----------------|--|
|      | Error          | Unclassified error (default classification).   |
|      | Ignore         | Error that the designer does not care about: to be fixed later in the process flow.  |
|      | Waive          | Intended design error. It exists at sign off.  |
|      | Watch          | Error that must be fixed in the next design cycle.   |
|      | Fixed          | Error that is fixed in the layout.   |
|      | Unmatched      | Error that exists in an imported error classification database (cPYDB), but was not produced by the current run.               |
|      | Debug          | Errors that are produced by the <code>density(pydb_output = true)</code> argument. This output is used only for VUE debugging. |

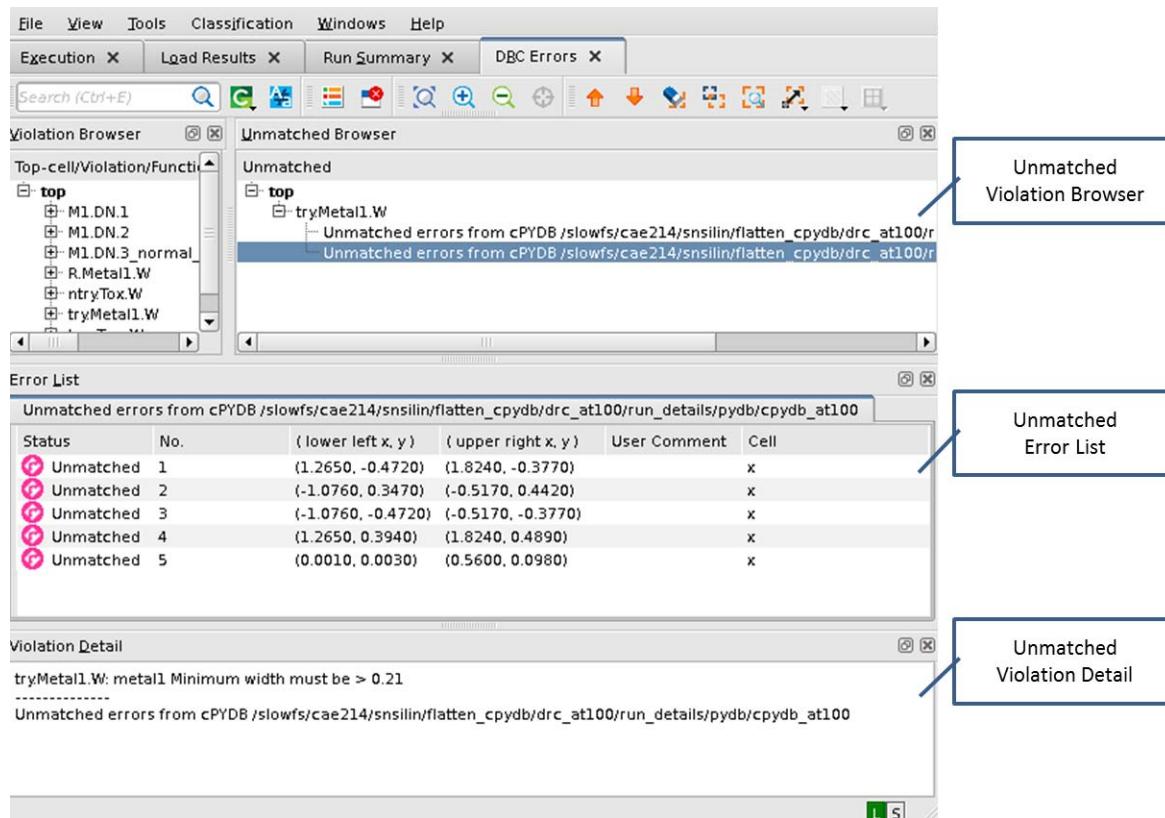
---

## Unmatched Error Classification Browser

Error classification details are available in the VUE DRC error classification browser. See Chapter 10, “DRC Error Classification,” in the *IC Validator User Guide* for more information.

**Figure 3-8** shows the unmatched error classification browser that contains the waiver information, but is not found in the current run. The existing unmatched errors are shown automatically when loading the \*.vue file.

Figure 3-8 Unmatched Error Classification Browser



## DRC Ease-of-Use Support

The VUE DRC interface provides ease-of-use functionality in the following sections:

- [Predefined Keyboard Shortcuts](#)
- [Predefined Mouse Keys](#)
- [Filter Settings](#)
- [Sorting the Error List](#)
- [Sorting the Violation Browser](#)
- [Copying the Error List](#)
- [Searching the Error List](#)
- [Selecting the Error List](#)
- [Adjusting Column Location](#)

---

## Predefined Keyboard Shortcuts

[Table 3-2](#) includes descriptions of each predefined keyboard shortcut.

*Table 3-2 Predefined Keyboard Shortcuts for Ease-Of-Use*

| Icon | Keyboard shortcut | Description                     |
|------|-------------------|---------------------------------|
|      | P                 | Highlight Previous              |
|      | N                 | Highlight Next                  |
| N/A  |                   | Select Previous                 |
| N/A  |                   | Select Next                     |
|      | Ctrl+Shift+C      | Switch to Search Cell mode      |
|      | Ctrl+Shift+V      | Switch to Search Violation mode |
|      | Ctrl+Shift+F      | Switch to Search Function mode  |

---

## Predefined Mouse Keys

LMB (left mouse button). Double-click.

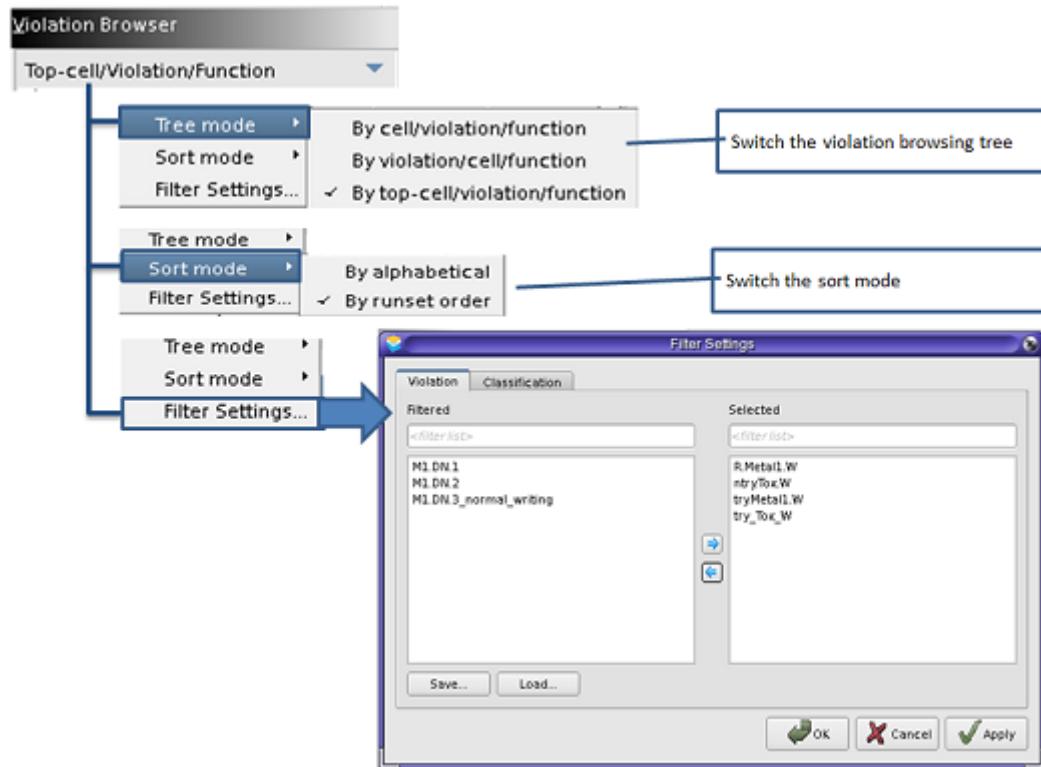
- Apply in Function name; highlights all function errors.
- Apply in Error list; highlights the single error.

RMB (right mouse button). Supports the most useful function within RMB.

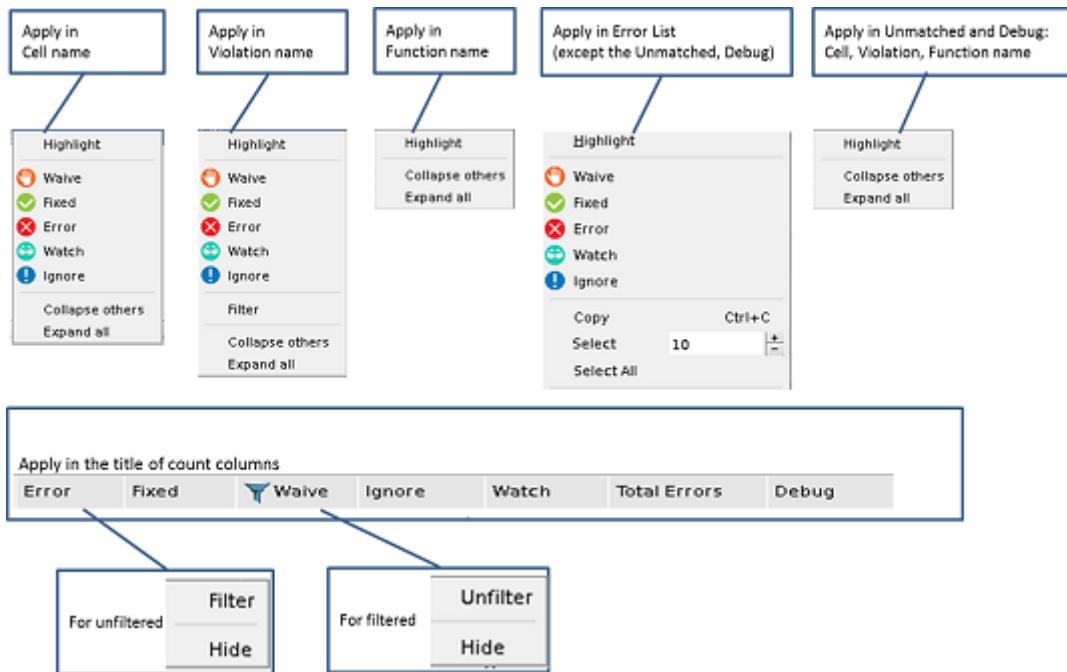
- Apply in Violation Browser title. See [Figure 3-9](#).
- Apply in other than Violation Browser title. See [Figure 3-10](#).
  - Cell, Violation, Function, and Error List
  - Unmatched and Debug
  - Count columns

Figure 3-9 shows RMB: Right-Mouse-Button support for the most useful function within RMB.

Figure 3-9 RMB: Apply in Violation Browser Title



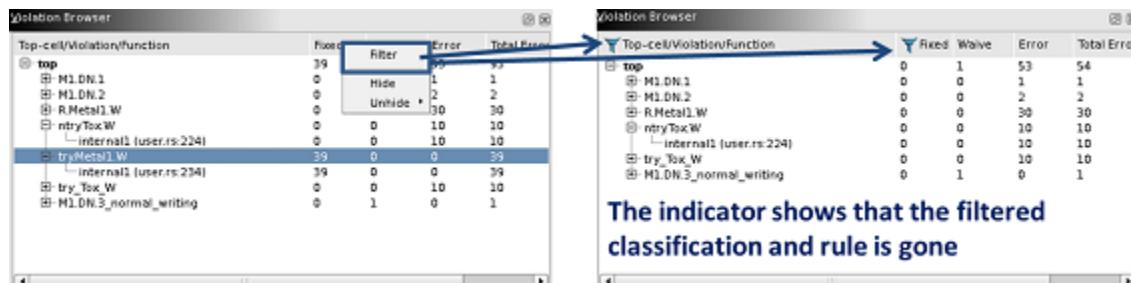
**Figure 3-10 RMB: Apply Other Than Violation Browser Title**



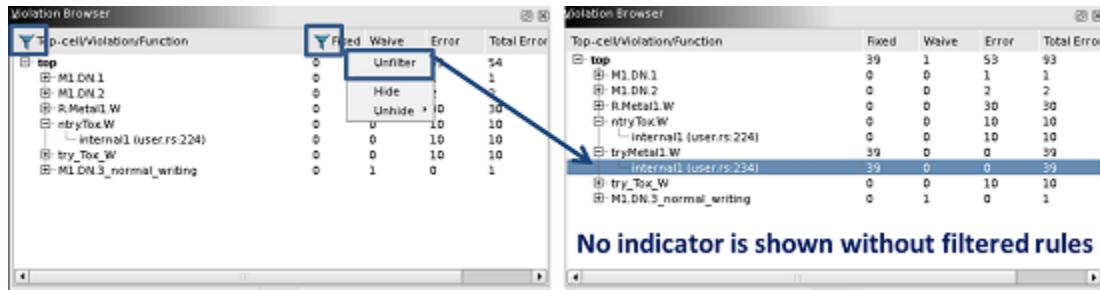
## Filter Settings

Figure 3-11 and Figure 3-12 show you how to quickly filter and unfilter a classification error by right-clicking the titles of the classification count.

**Figure 3-11 Example of Filtering Certain Classifications by RMB**



**Figure 3-12 Example of Unfiltering Certain Classifications by RMB**



You can quickly filter the violations you want to see by right-clicking the selected items in the Violation Browser. Select the unwanted violations and right-click to filter them out, shown in Figure 3-13.

**Figure 3-13 Example of Filtering Certain Violations by RMB**

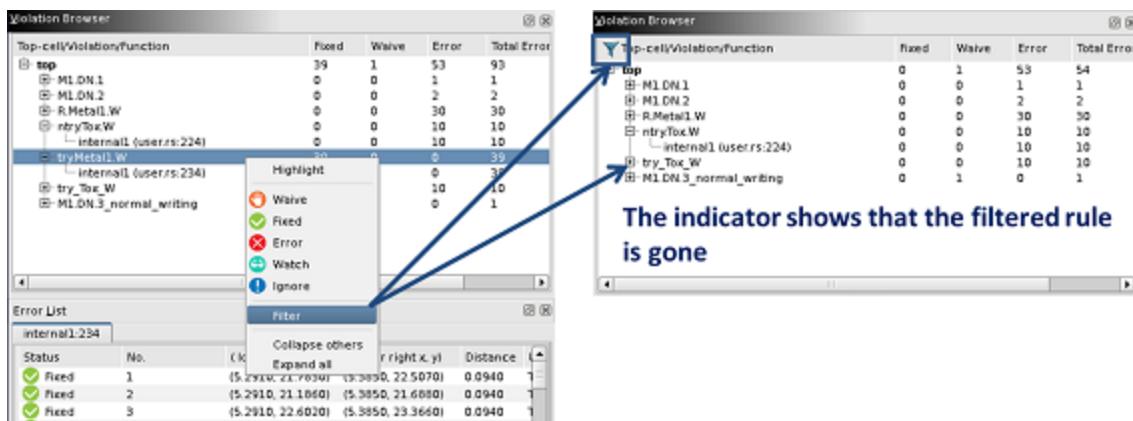
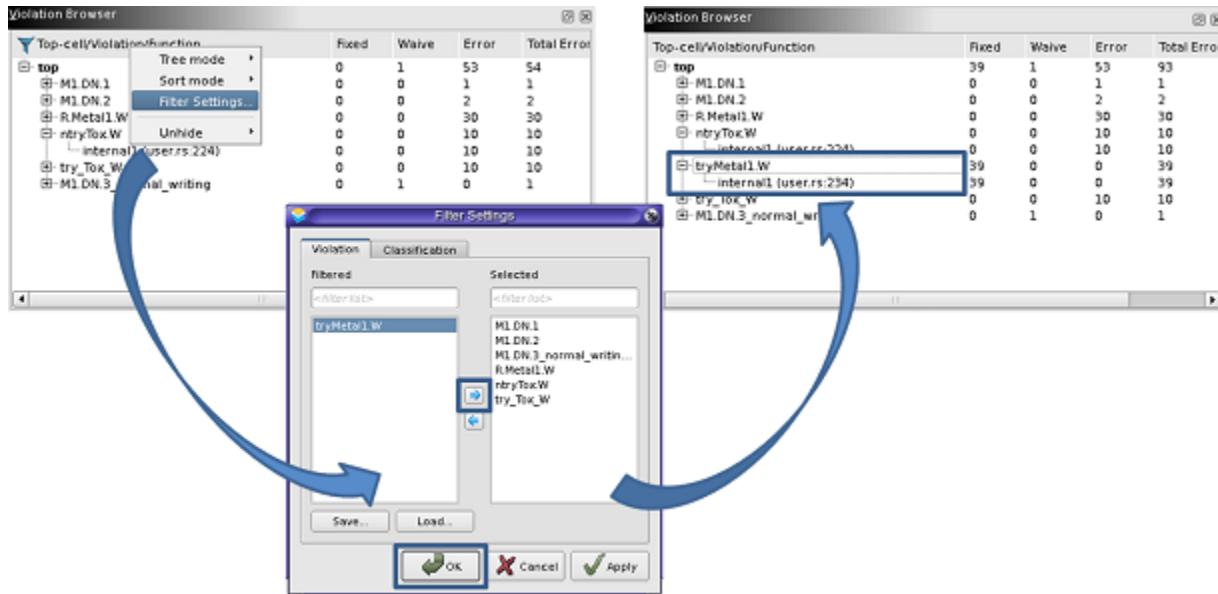


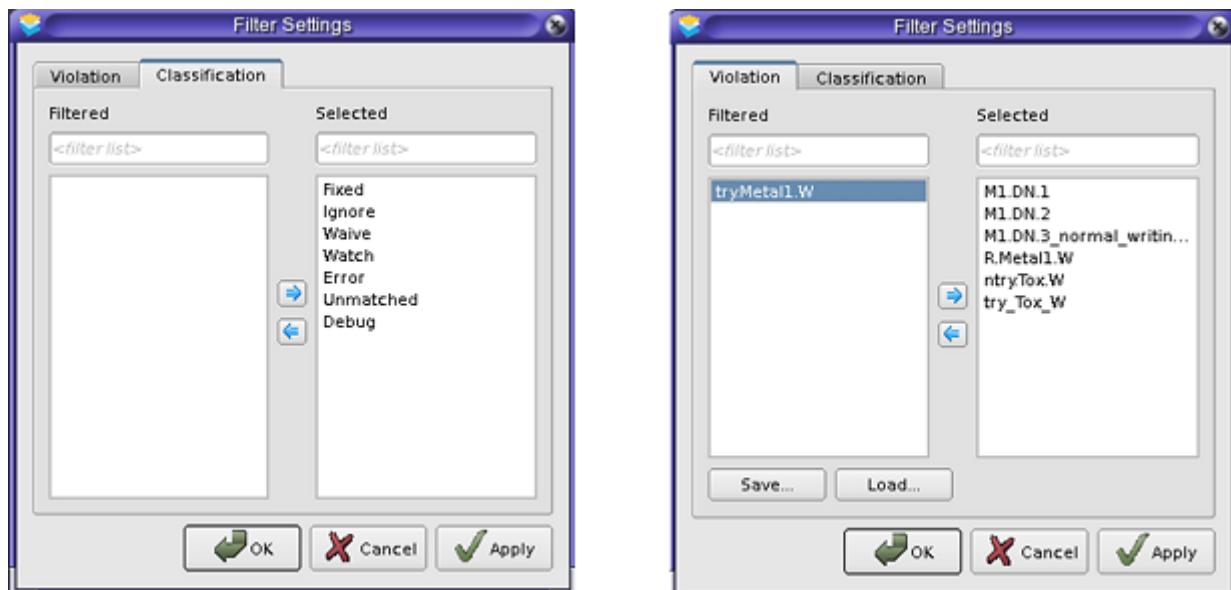
Figure 3-14 shows you how to unfilter violations. This capability can be accessed only from the RMB menu in the first column.

Figure 3-14 Example of Unfiltering by Violation



In addition to the ways described previously, in which you can quickly filter and unfilter classification errors, you can also use the Filter Settings to control the filter for both Violations and Classifications. You can access the Filter Settings from the RMB menu in the first column, shown in [Figure 3-15](#).

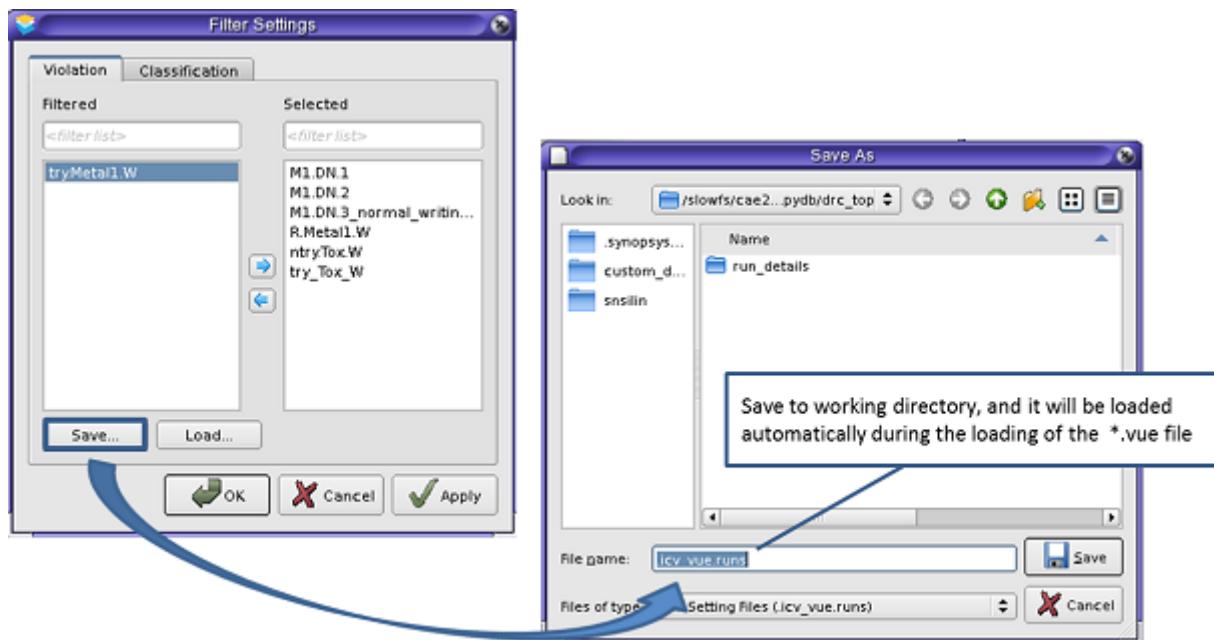
Figure 3-15 Example of Setting All Classifications and Violations



From the Filter Settings, you can save and load your setting for the Violation Filter. This is useful for quickly applying a long list of violation filters across different designs.

Save the Filtered Violation settings in the .icv\_vue.runs Run Setting File, shown in [Figure 3-16](#).

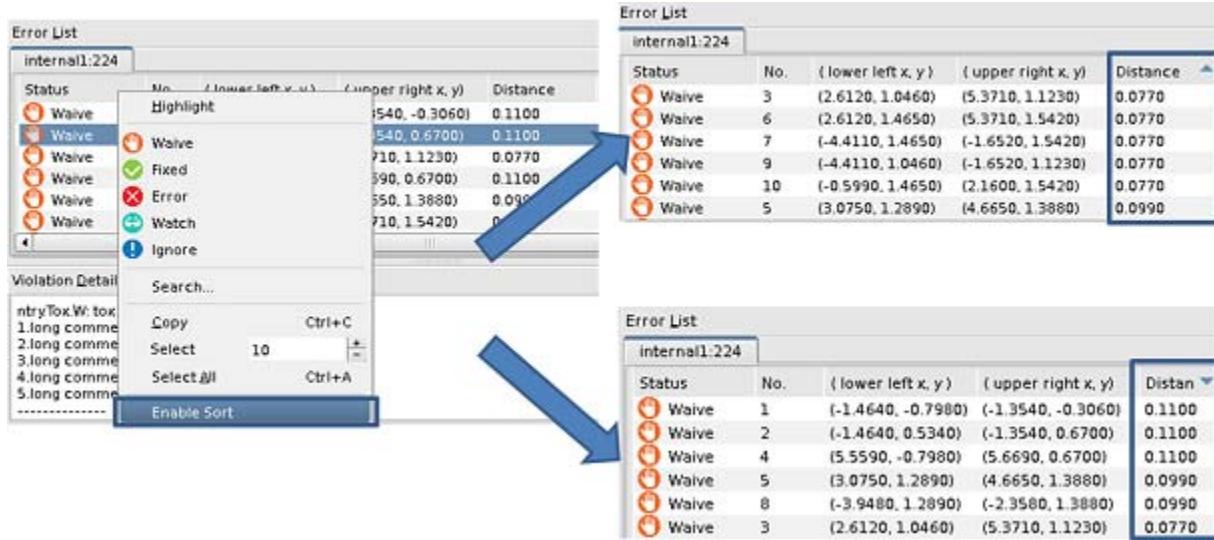
*Figure 3-16 Example of Filtered Violation Settings Saved to Run Settings File*



---

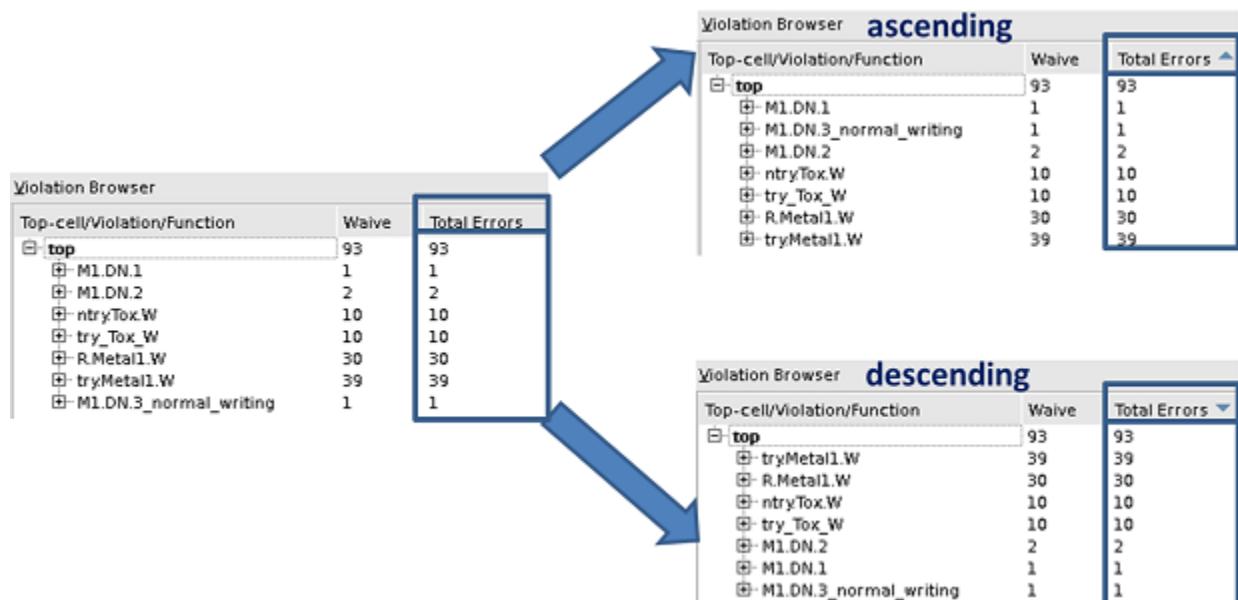
## Sorting the Error List

[Figure 3-17](#) shows how you sort the count in the Error List. With the turn-on feature enabled, sort by right-clicking and left-clicking what you want to sort by category title.

*Figure 3-17 Example of Sorting the Error List*

## Sorting the Violation Browser

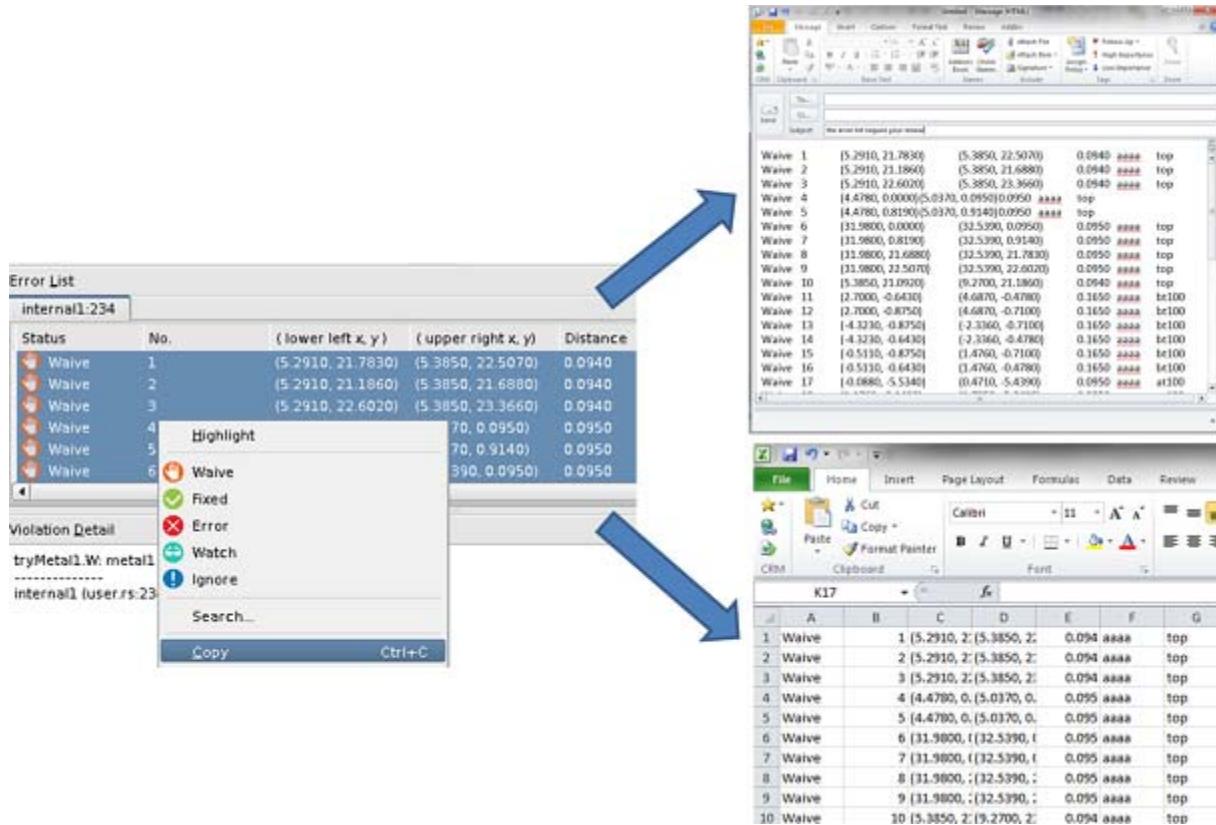
Figure 3-18 shows how you sort the count in the Violation Browser by left-clicking the category title directly.

*Figure 3-18 Example of Sorting the Violation Browser*

## Copying the Error List

[Figure 3-19](#) shows how to copy the Error List information to other text editor, such as an e-mail or in an Excel file.

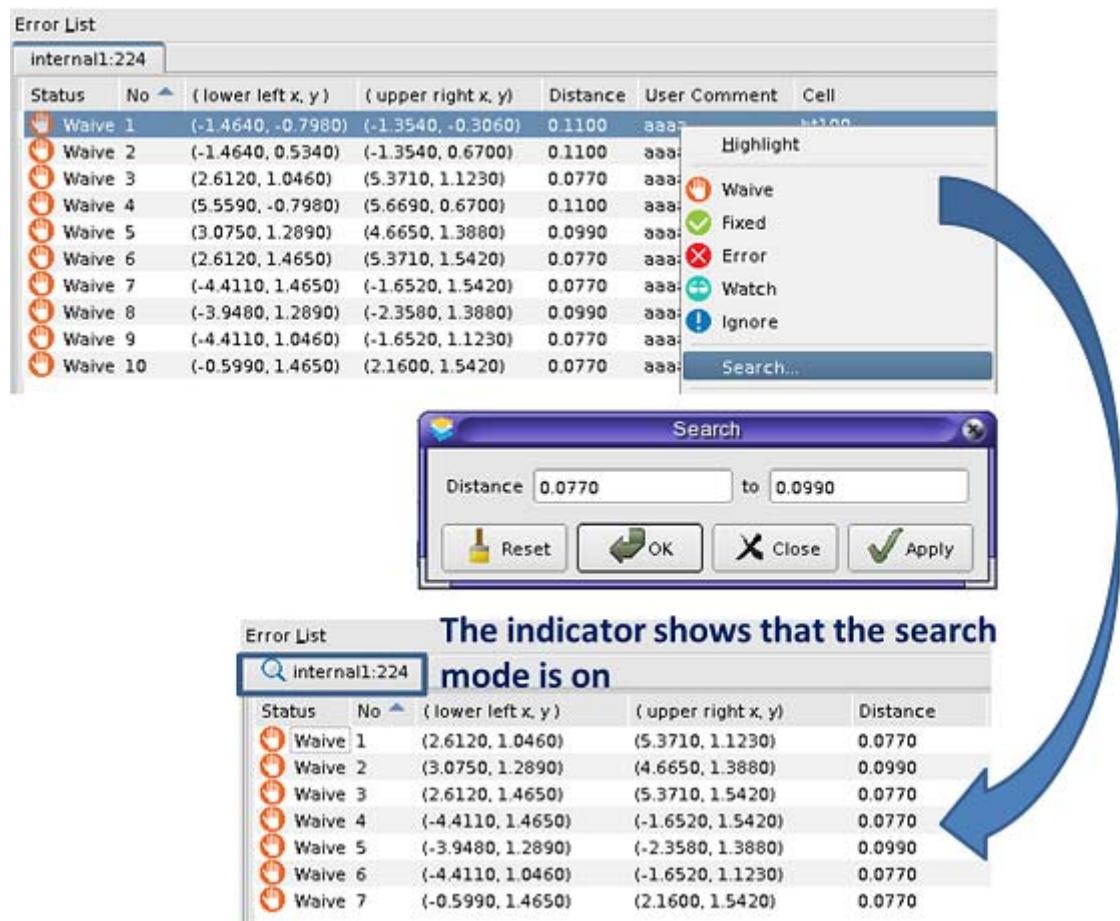
*Figure 3-19 Example of Copying the Error List*



## Searching the Error List

[Figure 3-20](#) shows an example of searching the Error List, as well as how to search limited data from the Error List.

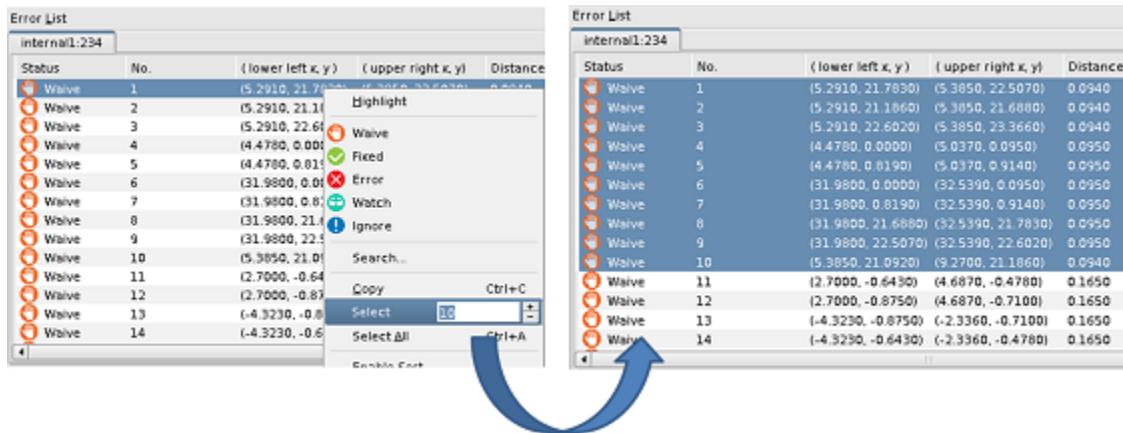
*Figure 3-20 Example of Searching the Error List*



## Selecting the Error List

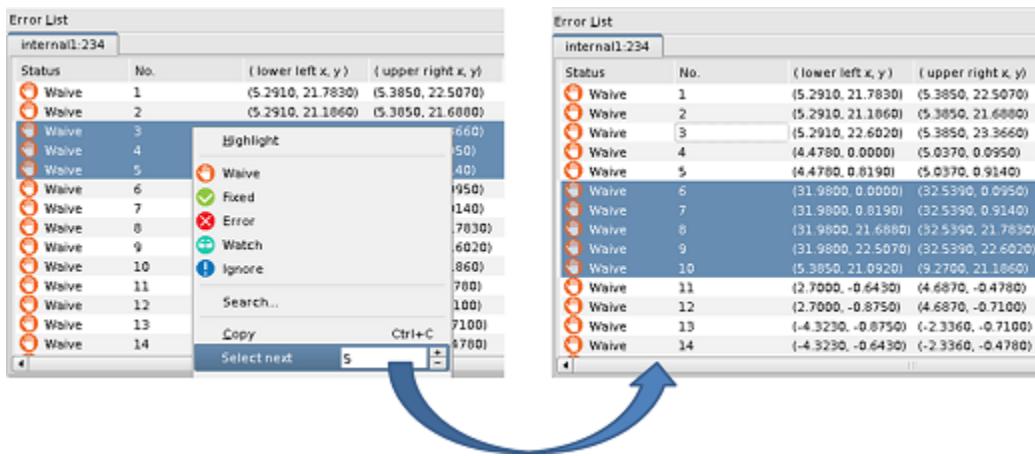
Figure 3-21 shows how to quickly select a series of error lists from the currently selected error.

*Figure 3-21 Example of Selecting the Error List*



**Figure 3-22** shows how to quickly select a series of error lists from the currently selected multiple errors.

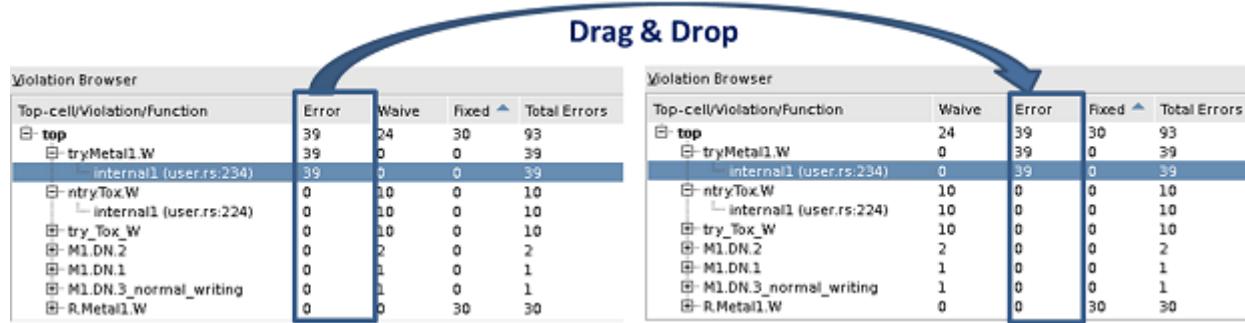
*Figure 3-22 Example of Selecting the Next Error List*



## Adjusting Column Location

**Figure 3-23** shows how to quickly adjust the column location for the Violation Browser

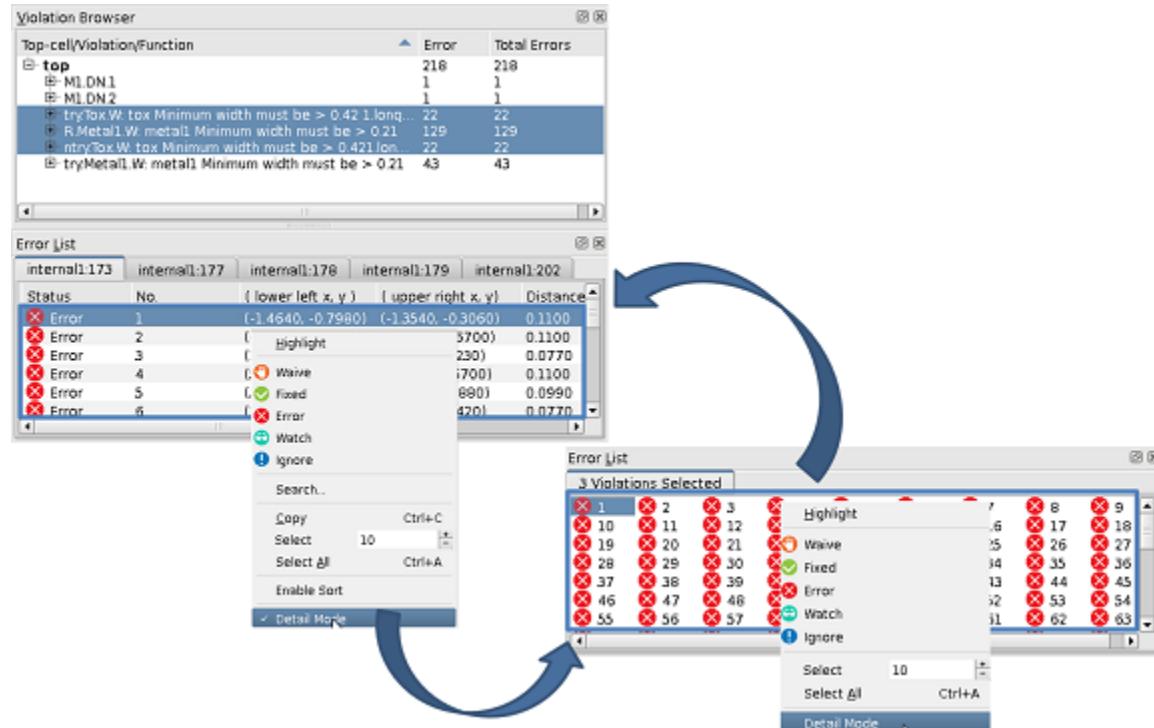
**Figure 3-23 Example of Adjusting the Violation Browser Columns**



## Switching to Non-Detail Mode

Figure 3-24 shows RMB on the error list and quickly switches from detail mode to non-detail mode.

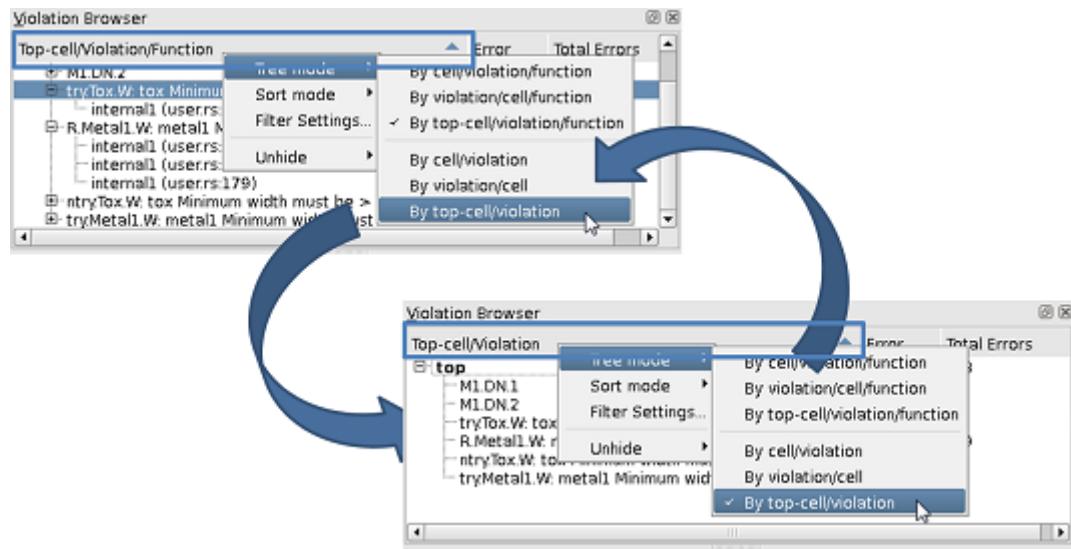
**Figure 3-24 Example of Switching From Detail Mode to Non-Detail Mode**



## Hiding Function Violations

Figure 3-25 shows RMB on the title and quickly hides the function violations.

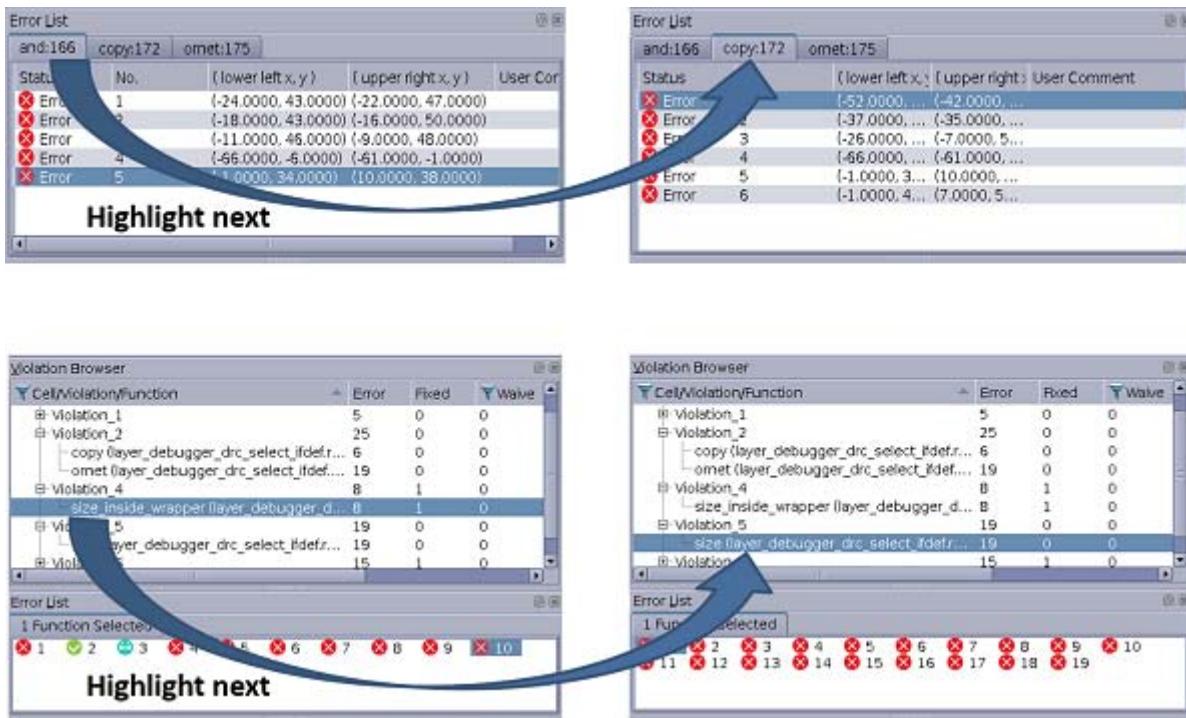
Figure 3-25 Example of Hiding Function Violations



## Iterative Highlighting

Figure 3-26 shows iterative highlighting in the detail and non-detail modes.

**Figure 3-26 Example of Iterative Highlighting**



## The soft\_check Graphical Violation Report

The soft\_check violations can be viewed within the Extraction Errors or DRC Errors tab. The number of soft\_check violations reported is based on the number of violating layer2 polygons.

### Note:

The following examples use the Cell/Violation Tree mode of VUE. Along with other arrangements of violations displayed, you might see slightly different menu arrangements.

Multiple soft\_check violations can be accessed one at a time by selecting the error in Error List.

- VUE highlights and zooms to the layer2 polygon extents when a soft\_check violation is initially loaded by selecting the soft\_check error in Error List.
- Color and Fill selection is available for layer2.

For each soft\_check violation, layer1 nets are sorted from the highest to lowest polygon counts, starting from the top of the list in the soft\_check pane.

- Color and Fill selection is available for layer1 on a per net basis.
- Nets with lower polygon counts have an increased probability of invalid soft\_check nets.
- All layer1 nets in violation can be selected using the select-all button.

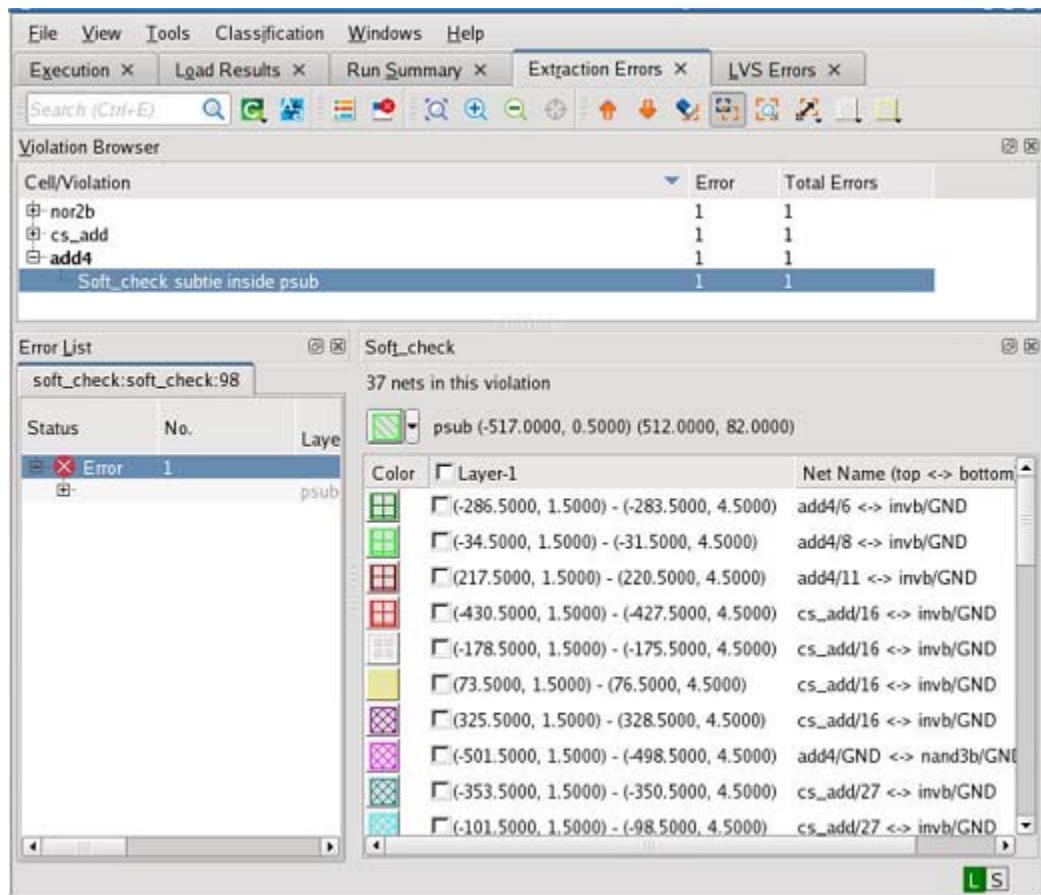
For each layer1 net in violation:

- Click the layer1 coordinates to zoom to the layer1 polygon. Check the layer1 check box to zoom to the extents of all layer1 polygons with active check boxes.
- Net paths for each layer1 polygon are reported in the Netname (top <-> bottom) column

[Figure 3-27](#) shows one loaded soft\_check violation and the highlight and zoom features that are available from the Highlight list for each soft\_check violation.

- The number of violating soft\_check nets
- The layer2 layer name and coordinates along with a color selection widget

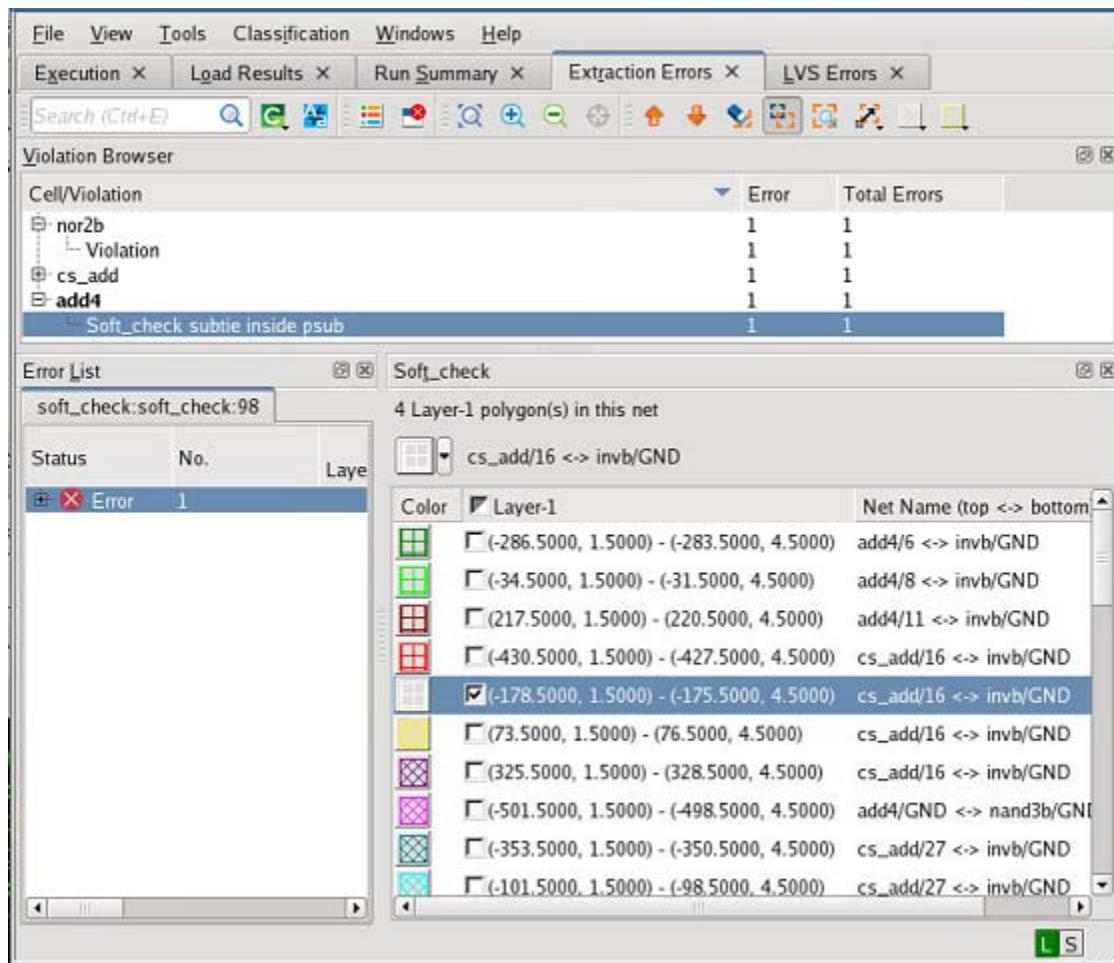
*Figure 3-27 Loading a soft\_check layer2 Violation*



**Figure 3-28** shows the DRC Errors page when a soft\_check violation is loaded. The tab displays:

- The number of layer1 polygons on net invb/GND
- A color selection widget for each soft\_check net
- A global check box that activates or deactivates all layer1 check boxes, or the select all button.
- Highlight and zoom control
- Net paths from top-to-bottom for layer1

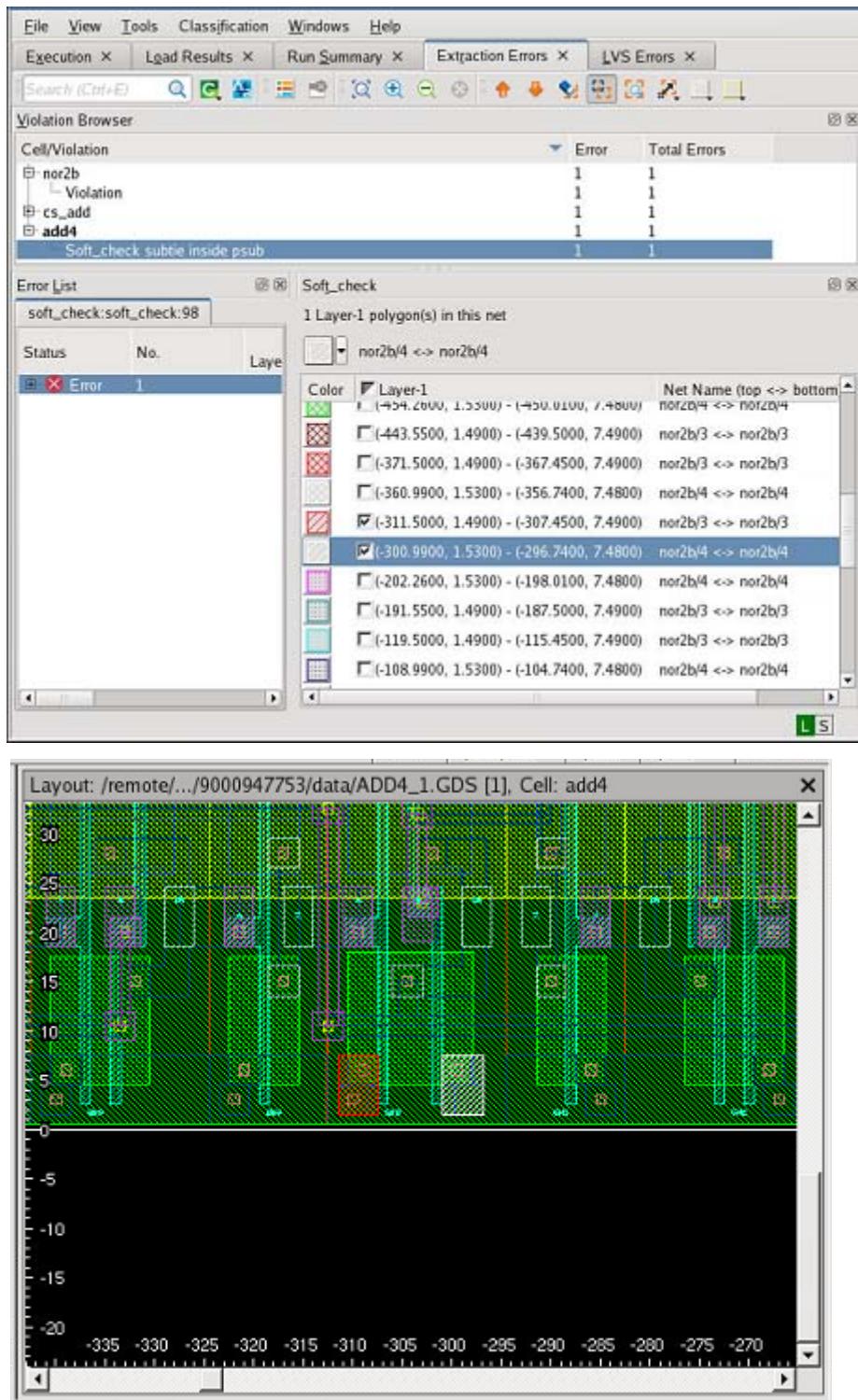
*Figure 3-28 Loading a soft\_check layer1 Violation*



**Figure 3-29** shows how the highlights for soft\_check might look when highlighting both violating nets (layer1) in a violation (layer2):

- Layers and colors:
  - Layer2 color is green
  - Layer1 color on the first soft\_check net is white
  - Layer1 color on the second soft\_check net is red

Figure 3-29 Cumulative Layer Highlight Zoom Control



## LVS Errors

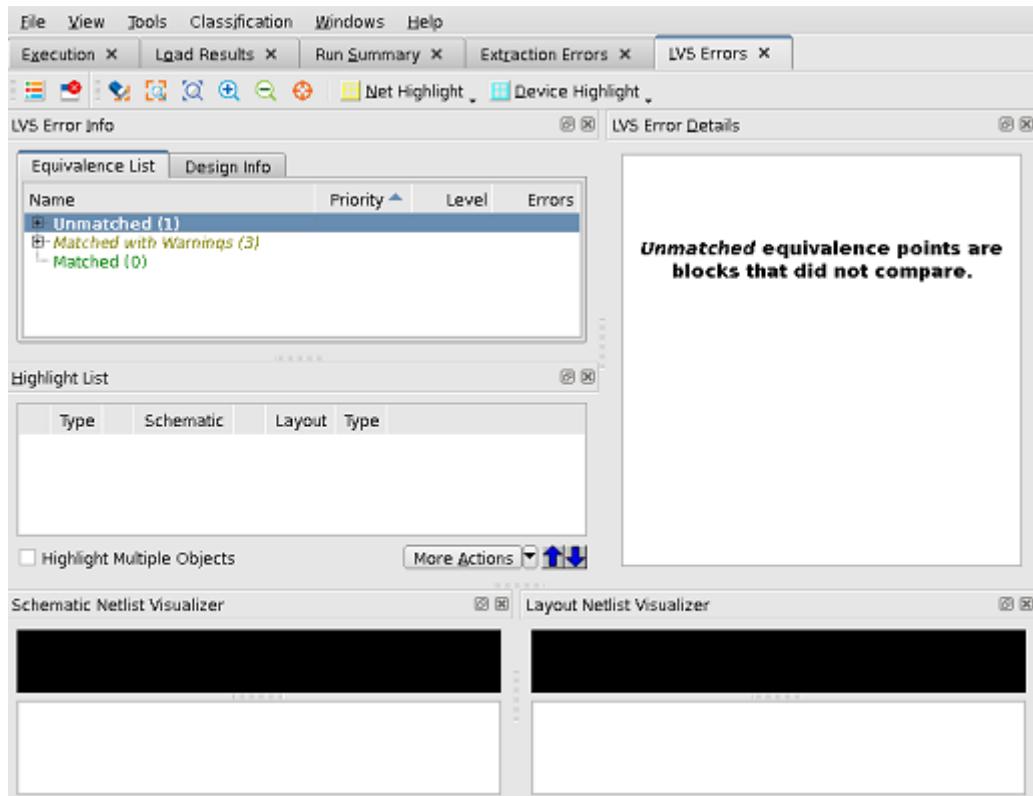
The LVS Errors tab provides an organized list of cells and LVS violations. The error viewing section includes the Design Level Info with Equivalence List, and Highlight List panes to the left as well as the main window, shown in [Figure 3-30](#).

Note:

To highlight nets and devices when debugging LVS errors, VUE requires IC Validator to create an output library with the polygon data. This output library is automatically created if the `create_vue_output` argument is `true`.

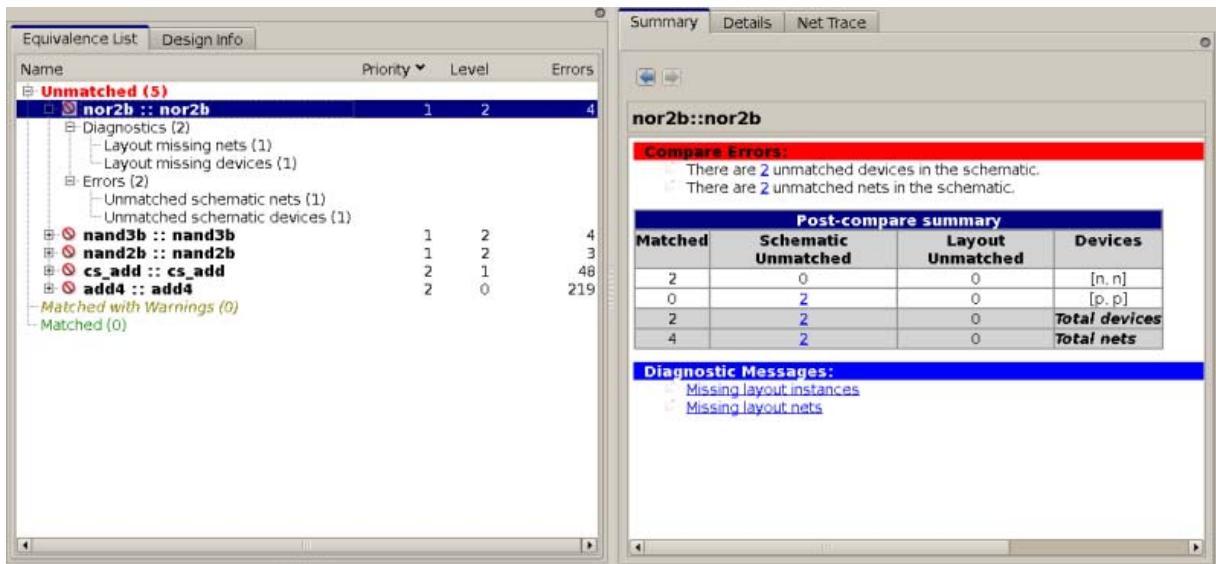
The Equivalence Cell Pair Error browser provides sorting of data by column and color-codes items in the equivalence list. You can sort on the Name, Priority, Level, or Errors columns, but the default sorting information is on the Priority column. The rows include unmatched or matched blocks.

*Figure 3-30 LVS Errors Tab*



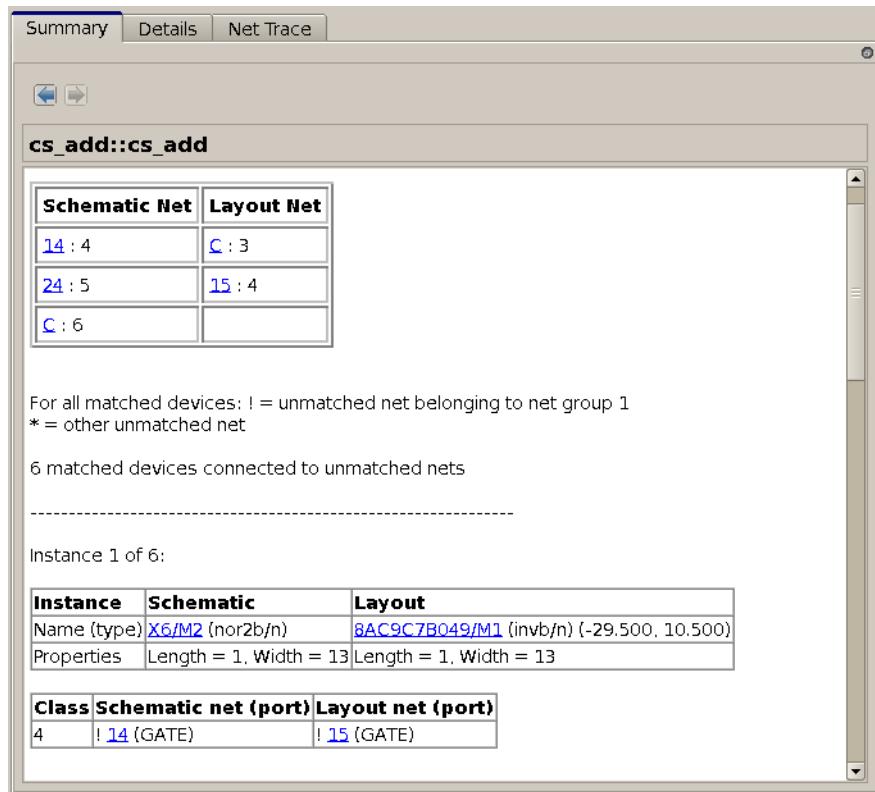
When you select an equivalence cell pair, shown in [Figure 3-31](#), diagnostic messages and equivalence errors are displayed in the Summary view. You can expand or collapse all display levels in this pane.

*Figure 3-31 Viewing LVS Errors for an Equivalence Cell Pair*



When you select an equivalence point, the information in the Summary view, Details view, and Netlist browser is updated.

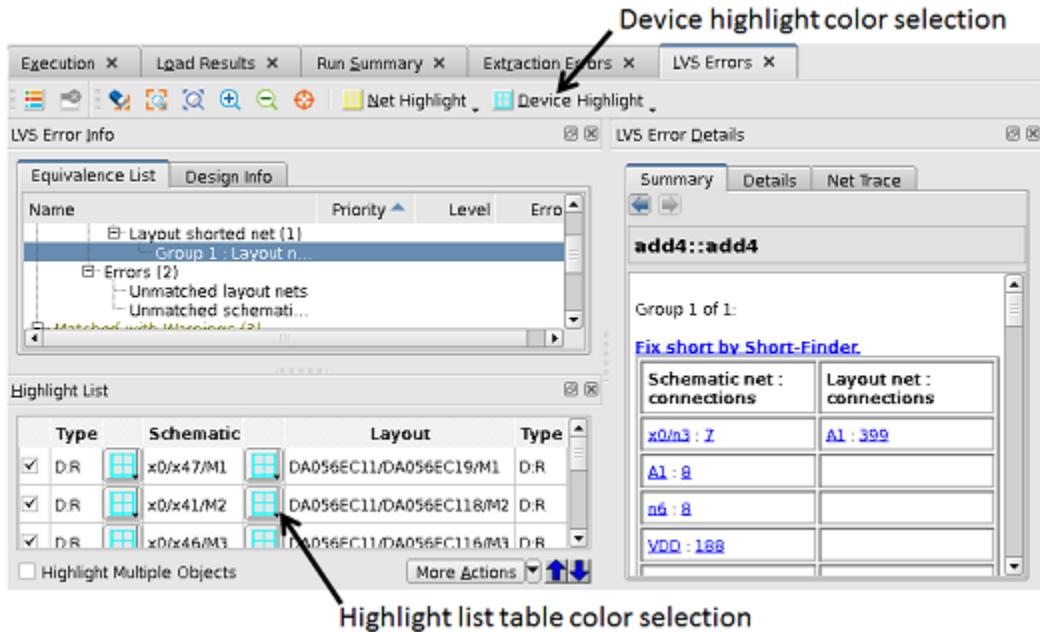
The Summary view, shown in [Figure 3-32](#), behaves like an HTML browser. For example, blue underlined objects act as links that can open new content in the Summary view or highlight objects in the Schematic or Netlist browser. Back and forward arrows provide easier navigation to the previous or next pages.

**Figure 3-32 Summary Tab**

## Device Grouping

The color of the device grouping highlight is set by the Device Highlight color selection. The groups of devices are highlighted in the layout window with the color displayed, shown in the Highlight List table in [Figure 3-33](#).

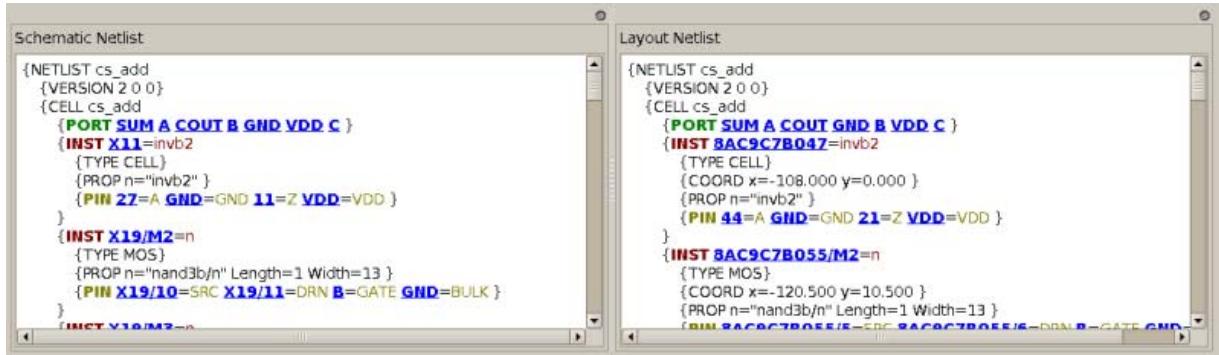
Figure 3-33 Device Grouping



The Device Highlight color selection is applied to new objects added to the Highlight List. This setting can be changed, as can the setting for the particular object in the Highlight List. Each device is drawn according to its currently defined color in the Highlight List table.

## Netlist Browser

The Netlist browser displays the cell-level schematic and layout netlists for the equivalence point. This netlist is generated during the LVS run. It is a flat netlist where all blocks that have been successfully compared at a lower level are now only black box instances. When debugging a comparison error, using both netlists for cross-probing or probing in the layout is helpful. By default, the equivalence netlists are generated only for failed equivalence points. See [Figure 3-34](#) for more information.

**Figure 3-34 Netlist Browser**

To cross-probe between the schematic and layout, select a net or device in either netlist. If it is a matched net or device, the Highlight List pane shows the details shown in [Figure 3-35](#). If it is an unmatched net or device, the details are not displayed.

To search the netlists, right-click in the highlight List to open the find window. This window keeps a history of highlighted objects, and you can easily jump between highlighted objects.

VUE automatically finds and scrolls to the first highlight that exists. VUE also allows you to tell it which net or device it should zoom, and which occurrence of that net or device should be highlighted.

## Highlight List

The VUE Highlight List pane gives you complete control highlighting nets and devices for LVS. When a net or device is highlighted, it is listed here with cross-reference information, highlight color and fill information, and the type of object being highlighted. If the layout or schematic name is N/A, a matched cross-reference for that device or net does not exist.

The Highlight List pane displays all nets and devices that are currently highlighted. Each highlight has a selection check box that toggles the individual highlight on or off. The Schematic and Layout instance or net name and type are provided.

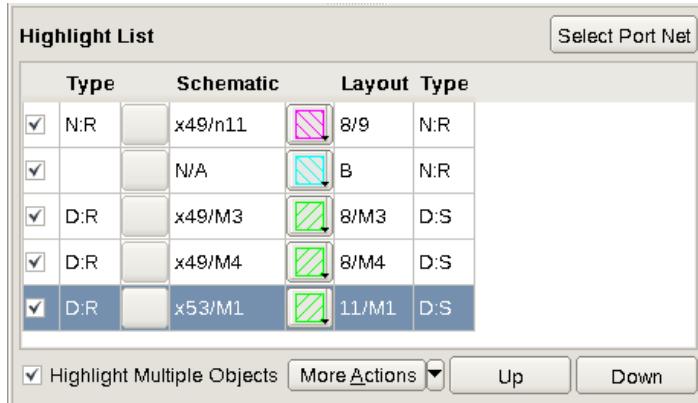
The types provided in the highlight list can be

- D:R for a real device.
- D:C for a composite device.
- D:P for a device that is pushed down from upper-level cells.
- N:R for a real net.
- N:C for a composite net (from `push_down_pins()` or `short_equivalent_nodes()` functions).

To highlight more than one object at a time, click the Highlight Multiple Objects check box below the highlight list. When highlighting a large net, the status bar in the lower-right corner tracks the progress of the highlight being displayed in the layout editor.

If more than one highlight is listed, select click to select a range of objects, or Ctrl-click to select multiple, nonconsecutive objects. Each color and fill highlight can be changed without removing the highlight by selecting the color and fill combo box, shown in [Figure 3-35](#).

*Figure 3-35 Highlight List for LVS Errors*



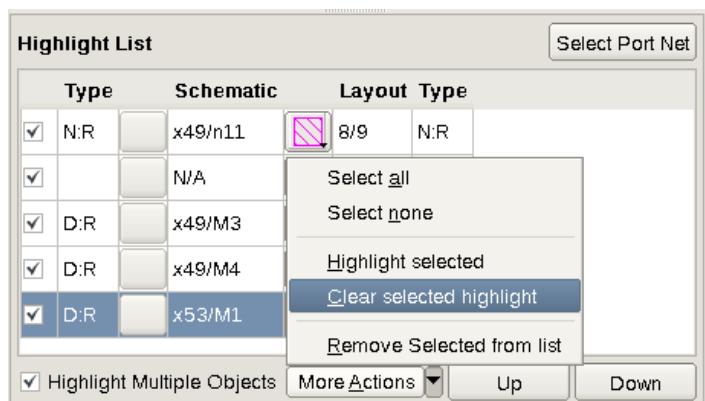
For additional highlighting control, right-click in the highlight list and choose either Select Layer List or Select Area. The LVS options window appears, and you can turn off highlighting layers or limit the highlighting to a specific region.

Note:

The LVS options apply to *all* highlights.

As shown in [Figure 3-36](#), the More Actions list in the Highlight List pane displays a list of actions you can perform on LVS errors.

*Figure 3-36 Highlight List Options for LVS Errors*



**Table 3-3** describes the actions that are available from the More Actions list.

*Table 3-3 More Actions*

| Action                    | Description  |
|---------------------------|--|
| Select all                | Selects each row in the highlight list. The current highlights that are drawn do not change.                       |
| Select none               | Removes each row selection.  |
| Highlight selected        | Draws highlights for each selected row in the highlight list.  |
| Clear selected highlight  | Removes the highlight that is drawn from the layout or schematic editor. The objects remain in the highlight list. |
| Remove selected from list | Clears highlight that is drawn and removes the object from the highlight list.                                     |
| Up                        | Moves the selected highlight rows up the list.   |
| Down                      | Moves the selected highlight rows down the list.   |

## Details Tab

The Details tab shows the compare summary of a cell. You can either scroll or search the information by using the text list in [Figure 3-37](#).

*Figure 3-37 Details Tab for LVS Errors*

The screenshot shows the 'Details' tab of the IC Validator VUE interface. At the top, there are three tabs: 'Summary', 'Details' (which is selected), and 'Net Trace'. Below the tabs, the title 'nand3b::nand3b' is displayed, along with a 'Compare' button. The main area contains two tables: 'Layout Netlist Statistics' and 'Post-Merge Netlist Statistics'.

**Layout Netlist Statistics**

| Initial | PushDown | Filter | Parallel | Path/Ser | Final | Device               |
|---------|----------|--------|----------|----------|-------|----------------------|
| 3       | 0        | 0      | 0        | 0        | 3     | n                    |
|         |          |        |          |          |       |                      |
|         |          |        |          |          |       |                      |
|         |          |        |          |          |       |                      |
| 3       | 0        | 0      | 0        | 0        | 3     | <b>Total devices</b> |
| 8       | 0        | 0      | -1       | 0        | 7     | <b>Total nets</b>    |

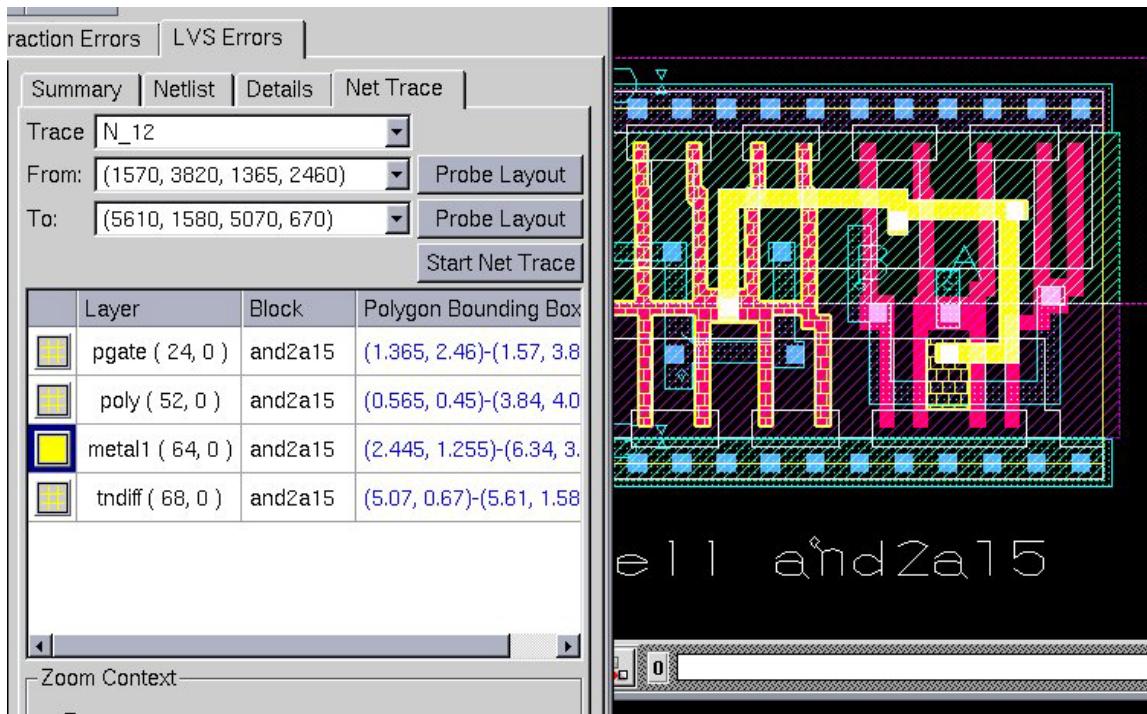
**Post-Merge Netlist Statistics**

| Schematic | Layout | Devices              |
|-----------|--------|----------------------|
| 3         | 3      | [n, n]               |
| 3         | 0      | [p, p]               |
|           |        |                      |
|           |        |                      |
| 6         | 3      | <b>Total devices</b> |
| 8         | 7      | <b>Total nets</b>    |

## Net Trace Tab

The Net Trace tab within the LVS Errors section of VUE enables you to walk a net, polygon by polygon, shown in [Figure 3-38](#). This can be very useful when working with isolated untexted shorts or device connection issues.

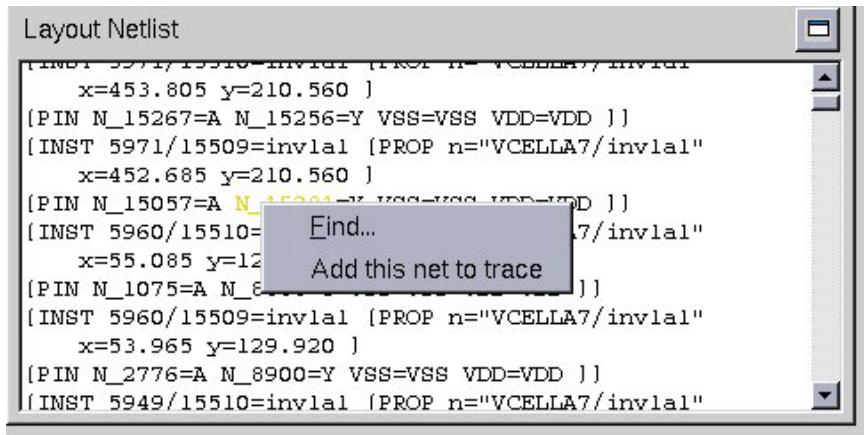
*Figure 3-38 VUE Net Trace Tab*



To begin, select a net from the Trace list box. All nets within the equivalence netlist are listed alphabetically in this list. The net is highlighted in the layout editor upon selection.

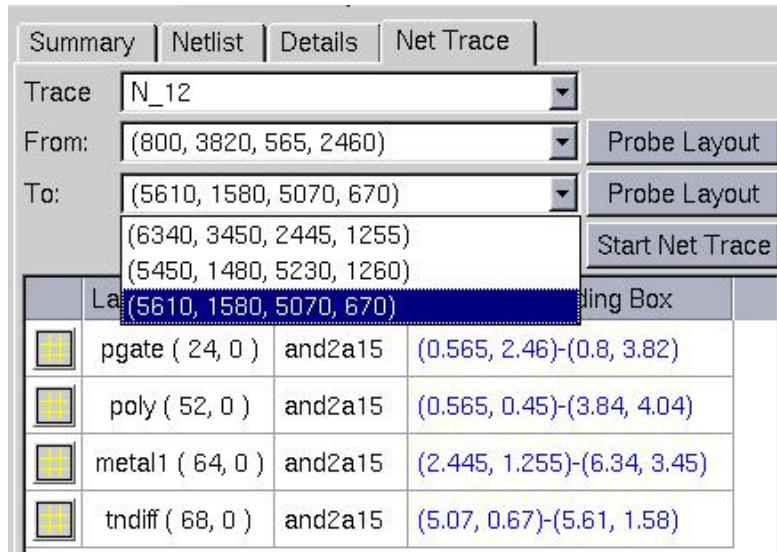
If there are more than 100 nets within the cell, the Trace list is empty. To populate the list, right-click a net in the Layout Netlist pane and choose Add this net to trace, shown in [Figure 3-39](#).

*Figure 3-39 Layout Netlist Pane*



Next, and as shown in [Figure 3-40](#), click the Probe Layout button to select a starting point to start tracing. The bounding box of the polygon is added to the From list box. If more than one polygon exists beneath the point selected, you can choose which one to use in the trace from within the To list box, shown here.

*Figure 3-40 Selecting the Starting Polygon Within Net Trace*



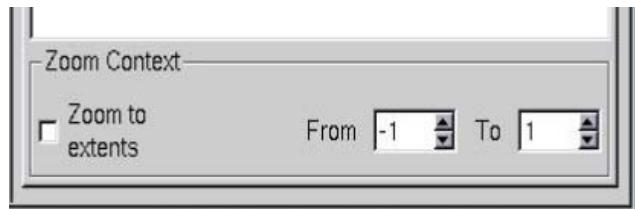
When a new starting box is selected, it is highlighted in the layout to help you identify it.

When both From and To lists are filled, click Start Net Trace to get the list of polygons. The path is highlighted in the layout. The color and fill pattern can be changed by right-clicking the colored box on the left side of the table.

Click an individual polygon to highlight it in a bolder color within the layout. Use the Shift key to highlight more than one polygon at a time. Use the Ctrl key to select nonconsecutive polygons in the list.

The Zoom Context section, shown in [Figure 3-41](#), is located at the bottom of the Net Trace tab and controls how VUE zooms the selected polygons. Zoom to extents uses the From and To settings to control the extents of the zoom. The range specifies the extents of the polygons before and after the selection in the list.

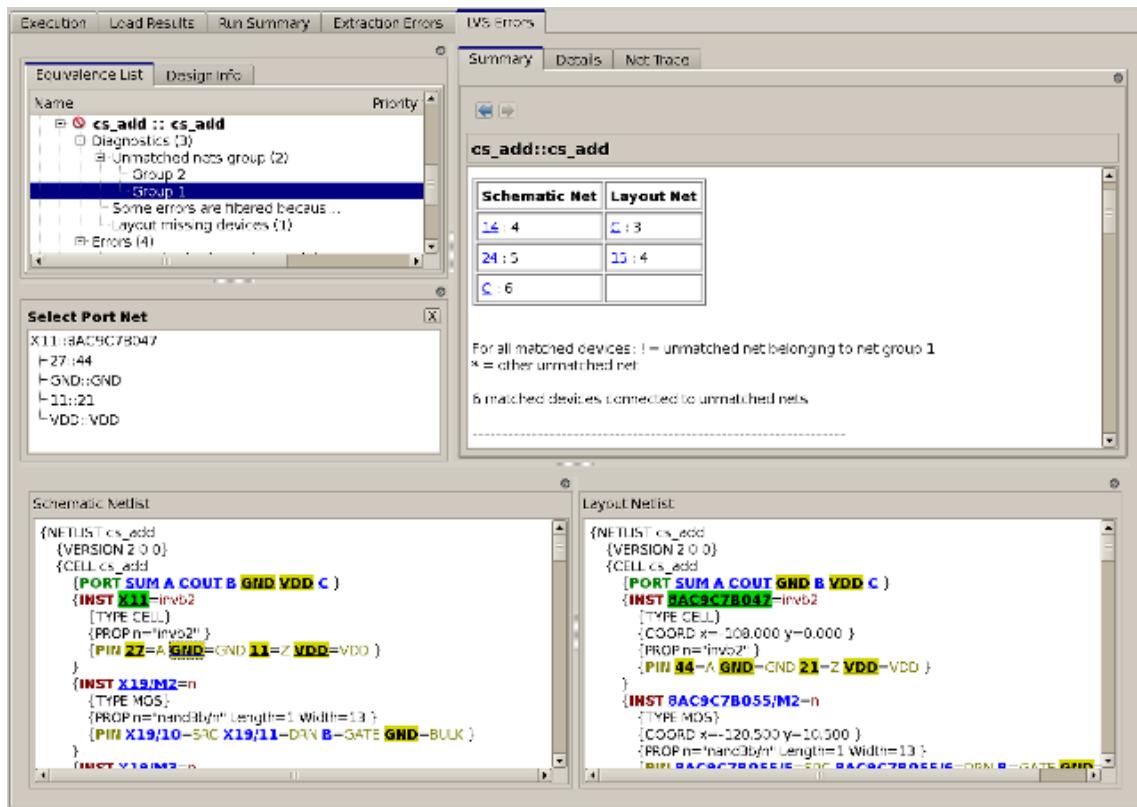
*Figure 3-41 Zoom Context Section*



## Select Port Net

The Select Port Net utility, shown in [Figure 3-42](#) enables you to highlight a specific device, along with all nets attached to the ports or pins of that device. The associated cross-reference information is also displayed as shown here.

*Figure 3-42 Select Port Net*



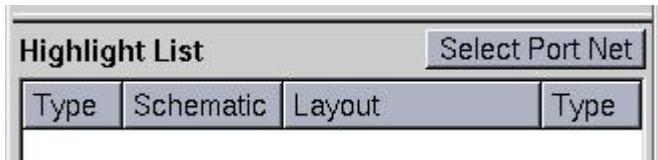
Each element can be individually highlighted from the tree structure listed in the Select Port Net pane.

**Note:**

If a net is connected more than one time to a device, that net is listed only one time. An example of this is a PMOS whose source and bulk are connected to VDD.

To use this utility, click the Select Port Net button in the Highlight pane, shown in [Figure 3-43](#). Click a device within the Summary tab or Netlist browser.

*Figure 3-43 Select Port Net Button*



To close the Select Port Net frame, click the small x in the upper-right corner. The Highlight List pane returns.

---

## Netlist Visualizer

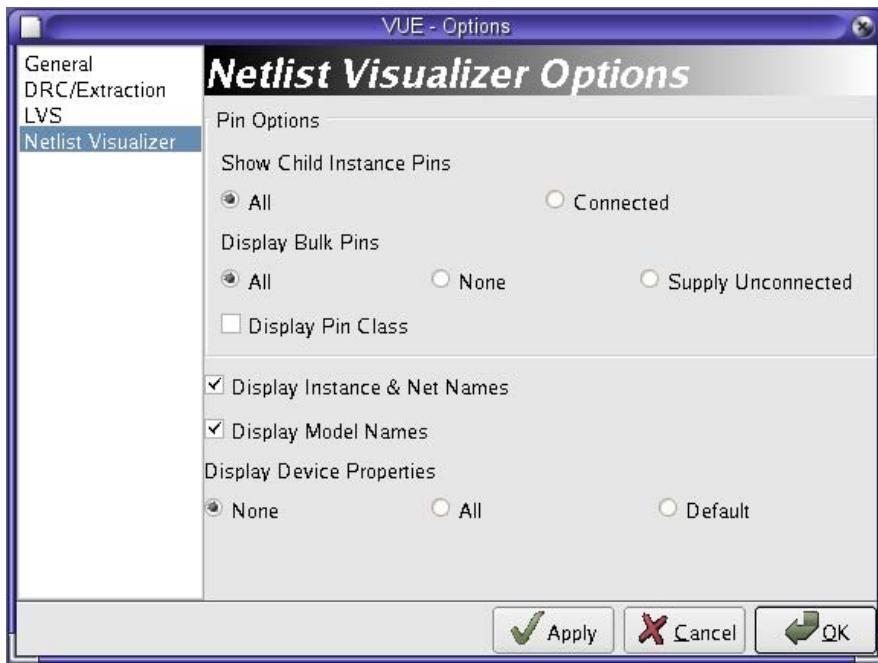
The Netlist Visualizer feature provides an embedded VUE interface to graphically and interactively render a section of a netlist. This interface can draw and incrementally expand a small section of the circuit depending on the debug task.

The Netlist Visualizer draws standard devices and ports in the graphical canvas using engineering symbols. The interface draws merged devices, nonstandard devices, and child cells as rectangles. Nets are represented as orthogonal lines between points. All devices, cells, ports, and pins are annotated with information from the netlist.

## Netlist Visualizer Options

The Netlist Visualizer Options dialog box is shown in [Figure 3-44](#).

*Figure 3-44 Netlist Visualizer Options*



The following options are available in the Netlist Visualizer dialog box:

- **Show Child Instance Pins.** There are two options: All child pins (the default) or connected child pins. This option appears in the Options window, and is also available from the right-click menu.
- **Display Bulk Pins.** There are three options: All bulk pins (the default), no bulk pins, or bulk pins that are not connected to the supply.
- **Display Pin Class.** Pin class numbers are either displayed or hidden (the default).
- **Display Instance & Net Names.** Shows names (the default) or hides names.
- **Display Model Names.** Toggles device model names.
- **Display Device Properties.** There are three options: No properties (the default), all properties, or the default properties, where default means the typical properties for a standard device (for example, width and length for a standard MOS transistor.)

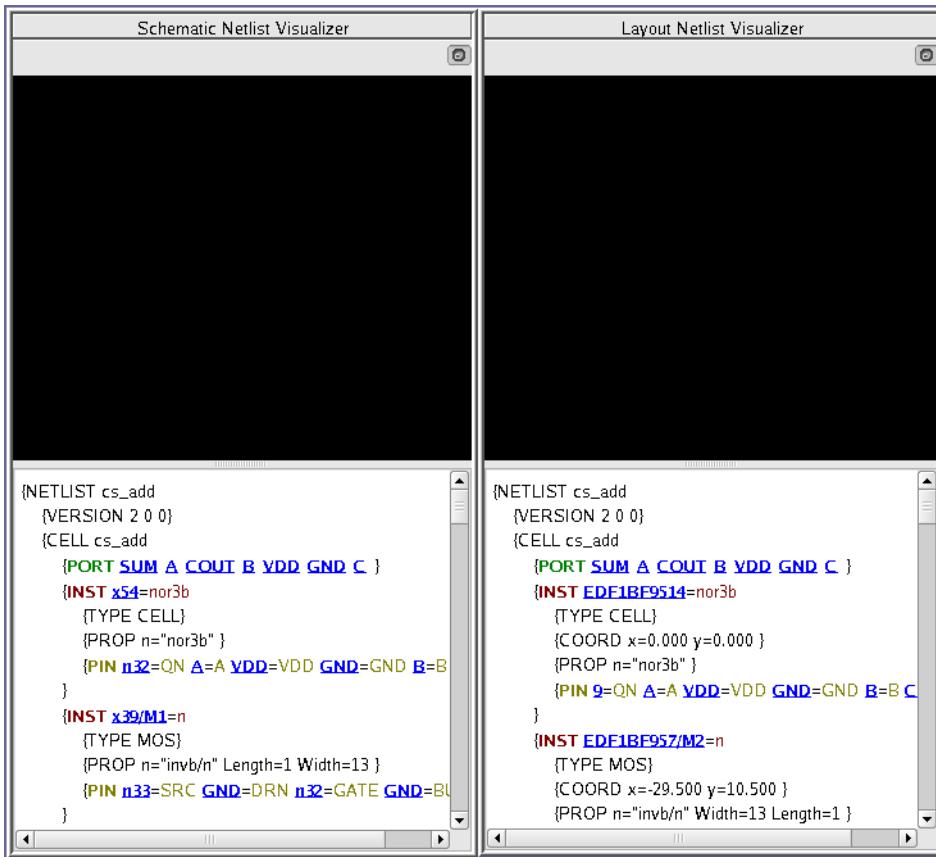
## Netlist Visualizer Interface

When LVS information is present in the VUE file, the LVS Errors section is displayed automatically. The Netlist Visualizer interface does not appear until you select an equiv from the Equivalence List.

After you select the equiv from the Equivalence list, two similar panes appear at the bottom of the VUE window: the first is on the left and is labeled, Schematic Netlist Visualizer; the second is on the right and is labeled, Layout Netlist Visualizer. Each of these panes contains two panes - the top pane renders the netlist graphically and the bottom pane presents the netlist as text.

The text panes appear on startup with a text representation of the equivalence netlist. The graphic panes appear empty on startup, shown in [Figure 3-45](#).

*Figure 3-45 Netlist Visualizer With a Blank Canvas*



You can undock and stretch the two panes to provide a better working environment. For more information about undocking a pane, see the “Windows Menu” section in [Chapter 1, “Introduction to IC Validator VUE.”](#)

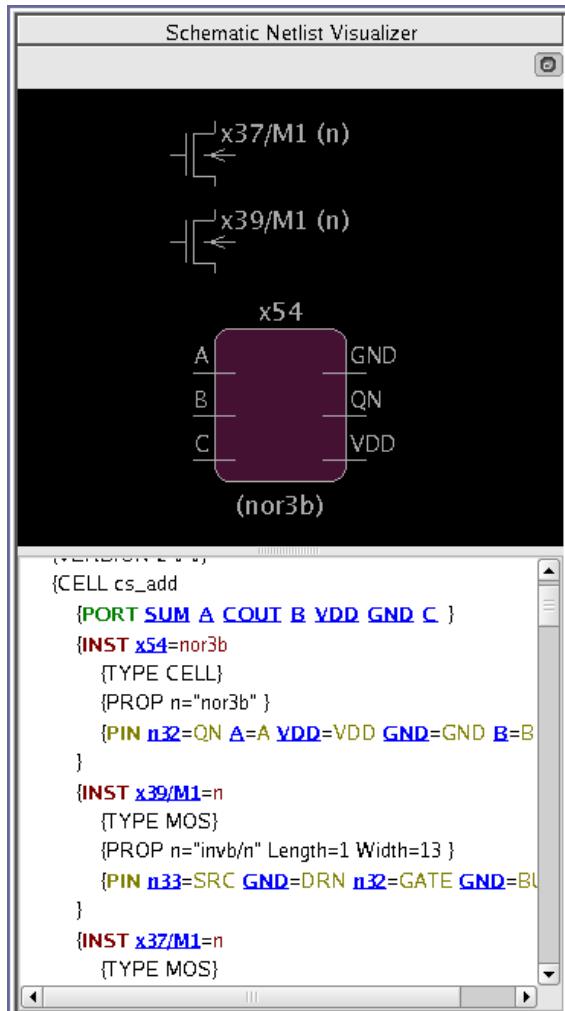
## Rendering Objects in the Graphical Canvas

The canvas displays objects from the netlist using standard engineering symbols. This section describes how these symbols are drawn and how to interact with them.

### Devices

The Netlist Visualizer interface renders primitive devices using standard engineering symbols. Furthermore, it renders Child cells or Gendev devices as rectangles. Instances are labeled with the instance and model name, shown in [Figure 3-46](#).

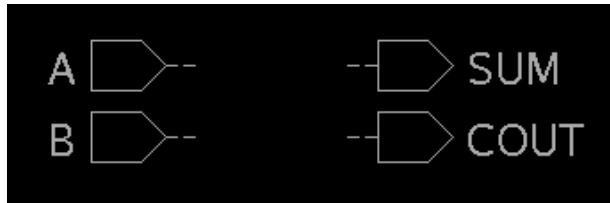
*Figure 3-46 Rendering a Device*



## Ports

Ports are drawn according to their direction. For example, input ports are drawn on the left of the canvas, and output ports are drawn on the right of the canvas, shown in [Figure 3-47](#).

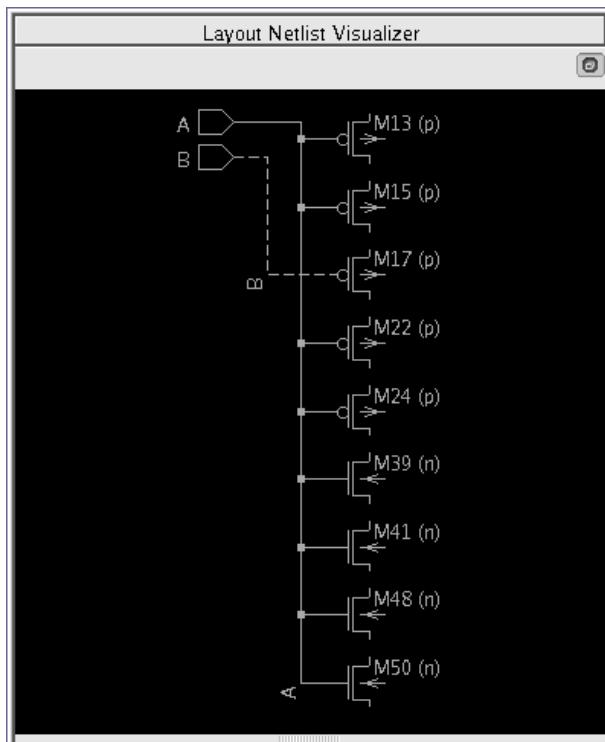
*Figure 3-47 Drawing a Port*



## Nets

Nets are drawn as orthogonal lines with dots representing connection points. Complete nets are represented as solid lines and incomplete nets are represented as dashed lines, shown in [Figure 3-48](#).

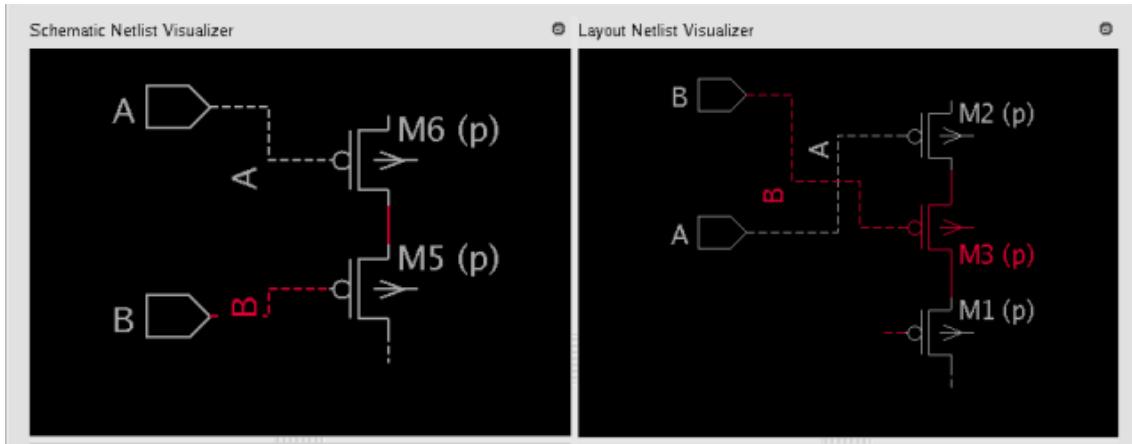
*Figure 3-48 Drawing a Net*



## Rendered Color for Matched and Unmatched Objects

Matched devices and nets are rendered in the canvas in gray, shown in [Figure 3-49](#).  
 Unmatched devices and nets are rendered in red.

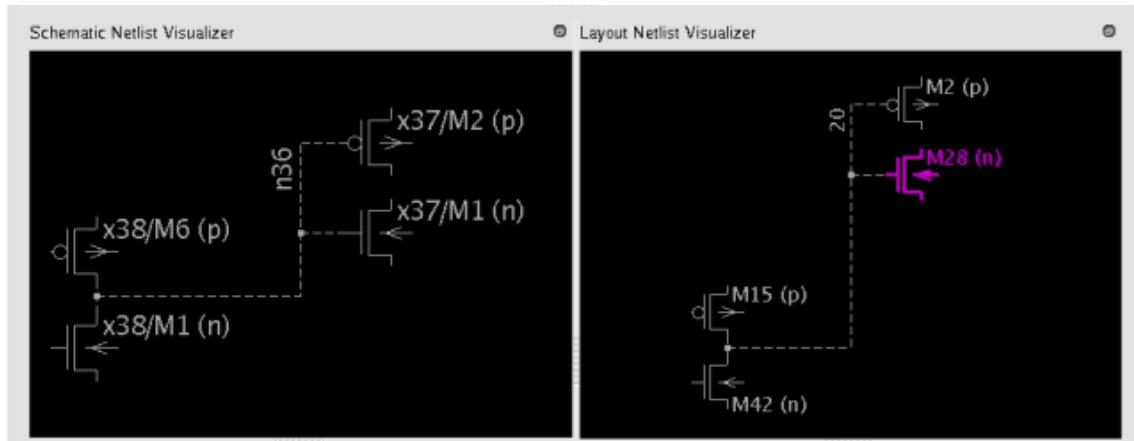
*Figure 3-49 Unmatched Devices and Nets*



## Selected Objects

You can select an object by clicking it. Selected objects are displayed in magenta and slightly bold. Selection is not synchronized between the two canvases, shown in [Figure 3-50](#).

*Figure 3-50 Selecting Objects*



You can select multiple objects in one of several ways. You Shift-click objects to add them one at a time to the selection set. You can press bindkey “A” before each left-click to add objects one at a time to the selection set, and you can click or drag to draw a box in the canvas and add all of the objects in the box to the selection set. For more information, see [Table 3-6](#) and [Table 3-7](#) on page 3-58.

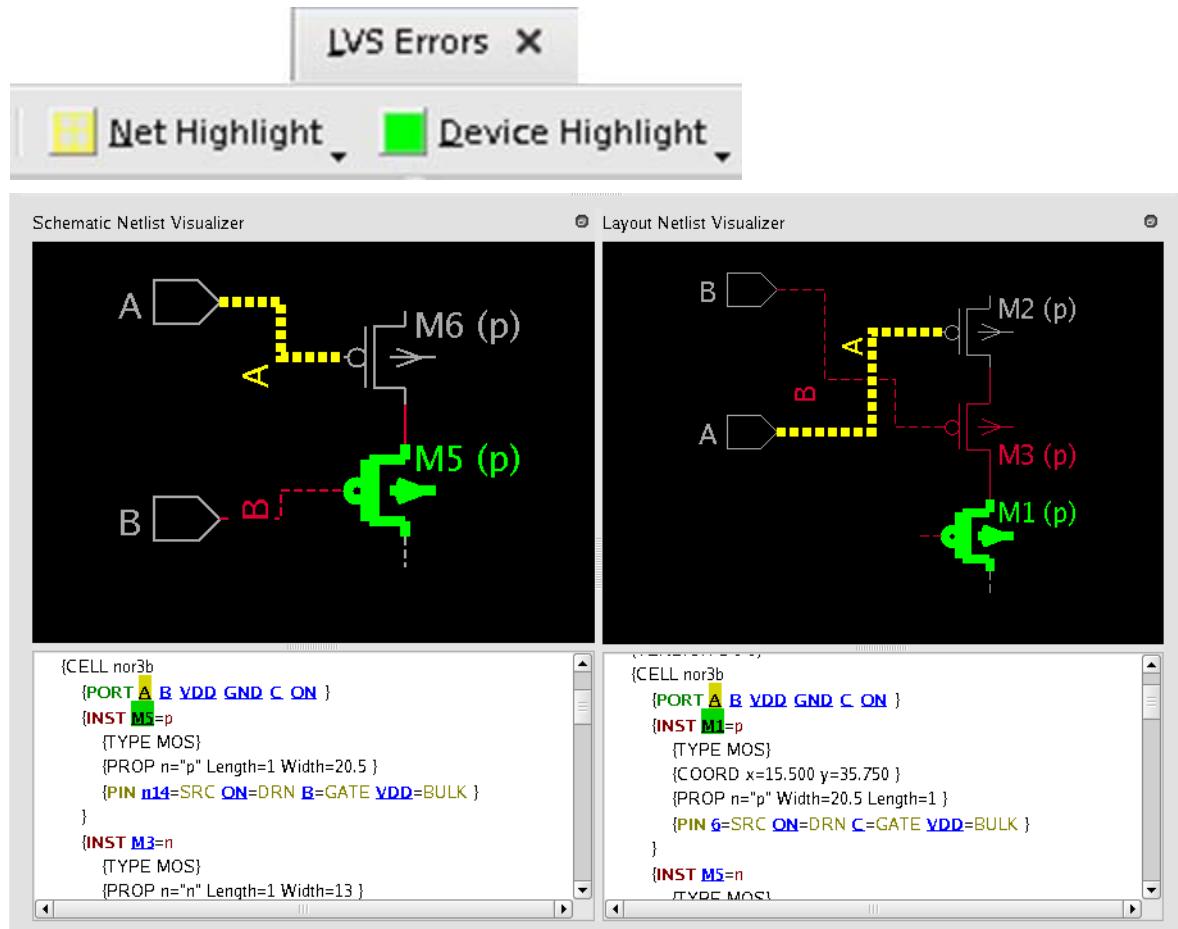
After you select the objects, a pop-up menu is available for actions on the selected objects, described in [Table 3-4 on page 3-57](#).

## Highlighted Objects

To highlight an object, right-click the object and choose Highlight. In the canvas and the text netlist, highlighted objects are differentiated by their colors. As shown in [Figure 3-51](#) and [Figure 3-52](#), you can choose the highlight colors for devices and nets by using the Net Highlight and Device Highlight menus on the Netlist toolbar. To highlight multiple nets and devices, select the Highlight Multiple Objects option, shown in [Figure 3-35](#).

In [Figure 3-51](#), the nets are highlighted in yellow and the devices are highlighted in green in the canvas and text netlist.

*Figure 3-51 Highlighted Objects*



In [Figure 3-52](#), the nets are highlighted in magenta and the devices are highlighted in red in the canvas and text netlist.

Figure 3-52 Highlighted Objects

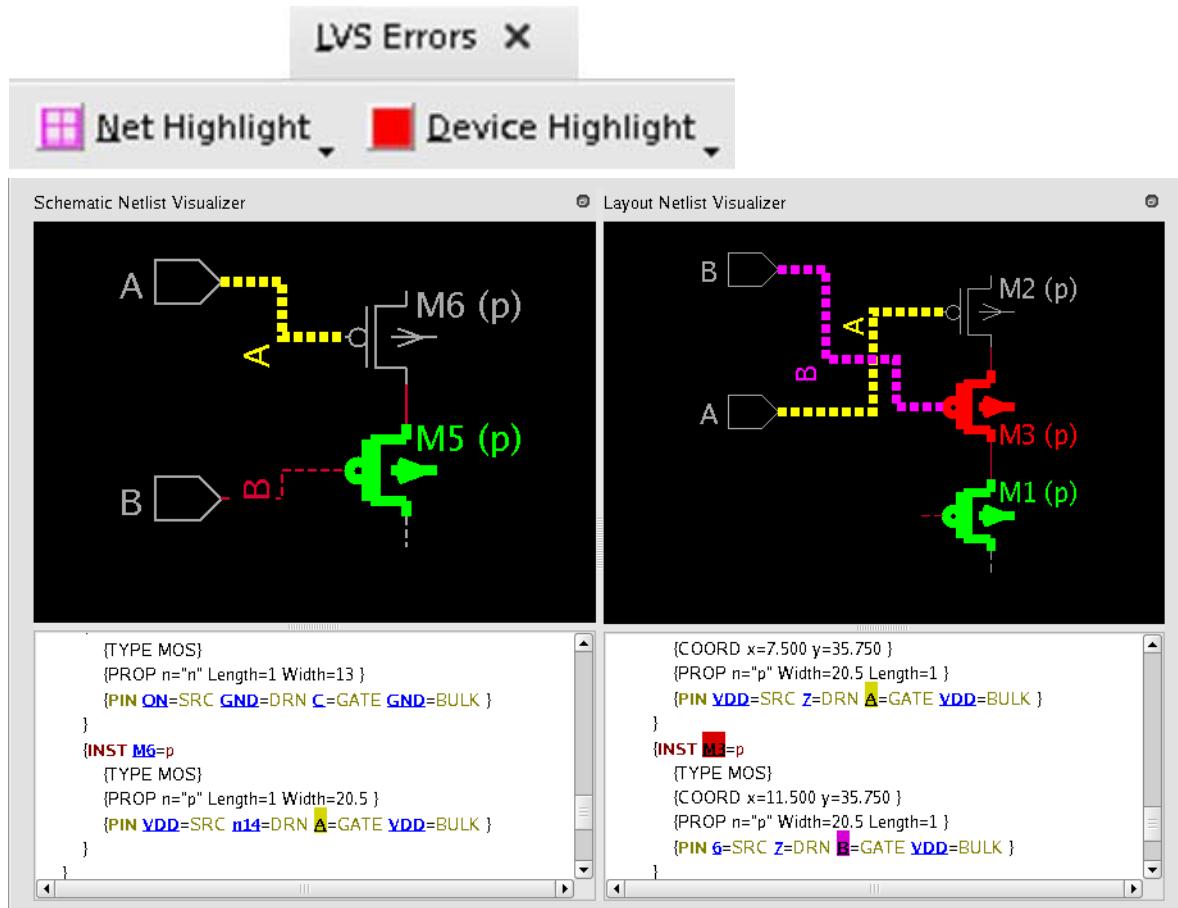
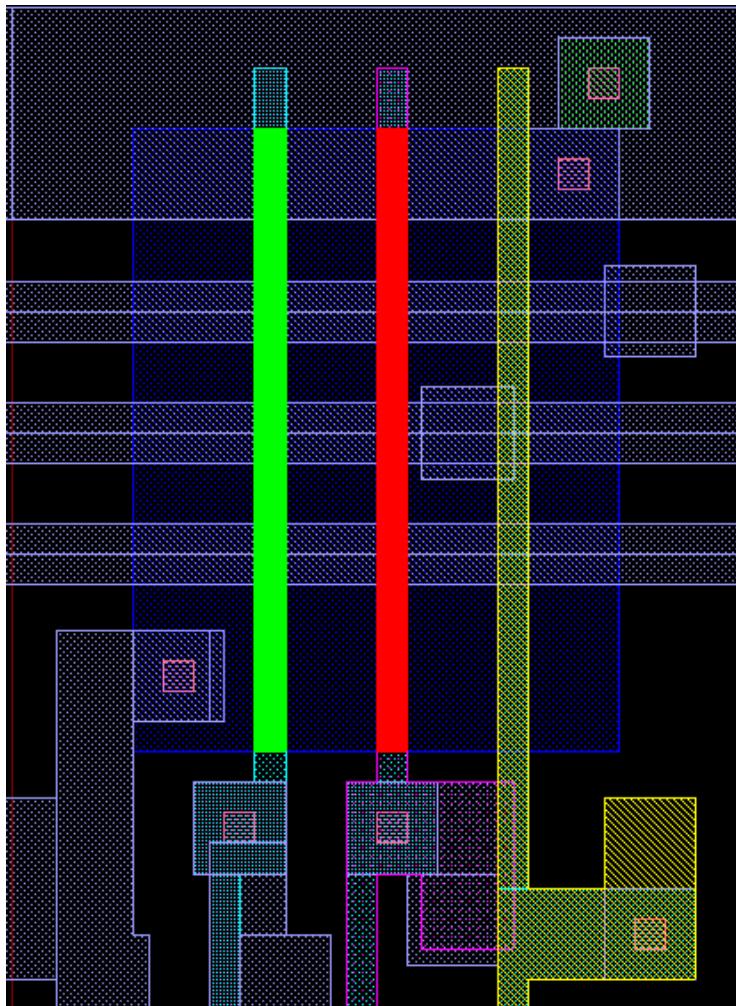


Figure 3-53 shows the layout window with multiple nets and devices highlighted.

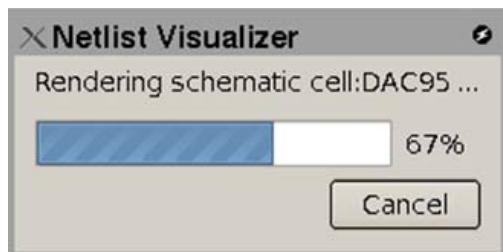
Figure 3-53 Multiple Nets and Devices Highlighted



## Progress Bar

The progress bar, shown in [Figure 3-54](#), provides information about the progress of the drawing task.

Figure 3-54 Progress Bar



## Adding Objects to the Graphical Canvas

There are three ways to add objects to the canvas: from the text window, from the canvas, and from the equivalence list. These are discussed in the following sections.

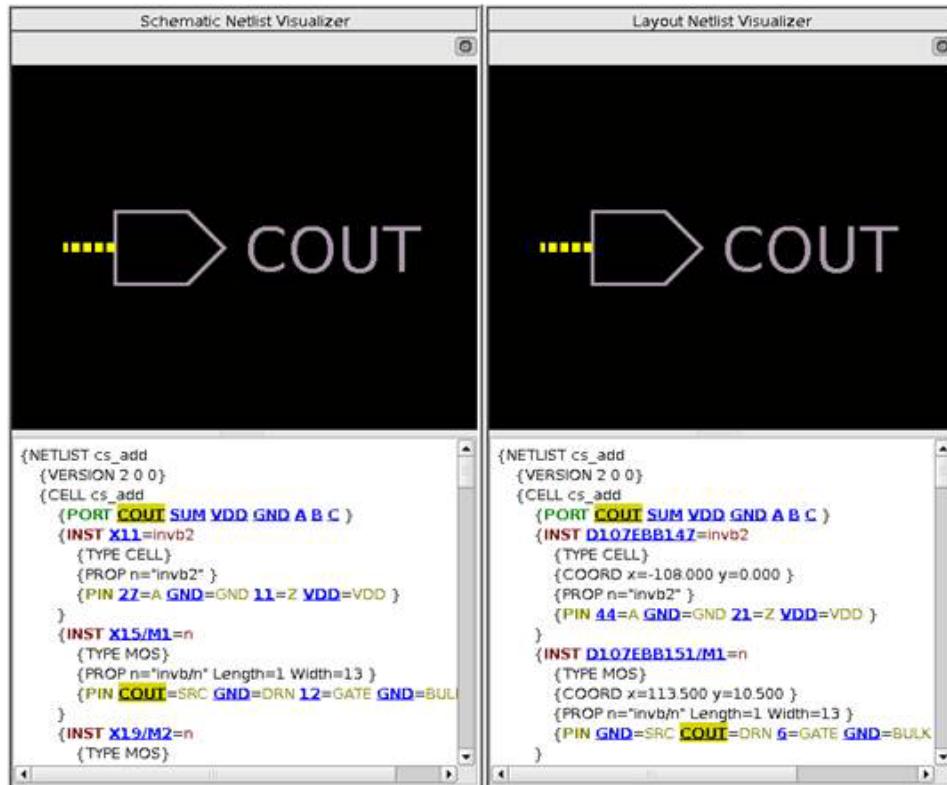
### Adding Objects From the Text Window

This section outlines how to add content to the graphical viewer by interacting with the text netlist.

#### Adding a Port

When you draw ports, they are rendered according to their direction. In addition to two different symbols for input and output ports, the input ports are drawn on the left of the canvas and the output ports on the right, shown in [Figure 3-55](#).

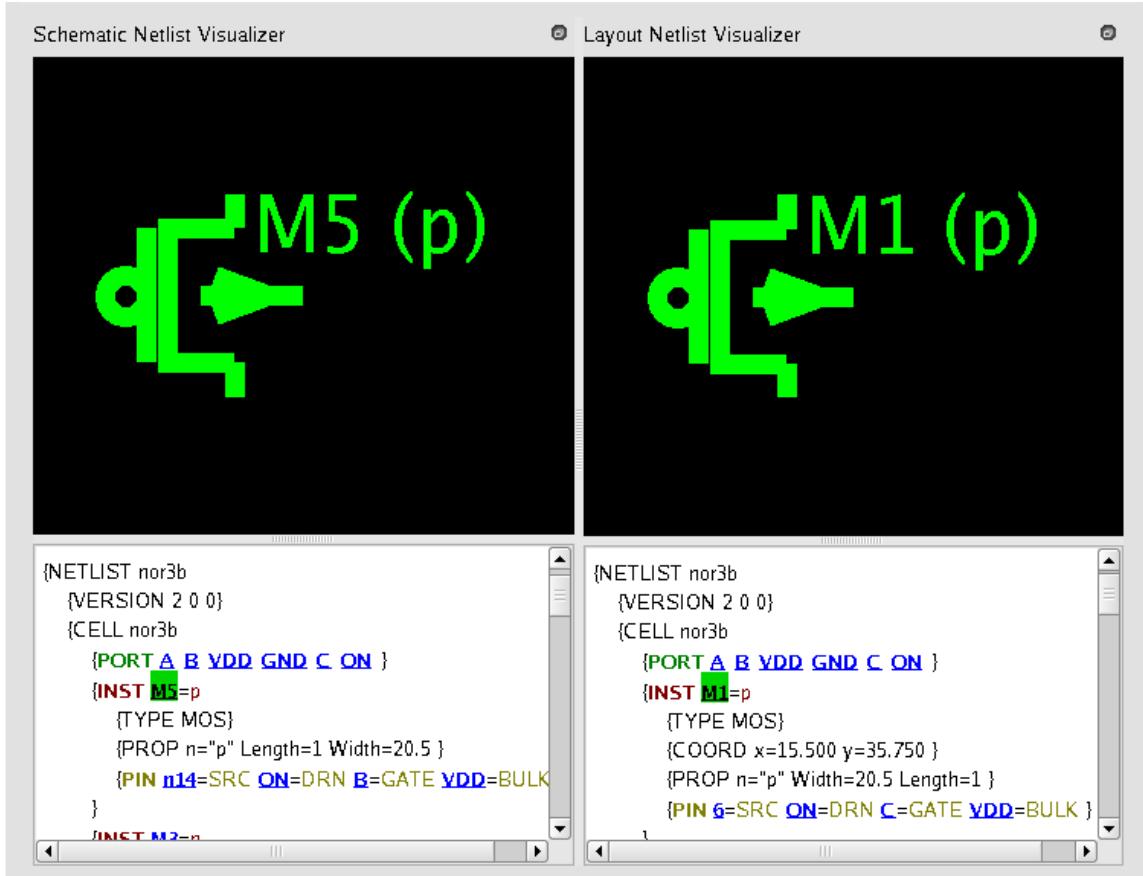
*Figure 3-55 Adding a Port*



## Adding a Device

When you select a device, it is added to the canvas and highlighted. The highlight color green is determined by the selection in the Device Highlight menu in the Netlist toolbar. A selected matched device is drawn in both canvases, shown in [Figure 3-56](#).

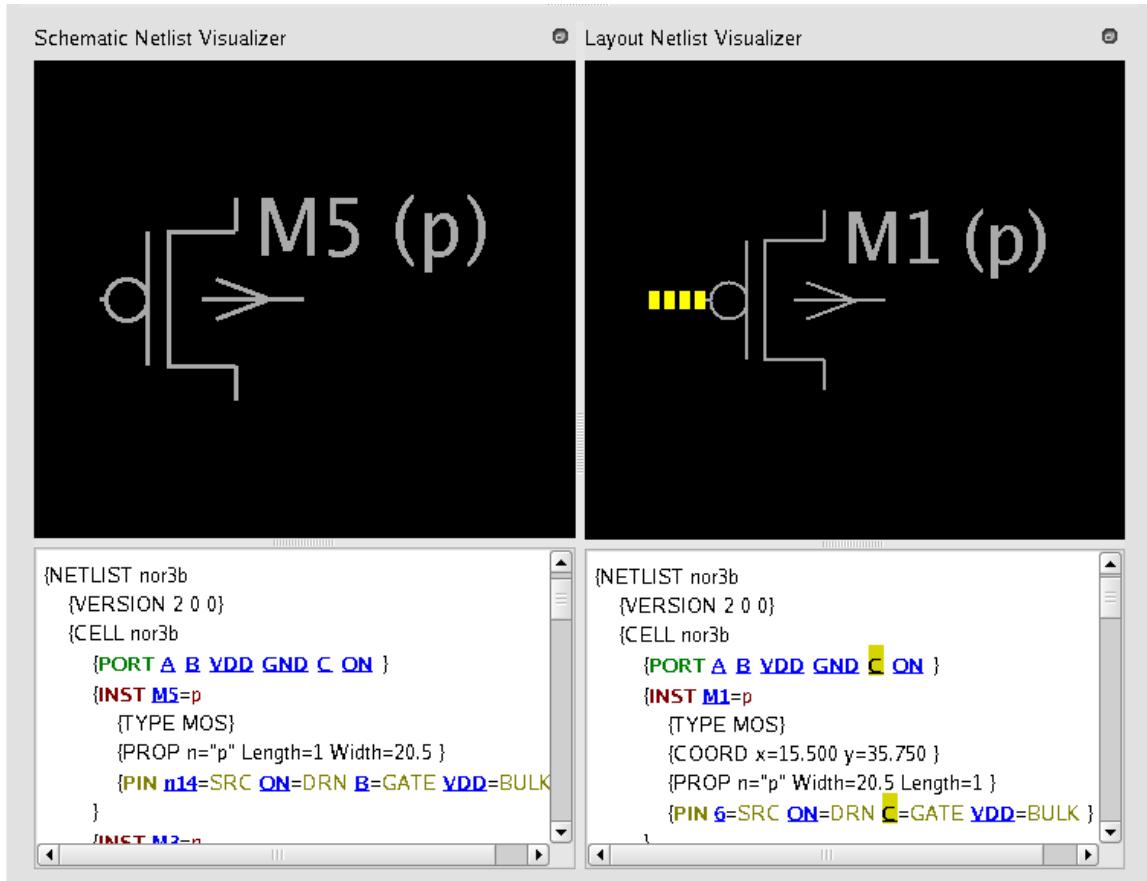
*Figure 3-56 Adding a Device*



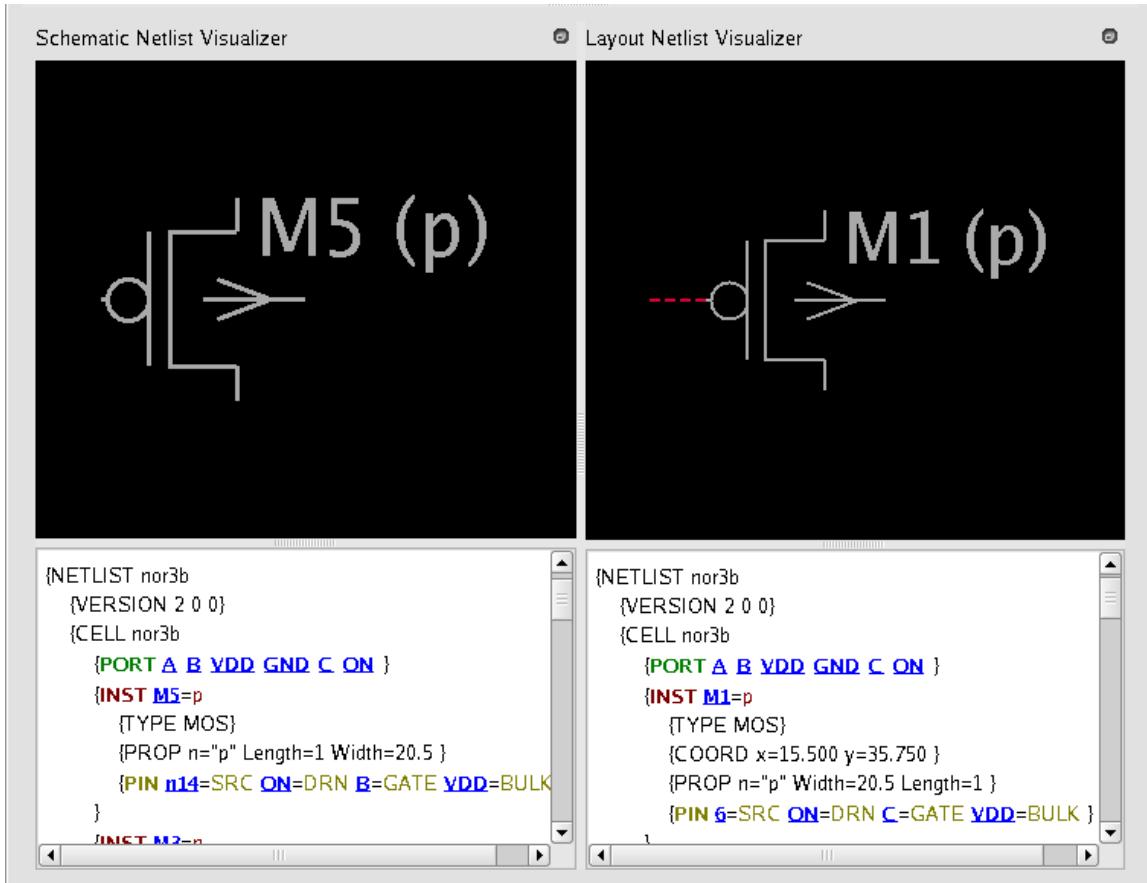
## Adding a Device Pin

You can click pins of cells and devices in the text netlist. These pins are not represented without also drawing the parent object, shown in [Figure 3-57](#).

Figure 3-57 Adding a Device Pin



Select the pin of a device to add the device and the net connected to the pin. Then, highlight the pin. This net is colored yellow on the layout side and undrawn on the schematic side, shown in [Figure 3-57](#). The yellow highlight color is determined by the Net Highlight menu, and the net is undrawn on the schematic side because it is an unmatched net.

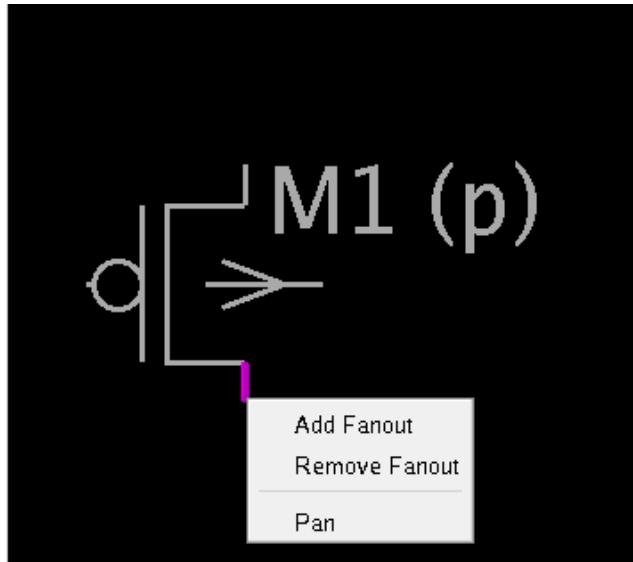
*Figure 3-58 Unmatched Pin*

By clearing the yellow highlight, shown in [Figure 3-58](#), you can see that the rendered partial net connected to the pin that was originally highlighted is shown in red because it is unmatched.

## Adding Objects From the Graphical Canvas

After objects are drawn in the canvas, you can add more connected objects by using the Add Fanout feature, shown in [Figure 3-59](#).

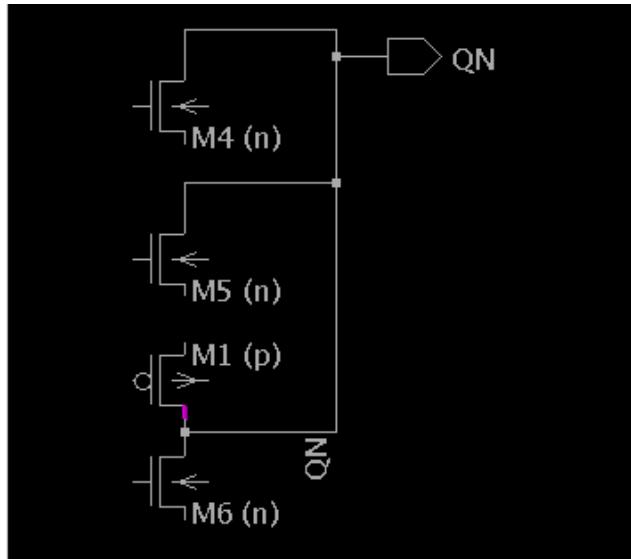
Figure 3-59 Add Fanout Selection



After selecting a pin, net, or port, you can right-click and choose Add Fanout, shown in [Figure 3-60](#). This causes all of the objects connected to the current object to be added to the canvas.

It is also possible to right-click the canvas and choose Load All Objects to draw all of the objects in the current equivalence list. See the “[Pop-Up Menu](#)” section for more information about the functions you can select.

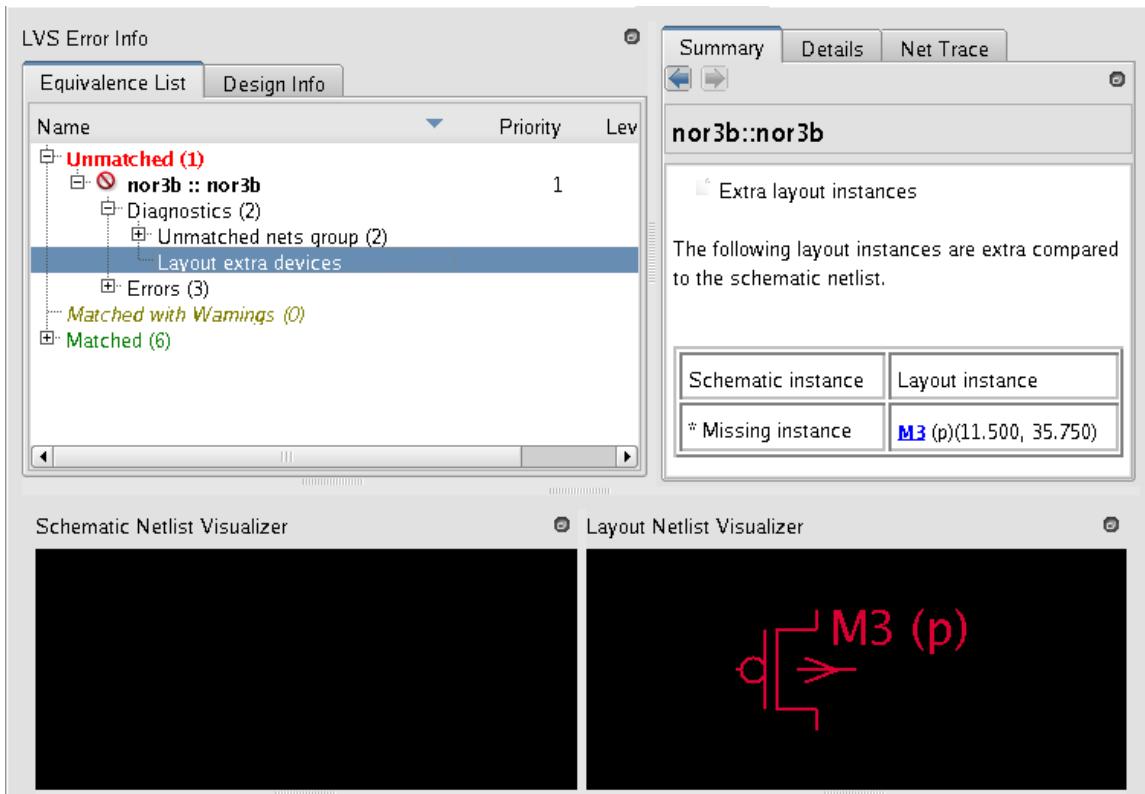
Figure 3-60 Result of Adding Fanout



## Adding Objects From the Equivalence List

After you select a diagnostic from the equivalence list, objects are added to the canvas that are appropriate for debugging that particular diagnostic. The objects are added incrementally to the canvas. Note, however, that not all diagnostics are supported.

Figure 3-61 *Equivalence List Diagnostics*



As an example, Figure 3-61 shows the Layout extra devices Diagnostic. When this diagnostic is selected in the equivalence list, the extra instance present in the layout but not in the schematic are added to the Layout Netlist Visualizer canvas.

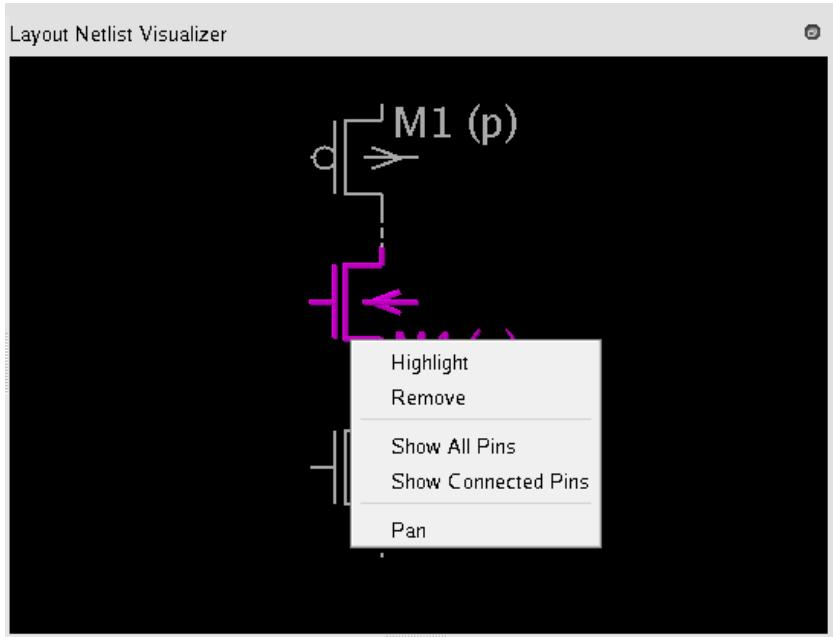
## Interacting With Objects in the Graphical Canvas

After objects are drawn in the canvas, you can interact with them. Although you cannot change the placement of any of the objects, it is possible to add more objects to the canvas, remove objects from the canvas, and change the color and style of the objects drawn in the canvas.

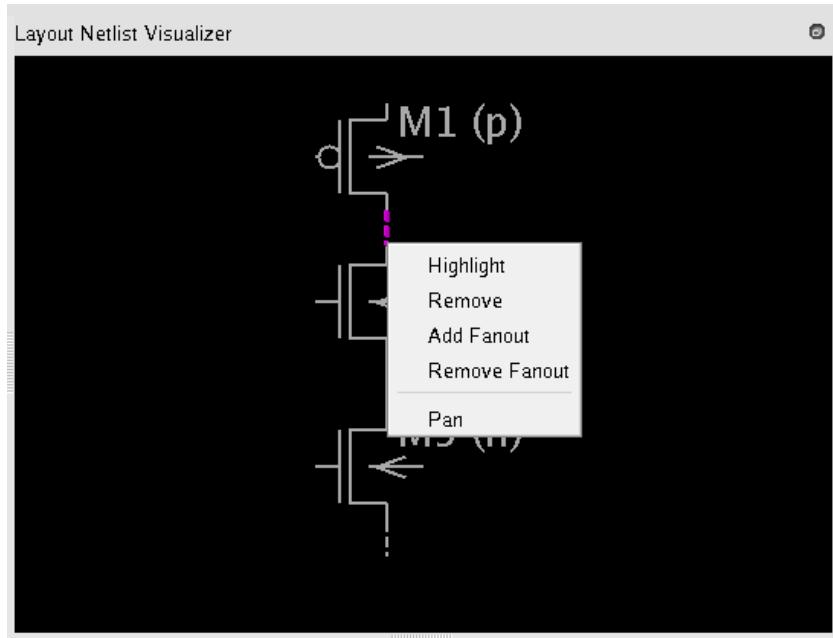
## Pop-Up Menu

When you right-click a selected object, a pop-up menu is displayed. The functions of the menu depend on the objects you select, shown in [Figure 3-62](#) and [Figure 3-63](#).

*Figure 3-62 Menu Selections*



*Figure 3-63 Alternative Menu Selections*



**Table 3-4** shows the commands you can choose to change your design.

*Table 3-4 Pop-Up Menu Selections*

| Command            | Description  | Applies to                            |
|--------------------|--|---------------------------------------|
| Add All Contents   | Add all content of this child cell to the canvas.  | Cell                                  |
| Add Fanout         | Add all objects that are connected to this object.   | Port, Net, Instance Pin, Cell Pin     |
| Fold/Unfold        | Switch between displaying or not displaying the content of this cell.  | Cell                                  |
| Highlight          | Highlight this object on both canvases and in the text window.   | Net, Instance, Cell                   |
| Load All Objects   | Draws all objects of the current equivalence in the canvas.  | Only visible when nothing is selected |
| Pan                | Put the canvas into Pan mode, during which a left click pans the canvas.   | Always present                        |
| Remove             | Remove this object from the canvas.  | Port, Net, Instance, Cell             |
| Remove All Objects | Remove all objects from the canvas. This leaves the canvas blank.  | Always present                        |
| Remove Fanout      | Remove all objects that are connected to the current object. If the net to be removed is power or ground, remove only the net. | Port, Net, Instance Pin, Cell Pin     |

## Graphical Canvas Navigation

**Table 3-5** describes the commands you can use to zoom and pan.

*Table 3-5 Canvas Navigation*

| Command | Mouse action                                    |
|---------|---|
| Pan     | Middle-click or drag.                           |
| Pan     | Right-click and choose Pan, then click or drag. |

## Bindkey Settings for Netlist Visualizer

[Table 3-6](#) lists the bindkey settings for the mouse.

*Table 3-6 Bindkey Settings for the Mouse*

|              | <b>Left button</b>   | <b>Middle button</b> | <b>Scroll wheel</b> | <b>Scroll wheel</b> |
|--------------|----------------------|----------------------|---------------------|---------------------|
|              | Select by point      |                      | Zoom in             | Zoom out            |
| Shift        | Add select by point  |                      | Pan north           | Pan south           |
| Ctrl         | Deselect by point    |                      | Pan west            | Pan east            |
| Dbl          | Fold/Unfold          |                      |                     |                     |
| Drag         | Select by window     | Pan                  |                     |                     |
| Shift + Drag | Add select by window |                      |                     |                     |
| Ctrl + Drag  | Deselect by window   |                      |                     |                     |

[Table 3-7](#) lists the bindkey settings for the keyboard.

*Table 3-7 Bindkey Settings for the Keyboard*

| ↑         | ↓         | ←        | →        | A                   | D                     | Z                         | F   |
|-----------|-----------|----------|----------|---------------------|-----------------------|---------------------------|-----|
| Pan North | Pan South | Pan West | Pan East | Select by point     | Deselect by point     | Zoom in and out rectangle | Fit |
| Shift     |           |          |          | Select by rectangle | Deselect by rectangle | Zoom out                  |     |
| Ctrl      |           |          |          | Select all          | Deselect              | Zoom in                   |     |

---

## Layer Debugger

The VUE layer debugger provides a simple method to visually debug a runset. It displays all of the intermediate layers created during the run to guide you to the source of the problems.

This can be very useful for debugging:

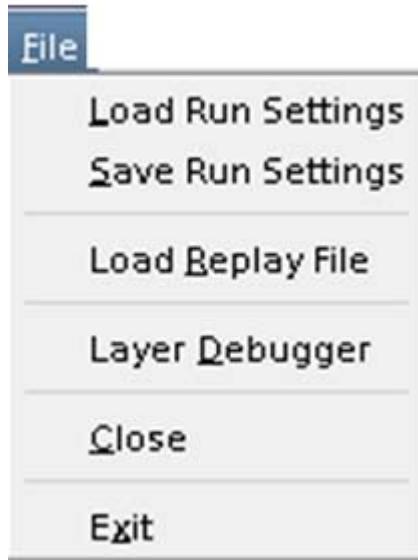
- Problems where a complex runset algorithm used to define a rule is not returning the intended results.
- Problems in the custom layout where the runset is correct but the layout is not correct.

---

## Setup

To run the layer debugger, choose the Layer Debugger command on the File menu, shown in [Figure 3-64](#).

*Figure 3-64 Layer Debugger Command on the File Menu*



After you choose this command from the menu, the advanced execution tabs are updated to the layer debugger mode.

There are two ways in which to generate and debug data using the Layer Debugger. You can run the IC Validator tool from the command line using the `-layer_debugger` command-line option, or you can run from within VUE. The results can be loaded or propagated in VUE for debugging.

To run the Layer Debugger using the `-layer_debugger` IC Validator command-line option,

1. Run `icv -layer_debugger ...`
2. Open VUE.
3. Choose File > Layer Debugger.
4. From the Execution tab, load the runset by clicking the Load Runset button.

5. Click the Load Results tab.

To run the Layer Debugger from within VUE,

1. Open VUE.
2. Choose File > Layer Debugger.
3. From the Execution tab, load the runset by clicking the Load Runset button. See the [“Layer Debugger Execution Options” section](#) for more information.
4. From the Execution tab, click Execute.

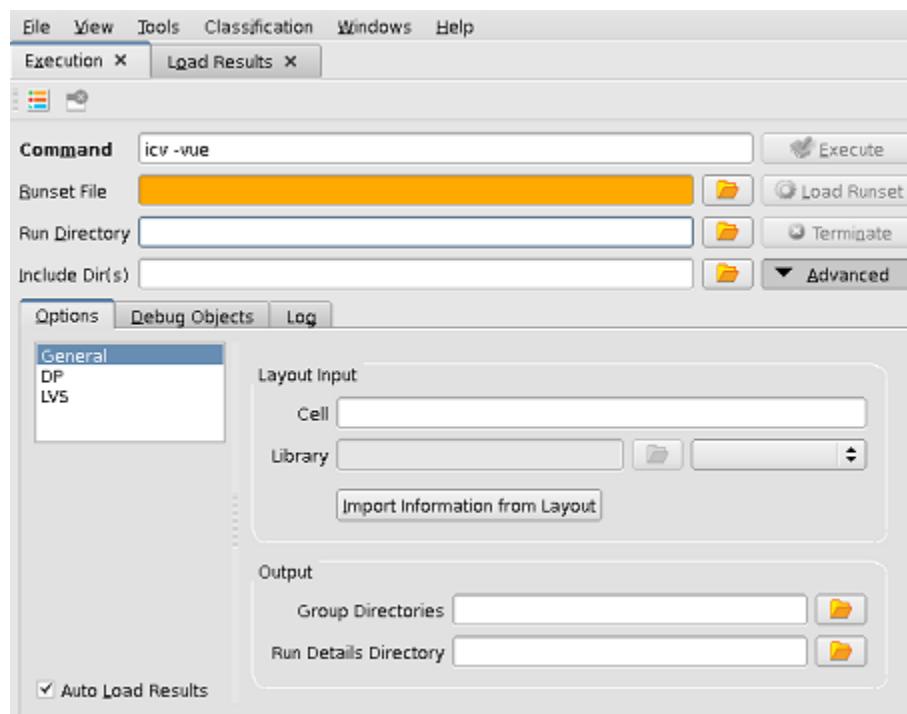
With the Layer Debugger activated, all of the commands in your runset are executed.

---

## Layer Debugger Execution Options

The Options tab is simplified to avoid function overlap with the layer debugger, and it removes controls that are not needed for a layer debugger run, shown in [Figure 3-65](#).

*Figure 3-65 Options Tab*



During the run, the layer debugger turns off most of the runset optimizations. As a result, all of the user-specified commands are executed as written.

---

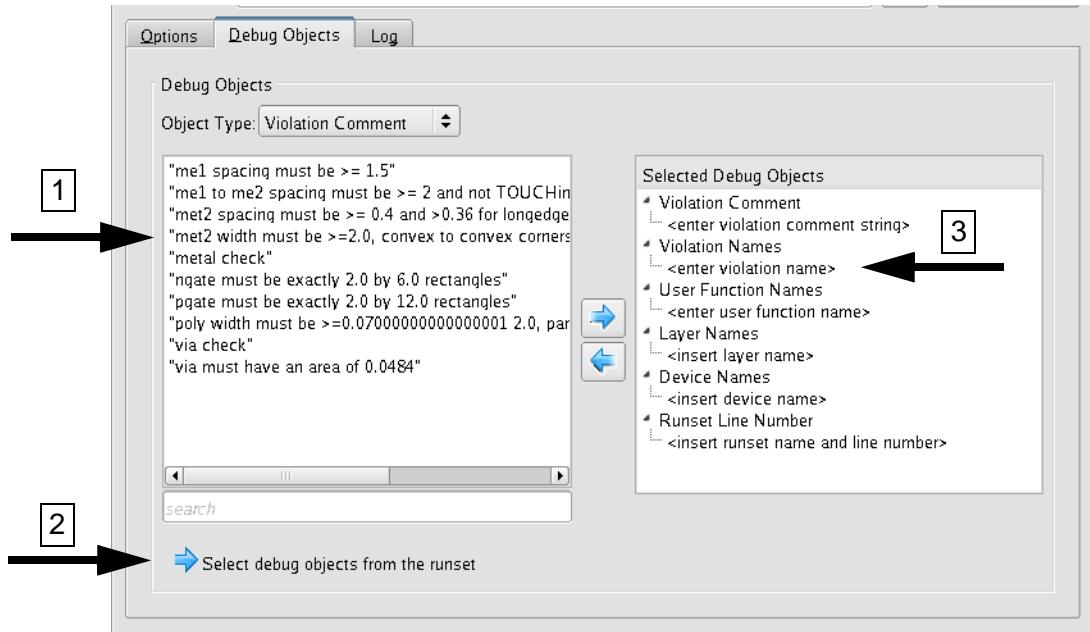
## Layer Debugger With Selected Debug Objects

A debug object is the endpoint of the runset branch being debugged. In the layer debugger, only the runset commands that lead to the specified debug objects are executed. No other runset branches, including creating output libraries, are executed. Multiple debug objects can be selected for a layer debugging run. To isolate the problem, select only the debug objects needed, as described in the following paragraph. If no debug objects are selected, all commands in the runset are executed.

To run the Layer Debugger in VUE with the selected debug objects,

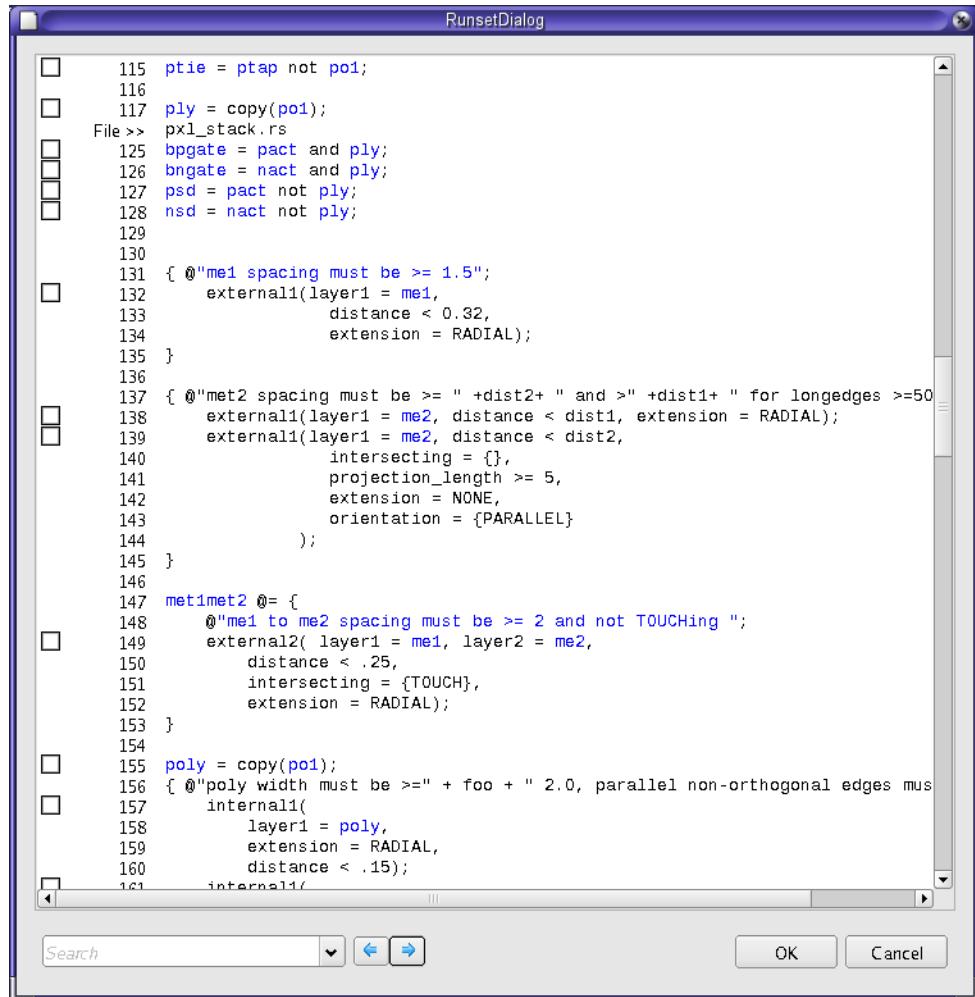
1. Open VUE.
2. Choose File > Layer Debugger.
3. From the Execution tab, locate a runset in the Runset File text box and click the Load Results button. The Debug Objects window is split into two main panes. The left pane shows the available debug objects in the runset. The right pane shows the selected list of debug objects.
4. To select a debug object, you can
5. Select objects from the available debug objects pane and move them to the selected pane using the Right Arrow button between the two panes.
6. Select debug objects from within the runset view, shown in [Figure 3-66](#).
7. Manually enter a debug object into this list. This feature is disabled and the value is ignored.
8. From the Execution tab, click Execute.

Figure 3-66 Debugging Objects



The Runset dialog box, shown in [Figure 3-67](#), is the entire runset consolidated into one file. All of the include file information is included here. The check boxes next to a runset line indicate to the debugger that the command on that line is a debug object. If the output layer on that line is defined multiple times after this command, those instances are not run.

Figure 3-67 Runset Dialog Box



There are five types of debug objects.

- Violations Comment

All commands that write specific violation comments can be selected for debugging by selecting the comment as a debug object. For example, select

```
{@"this is a violation comment"; ...}
```

To select a violation comment as a debug object, select Violation Comment in the Object Type list box, and then select the appropriate violation comments in the list. Each layer that produces a runset command by which that debugging endpoint is dependent is displayed in the layer debugger results.

- Violation Name

Commands that write specific violation names can be selected for debugging. For example, select

```
myTest @= {@"this is violation comment...}"
```

The violation name is myTest. Wildcards can be used to designate which violation comments get debugged. Any command that writes a violation comment becomes a debugging endpoint. Any layer that produces a runset command by which that debugging endpoint is dependent is added to the output.

- Layer Name

You can specify the name of the polygon layers or edge layers to be debugged. In the Debug Objects pane, each layer name is listed only one time even if there are multiple definitions in the runset in different scopes. When a layer is selected as a debug object, all definitions of the layer are commented as debug objects and are run.

- Device Name

You can specify the name of a device to be debugged. If a device is selected to be debugged, all of the layers needed to create that individual device, including any processing layers, reference layers, or recognition layers are considered as debug objects. Note that device extraction requires a connect sequence so that every layer within the connect database must be derived and shown in the output.

- Runset Line Number

You can select a debug object by the runset line number. To select a single command as a debug object, click the “select debug objects from the runset” arrow button and select the appropriate check box.

---

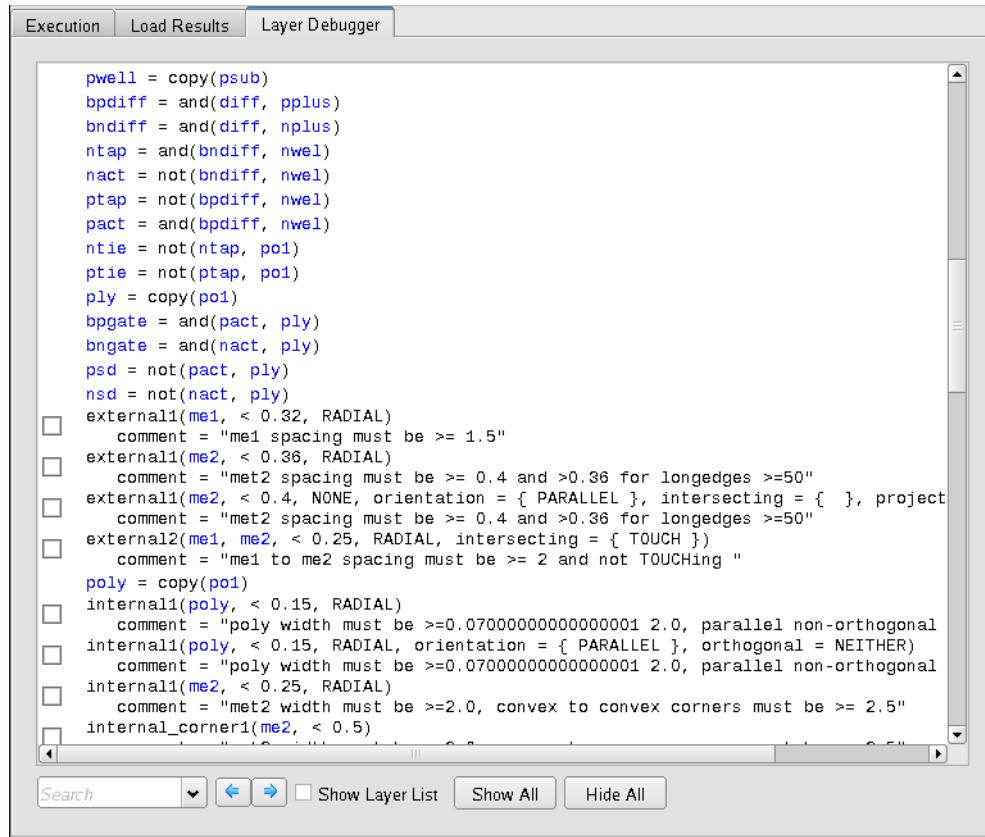
## Running the Layer Debugger

After all of the debug objects are selected, click the Execute button. The layer debugger executes the runset with the appropriate debug objects selected. After the run completes, the debugger output is loaded into a new tab.

---

## Debugging the Run

The Layer Debugger tab, shown in [Figure 3-68](#), shows each of the commands that was executed during the debug run. The layer debugger run generates an output library that contains all of the layers generated by the IC Validator tool with the final optimized hierarchy. When connected to IC WorkBench EV Plus, the layer debugger opens the output library. Click a layer in the Layer Debugger tab to turn the layer on. Click it again to turn the layer off. The entire layer is displayed in the top cell of the debugging run.

**Figure 3-68 Layer Debugger Interface**

All of the layers are shown in either blue or cyan. Empty layers are shown in cyan and non-empty layers are shown in blue, shown in [Figure 3-69](#).

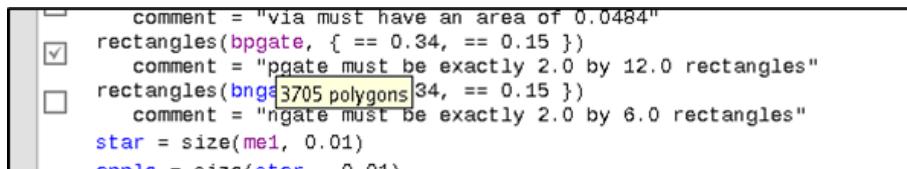
**Figure 3-69 Non-Empty and Empty Layers**

The screenshot shows the 'Layer Debugger' interface with two specific layers highlighted:

- Blue (non-empty layer):** This layer is highlighted with a blue border. An arrow points from the text 'Blue (non-empty layer)' to this highlighted section. Inside this section, the code includes: nwel\_res = interacting(nwel, wsymbol), nwres\_area = and(nwel\_res, wsymbol), nwres\_area = interacting(nwres\_area, ntie), nwres\_all = or(nwres\_bod, nwres\_bod), nw\_res\_sz = size(nwres\_all, 0.01), nwel = not(nwel, nw\_res\_sz).
- Cyan (empty layer):** This layer is highlighted with a cyan border. An arrow points from the text 'Cyan (empty layer)' to this highlighted section. Inside this section, the code includes: not(bpgate, iram), not(bngate, iram), trbpgate = and(bpgate, iram), trbngate = and(bngate, iram).

The output also has rollover help to show additional information about the layer and command run. When you hover over a layer, the total number of written polygons is displayed, shown in [Figure 3-70](#).

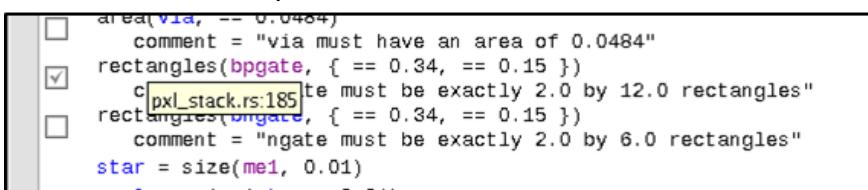
Figure 3-70 Rollover Help



```
comment = "via must have an area of 0.0484"
 rectangles(bpgate, { == 0.34, == 0.15 })
    comment = "pgate must be exactly 2.0 by 12.0 rectangles"
 rectangles(bngate, 3705 polygons, { == 0.34, == 0.15 })
    comment = "ngate must be exactly 2.0 by 6.0 rectangles"
star = size(me1, 0.01)
enable = size(star, 0.01)
```

When you hover over the command name, the full runset name and line number are shown. You can easily find the command in your runset, shown in [Figure 3-71](#).

Figure 3-71 Rollover Help



```
area(via, -- 0.0484)
    comment = "via must have an area of 0.0484"
 rectangles(bpgate, { == 0.34, == 0.15 })
    c [pxl_stack.rs:185]te must be exactly 2.0 by 12.0 rectangles
 rectangles(bngate, { == 0.34, == 0.15 })
    comment = "ngate must be exactly 2.0 by 6.0 rectangles"
star = size(me1, 0.01)
```

Violation output can also be shown together with the layer output by selecting the check box next to a violation-producing command. The comment shown on the next line in the output is the violation comment to which the violations were written.

You can search for layers, commands or violation comments in this output.

# 4

## Resolving a Short

---

*This chapter shows the ways in which you can debug the layout polygon shorts in your design.*

VUE provides two methods for debugging the layout polygon shorts in your design. They are:

- [Short Static Output in Extraction Errors Tab](#)
- [Interactive Short Finding in the Short Finder Tab](#)

---

## Short Static Output in Extraction Errors Tab

Using short static output is a simple way to list short text-pairs reports. The violations can be viewed within the VUE Extraction Errors tab.

---

### Configuration of Short Static Output

To enable short static output from the Extraction Errors tab, do one of the following:

- Specify the `-vueshort` command-line option and set the `short_debugging = {static_output = {error = true}}` argument of the `text_net()` function.
- Set the `short_debugging = {static_output = {error=true}}` and `create_short_finder_nets = {{{{"*"}, {"*"}}}}` arguments of the `text_net()` function for all nets.

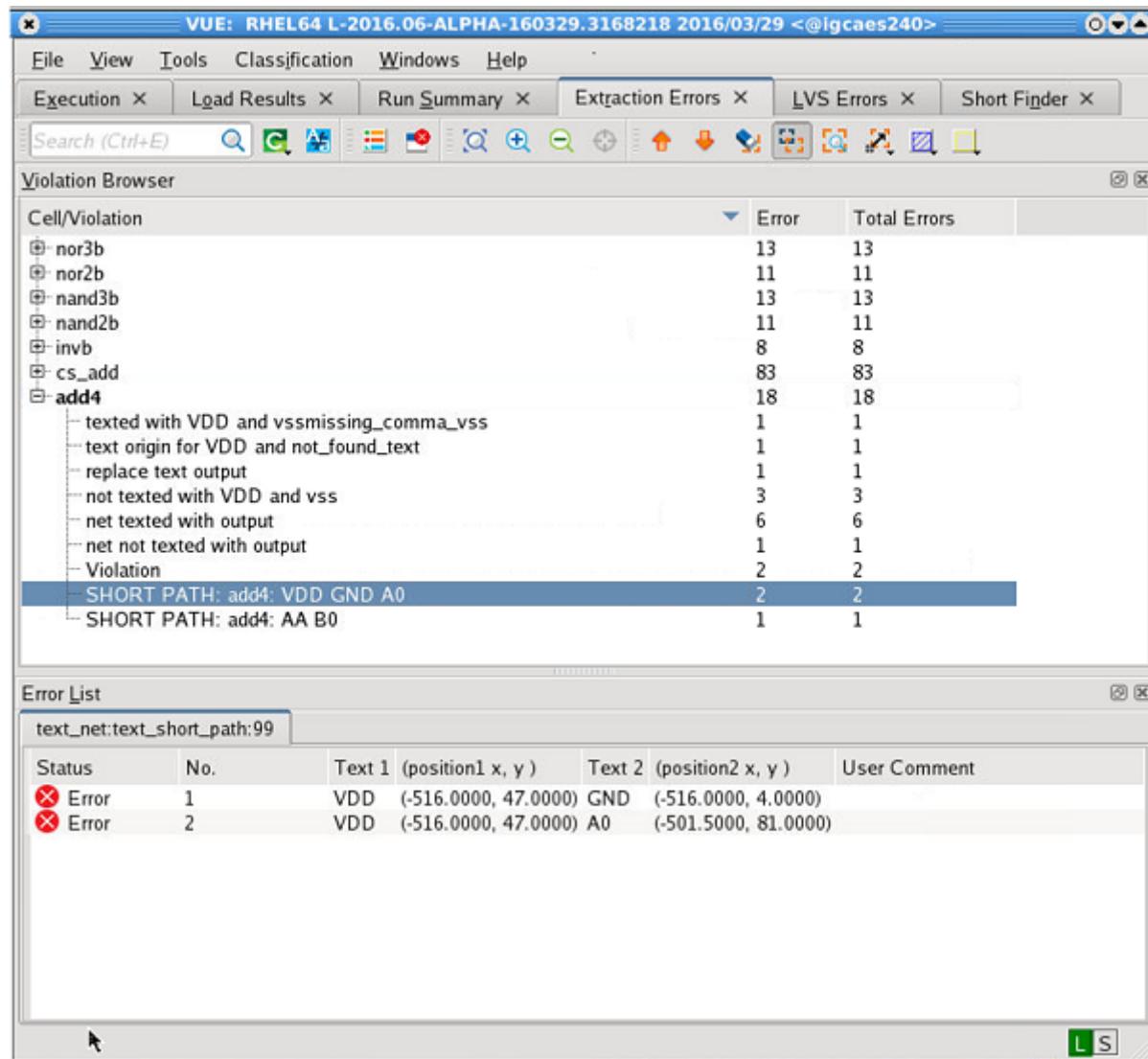
If a short is found using either of these settings, VUE results are loaded and the short static output is displayed in the Extraction Errors tab.

---

### Loading Short Static Output

- [Figure 4-1](#) shows one loaded violation. The highlighted items are available from the Error list.
  - The short path, SHORT PATH: add4: VDD GND A0, is selected in the Violation Browser pane.
  - The short text-pairs, VDD-GND and VDD-A0, text coordinates are listed in the Error List pane.

Figure 4-1 Short Static Output in the Extraction Errors tab



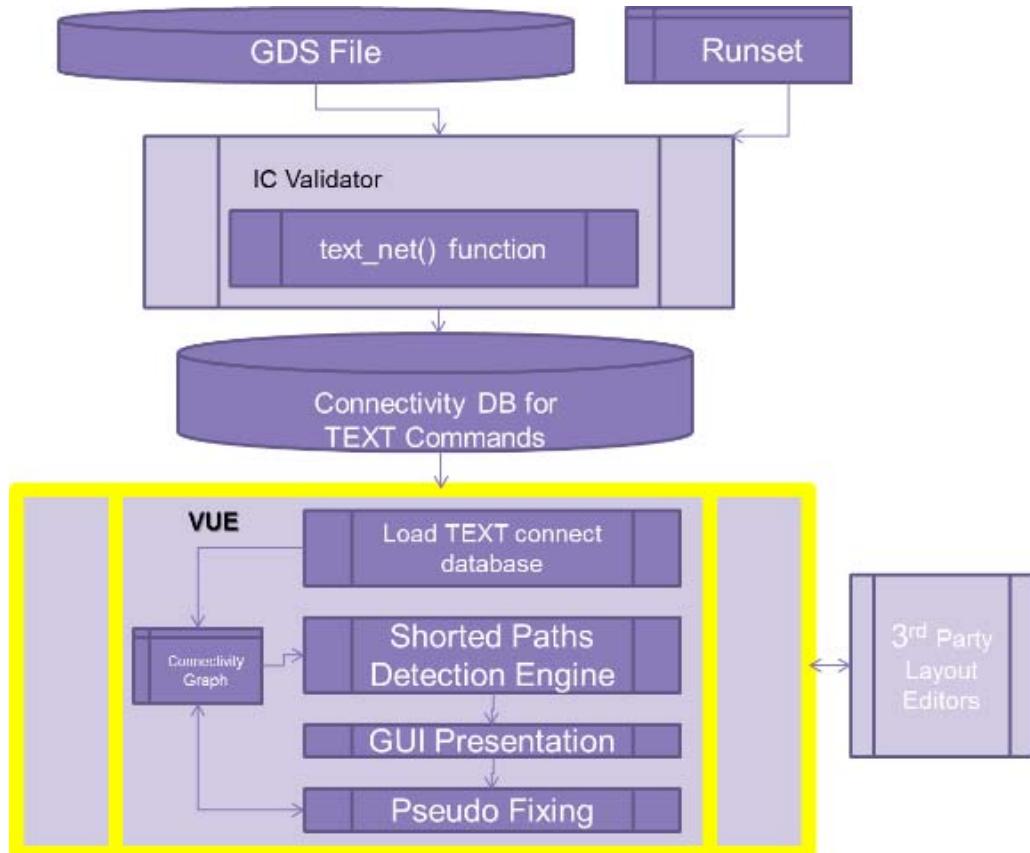
## Interactive Short Finding in the Short Finder Tab

The Short Finder tab allows you to interactively identify the root cause of full-path text using text-based and LVS-based short sources from the layout text extraction and LVS results respectively.

## Text-Based Short Finder

The text-based Short Finder allows you to easily identify the root cause of text shorts discovered during an IC Validator run. See [Figure 4-2](#) for details on the text-based Short Finder flow.

*Figure 4-2 Text-Based Short Finder Flow*

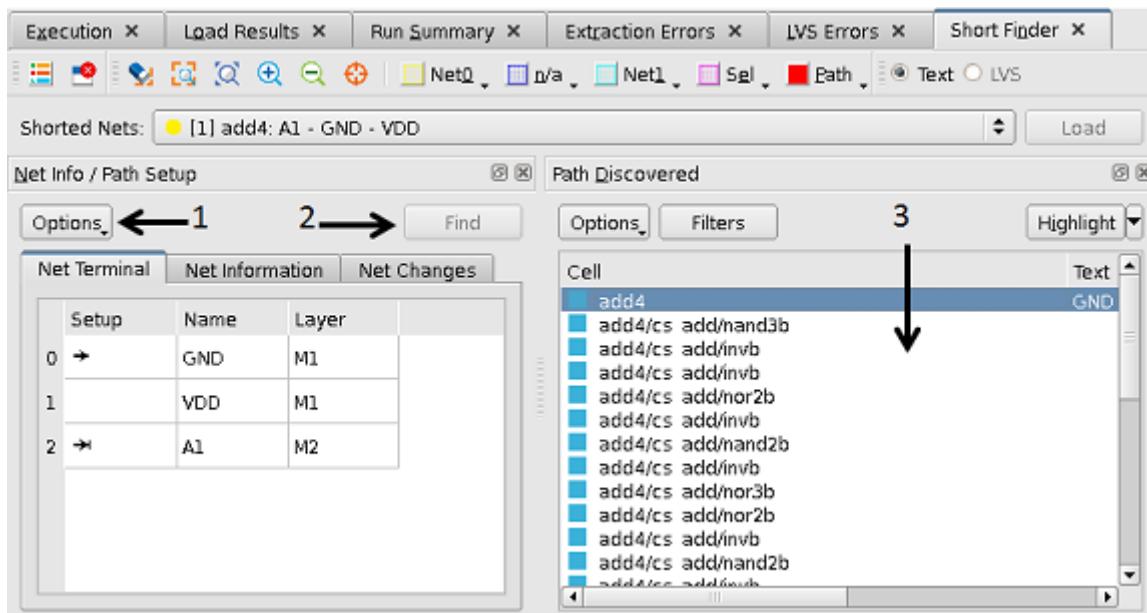


A number of features within the Short Finder tab allow you to trace the path of a reported short, label polygons according to your knowledge of the connectivity, and find multiple shorting paths between the two texts that are shorted. Polygons can be highlighted and zoomed, and you can use different colors to show pruning of polygons that are properly connected. There are two basic steps, shown in [Figure 4-3](#).

1. Verify that Auto-Selection is selected in the Options list box.
2. Click Find in the Net Info / Path Setup pane.

The complete short path, including Cell, Text, Net, and Layer, are then displayed in the Path Discovered pane.

Figure 4-3 Path Discovered Pane

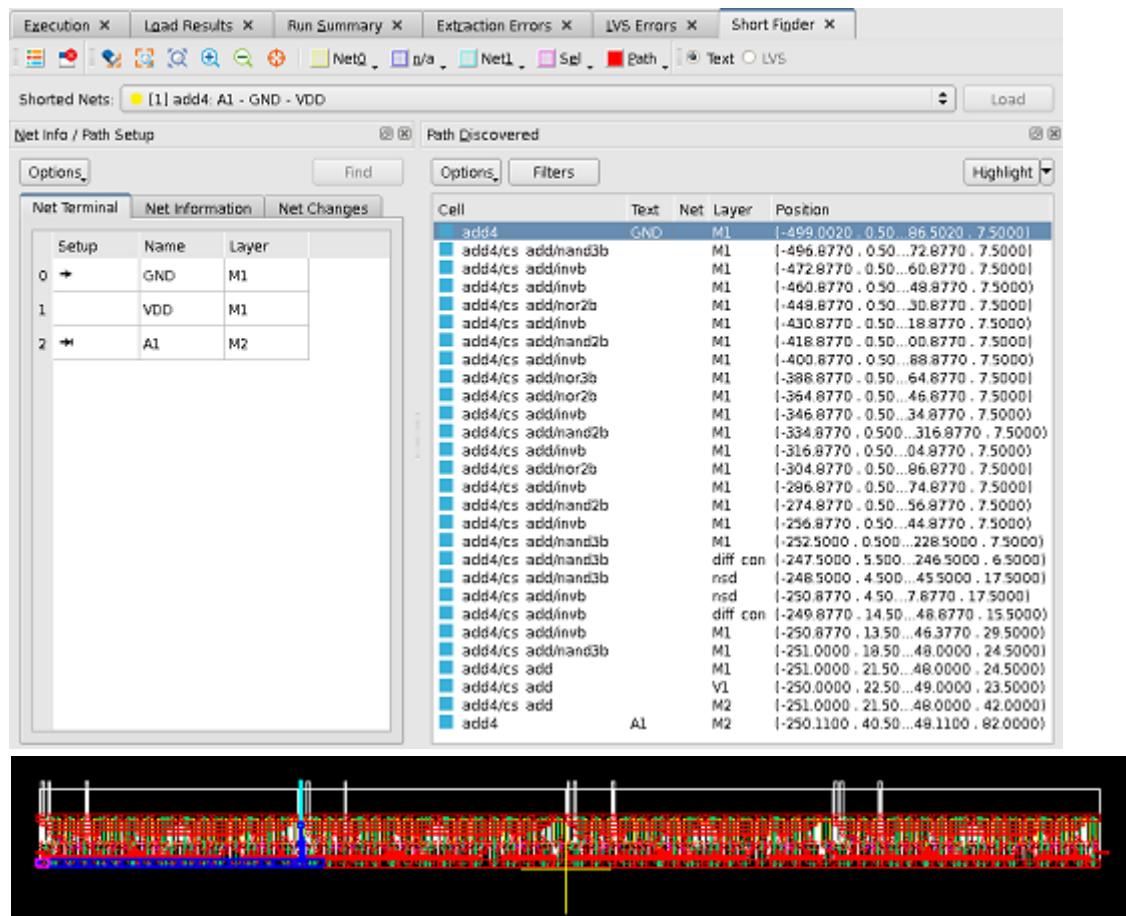


To see the short path, you can

- Select Highlight All in the Options list box. Click the Highlight button to highlight the entire short path in the layout editor connected to VUE.
- Click the polygons listed in the Path Discovered pane. Click the Highlight button to see the location of the polygons in the whole short path.

The text-based Short Finder tab, shown in [Figure 4-4](#), provides an overview of the whole short path, including the hierarchical shorts.

Figure 4-4 Text-Based Short Finder



## Configuration of the Text-Based Short Finder

To enable the text-based short output from the Short Finder tab, do one of the following:

- Specify the `-vueshort` command-line option.
- Set the `short_debugging = {vue_short_finder = true}` and `create_short_finder_nets = {{{}*}, {"*}}}` arguments of the `text_net()` function for all nets.

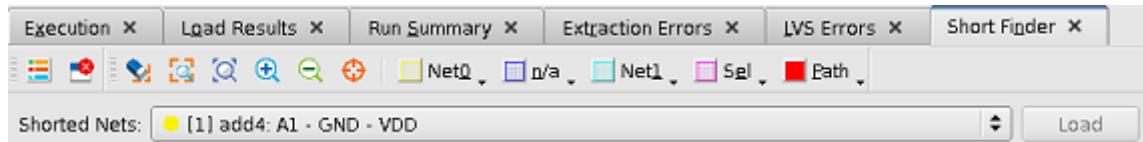
If a short is found using either of these settings, VUE results are loaded and text-based shorts are shown in the Short Finder tab.

### Loading Text-Based Short Finder Output

The Shorted Nets list, shown in [Figure 4-5](#), allows you to select the different shorts present in the design. A status color appears to the left of each net name:

- Red: Unloaded short; click the load button to load this short
- Yellow: Loaded short; this short is ready for processing
- Green: This short is fixed

*Figure 4-5 Shorted Nets*



### Net Info / Path Setup

#### Find

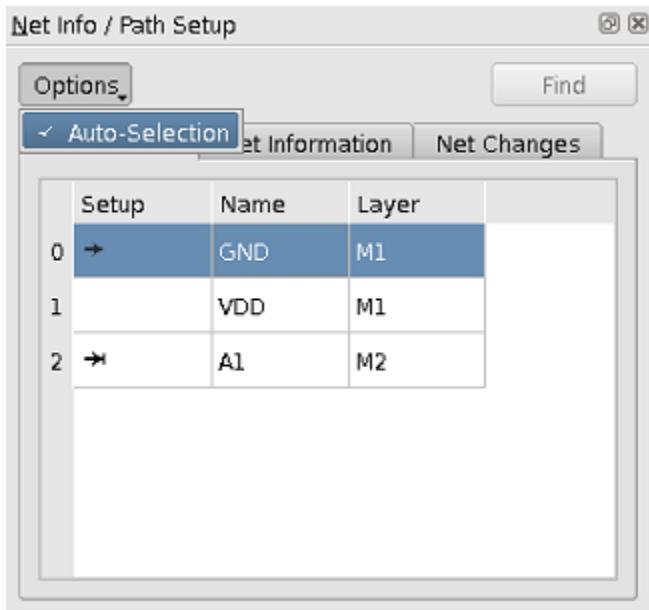
Click Find, shown in [Figure 4-6](#), to use the specifications from the Net Info / Path Setup pane to display the path in the Path Discovered pane. After you make changes, click Find again to display the updates.

#### Net Text

After you click Find, the Net Text tab displays information about the selected short in an alphabetized list. The Setup column identifies the start and end of the net. The Name and Layer columns indicate the name of the texts and the layer on which the texts are located. If displayed, the Coordinate column indicates the coordinates of the texts. See the “[Net Changes](#)” section later in this chapter for more information.

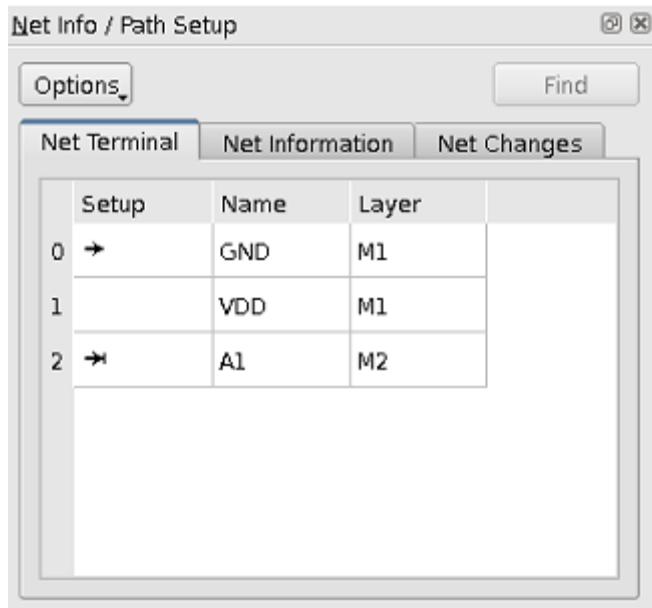
The options for Net Text are shown in [Figure 4-6](#).

Figure 4-6 Auto-Selection Option



With the Auto-Selection option selected, click Find. The Short Finder automatically selects start and end texts, which are shown with arrows in the Setup column, and finds the paths between the two selected texts. [Figure 4-7](#) shows the automatically-selected text.

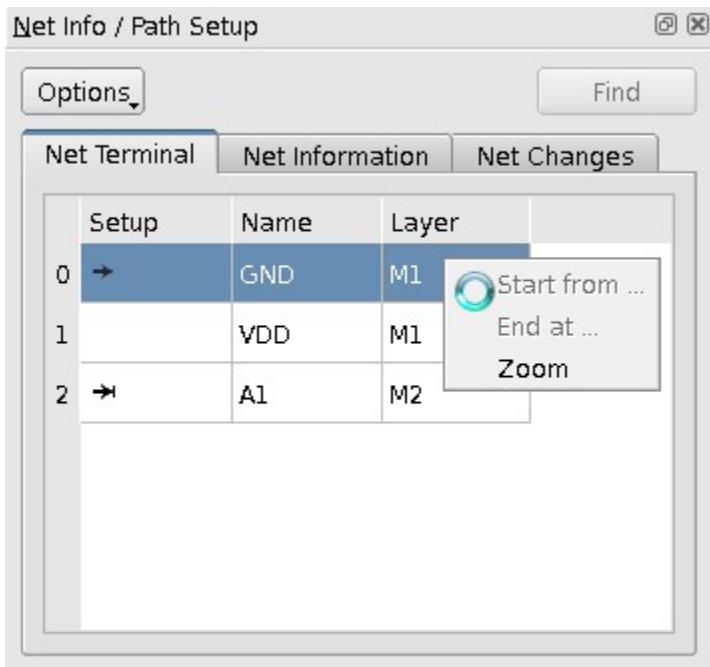
Figure 4-7 Start and End Texts



If Auto-Selection is not selected, you can select the start and end texts by performing the following steps:

1. To allow a net to be selected as the start net, right-click the net and choose Start from, shown in [Figure 4-8](#).

*Figure 4-8 Start from Command*



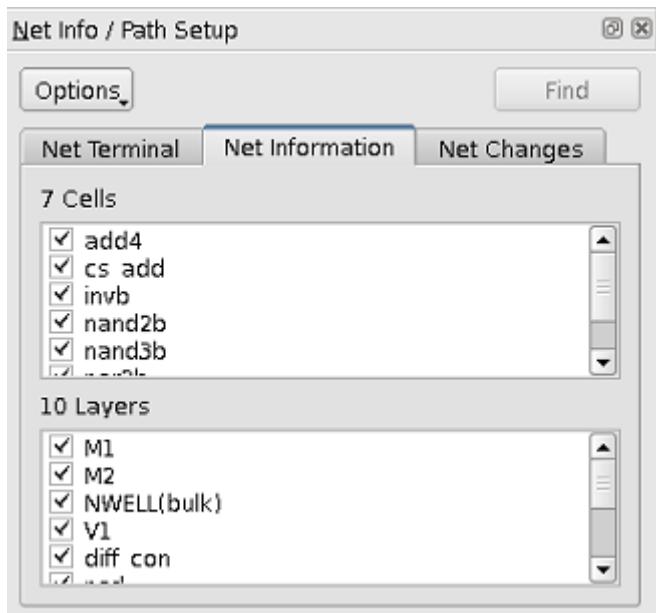
2. Right-click the text and choose the End at option.
3. To allow a different net to be selected as the end net, right-click the net and choose End at.

With this feature, you can focus on the paths you are interested in.

### Net Information

The Net Information tab, shown in [Figure 4-9](#), lists all of the cells and layers involved in the short paths. You can deselect specific cells or layers to exclude the paths that you are not interested in.

Figure 4-9 Net Information Tab



### Net Changes

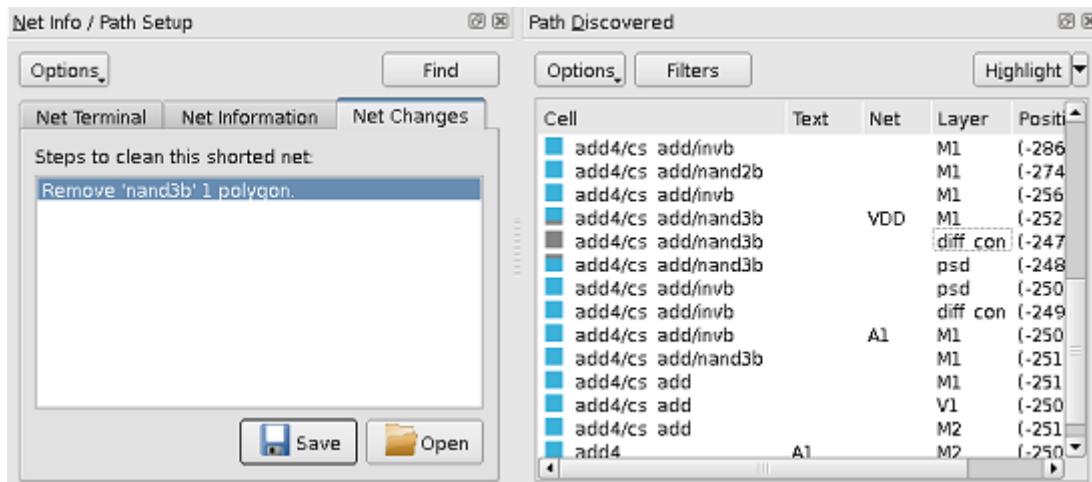
The Net Changes tab, shown in [Figure 4-10](#), displays your changes. When you choose Options > View Changes, the connected layout editor zooms to and highlights the related polygons, texts, and nets.

For example, if you remove a polygon from the Path Discovered pane, click Find. A new change is listed in the Net Changes view. You can right-click that change to zoom to and highlight it. For example, click Remove Polygon #2 to zoom to and highlight Polygon #2.

You can also save the changes to a .cgs file by clicking the Save button, or load the .cgs changes file by clicking the Open button.

#### Note:

You can roll back your changes by selecting a specific change instead of starting from the beginning.

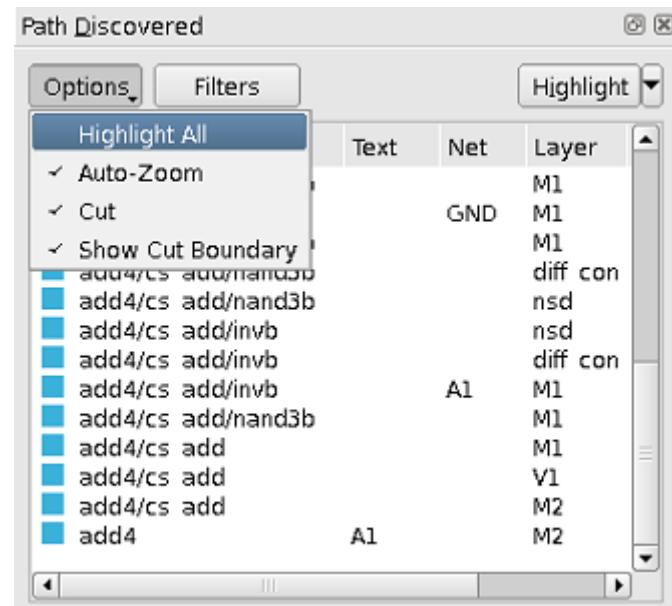
**Figure 4-10 Net Changes Tab**

Sometimes, Name and Layer cannot identify different texts because certain texts have the same name and location on the same layer. When you choose Options > Show Coordinates, the Short Finder displays the coordinates of the texts, so that you can identify the different texts.

## Path Discovered

### Options

Figure 4-11 shows the Options menu.

**Figure 4-11 Options Menu**

**Table 4-1** describes the actions that are available from the Options menu.

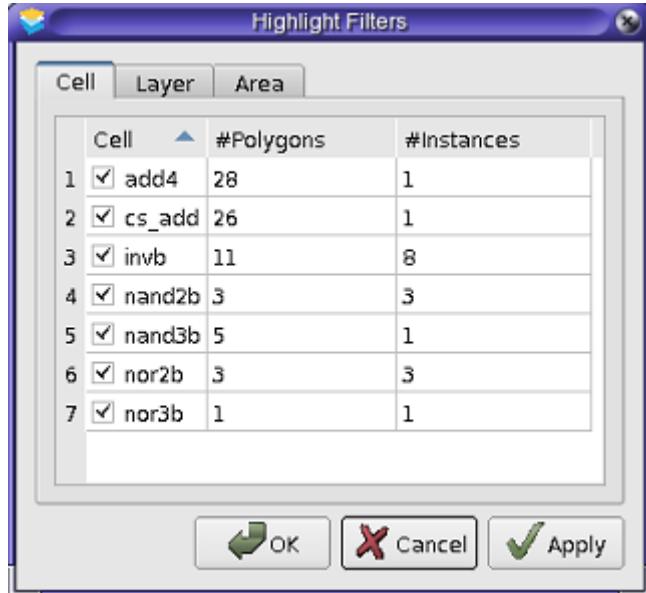
*Table 4-1 Options Menu Actions*

| Action            | Description  |
|-------------------|--|
| Highlight All     | Disable Highlight All to allow only the selected polygons to be highlighted when you click Highlight.  |
| Auto-Zoom         | When you enable Auto-Zoom, the layout editor automatically zooms to the extent of the highlight. If you do not select it, the layout editor does not zoom.                     |
| Cut               | When you enable Cut, which has a behavior similar to Filters > Area, if there is an include or exclude area and Cut is enabled, only polygons inside the area are highlighted. |
| Show Cut Boundary | When you enable Show Cut Boundary, which has a behavior similar to Filters > Area, the cut boundary is displayed.  |

## Filters

The Filters button opens a dialog box that allows you to filter the displayed polygons. See [Figure 4-12](#). Filtered polygons are dimmed in the polygon list and cannot be highlighted or selected in the layout.

*Figure 4-12 Highlight Filters Dialog Box*



**Table 4-2** describes the actions that are available from the Filters button dialog box.

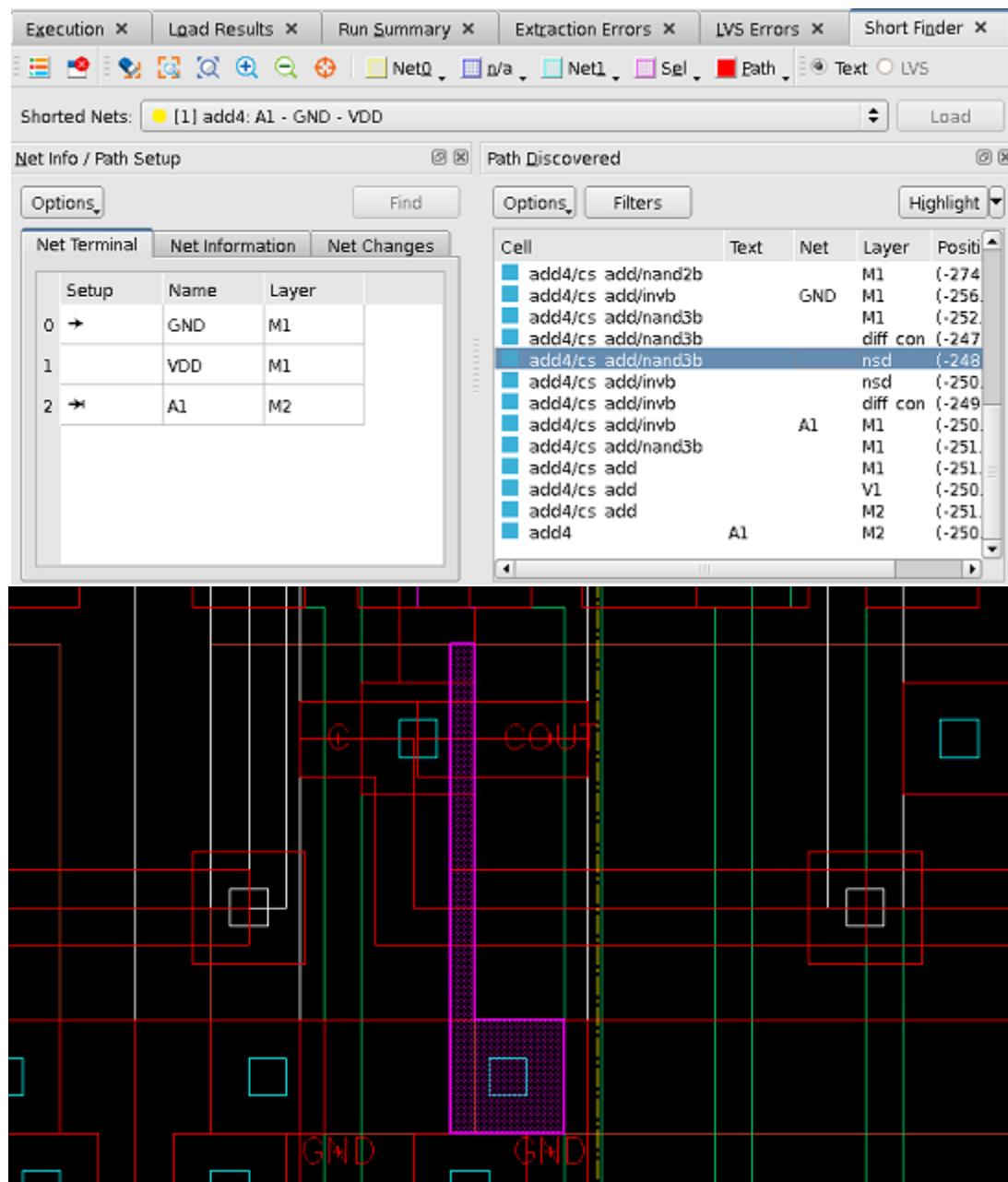
*Table 4-2 Filters Button Dialog Box Actions*

| Action | Description   |
|--------|---|
| Cell   | In the Cell view, shown in <a href="#">Figure 4-12</a> , you can select or deselect the cells to filter the polygons in the short path. “Polygon” in the Path Discovered pane.  |
| Layer  | In the Layer view, you can select or deselect layers to filter the polygons in the short path. See polygons in the Path Discovered pane.<br><br>Note:<br>Choosing Path Discovered > Filters > Cell or Layer and Net Info / Path Setup > Net Information shows that the first selection filters or hides only the polygons in the short path, while the second selection removes polygons from the short path. See the “Net Information” section for more information. |
| Area   | In the Area view, you can include or exclude the area where polygons are to be highlighted by selecting the area in the layout editor, or by inputting the coordinates of the area.<br><br>You can also set the Color/Fill of the cut boundary. See the Show Cut Boundary selection in the Options menu.  |

## Highlight

The Highlight button, shown in [Figure 4-13](#), highlights the selected polygon in the selected color. Moreover, the highlight button highlights all polygons in the short path only when you choose Highlight All from the Options menu. Polygons that are on the Start net are colored, as defined by Net0, End, and Net1. Polygons that are not associated with either the Start or End net are colored, as defined by n/a. The selected polygon is colored with the select color.

Figure 4-13 Highlight Button

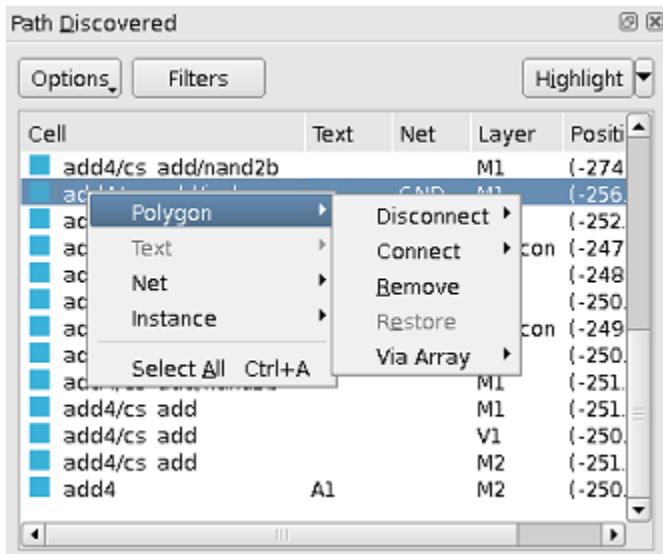


### Polygon List

The Short Finder tab lists the polygons in the short path in the Path Discovered pane, including Cell, Text, Net, and Layer columns. A single polygon can be selected by clicking it. To select multiple polygons, press Ctrl+Shift.

Figure 4-14 shows the Polygon menu.

*Figure 4-14 Polygon Menu*

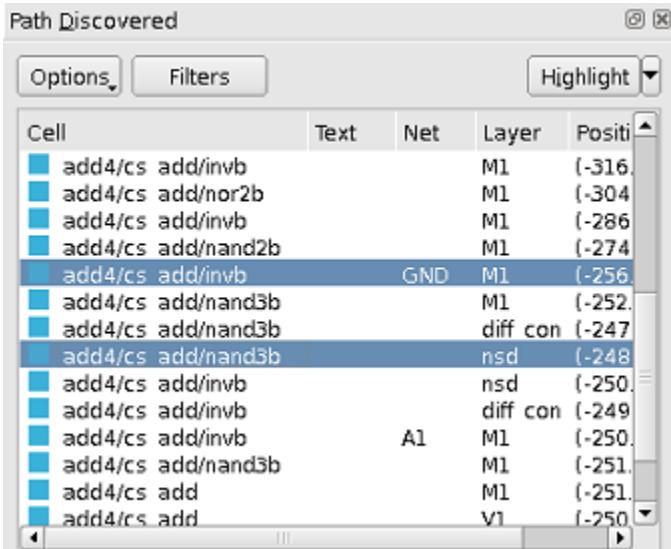


Click **Highlight** to highlight the selected polygons. Right-click the selected polygon to obtain options for the manipulation of those polygons. Certain changes you make might not be reflected until you click **Find**.

There are two methods for selecting multiple polygons.

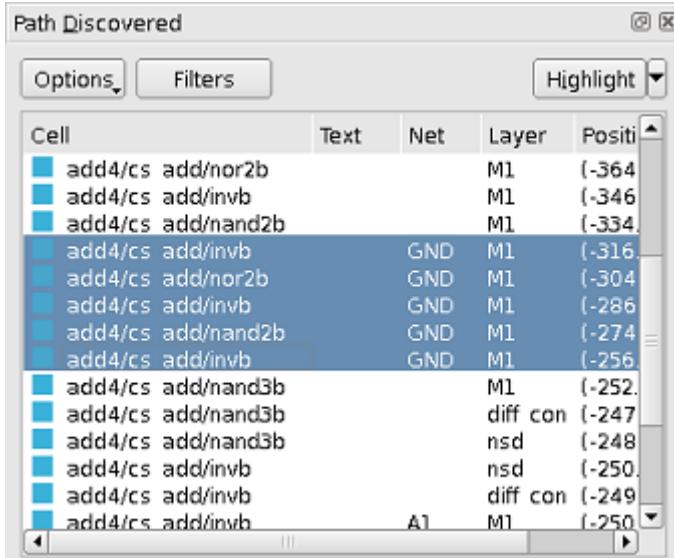
Press Ctrl-click to select the current multiple selected polygon, shown in Figure 4-15.

*Figure 4-15 Current Polygon*



Press Shift-click to select multiple selected polygons between the current and previous polygons, shown in [Figure 4-16](#).

*Figure 4-16 Multiple Select Polygons*



Pressing Ctrl+Shift has no effect on multiple selected polygons.

## Polygon

Within the polygon, a feature called pseudo-fixing changes the connectivity of the polygons at the cell level so that you can have a single run fix all of the shorts in memory. This feature allows you to eliminate the engine rerun time and increase the TAT time during power short correction.

[Table 4-3](#) describes the actions that are available from the Polygon menu:

*Table 4-3 Polygon Menu Actions*

| Action     | Description  |
|------------|--|
| Disconnect | The selected polygon is disconnected from the previous or next polygon.              |
| Connect    | The selected polygon is connected (if disconnected) to the previous or next polygon. |
| Remove     | The selected polygon is removed from the short path.                                 |
| Restore    | A previously removed polygon is added back.  |
| Via Array  | Set and unset the via array.   |

*Table 4-3 Polygon Menu Actions*

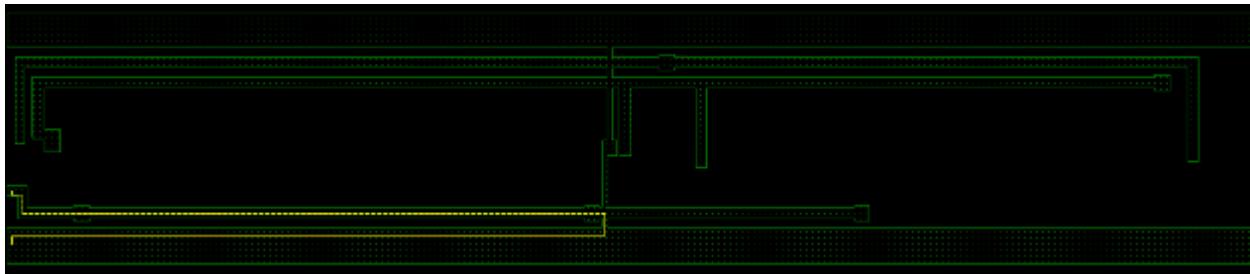
| Action             | Description  |
|--------------------|--|
| Draw Shorting Path | A line is drawn in the layout editor representing the shorting path for the selected polygons. See <a href="#">Figure 4-17</a> . |

### Note:

You must first operate on a polygon, for example, Disconnect or Remove, to enable the Set/Unset option of the Via Array selection.

Figure 4-17 shows a large metal polygon (green) with the shorting path also drawn (yellow).

*Figure 4-17 Draw Shorting Path*

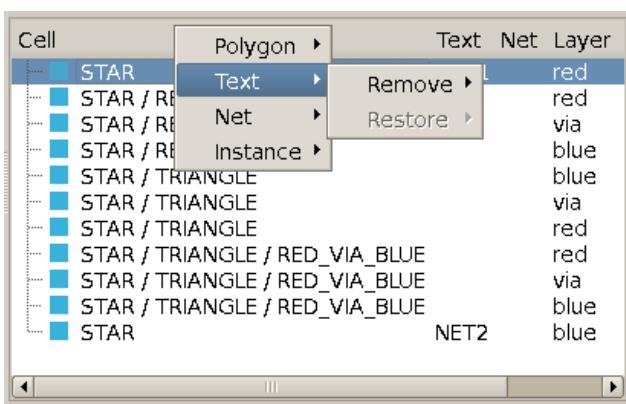


Use the Draw Shorting Path feature on shorts caused by large, merged polygons, such as power and ground structures or substrate shorts. When a large polygon is connected to two texts (or nets), choose the Polygon > Draw Shorting Path menu item to draw a line along the polygon connecting the two texts. The short can be found on the line.

Text

For a polygon with a text, you can remove the text from the polygon. You can also restore the original text to the polygon, shown in [Figure 4-18](#).

*Figure 4-18 Text Menu*



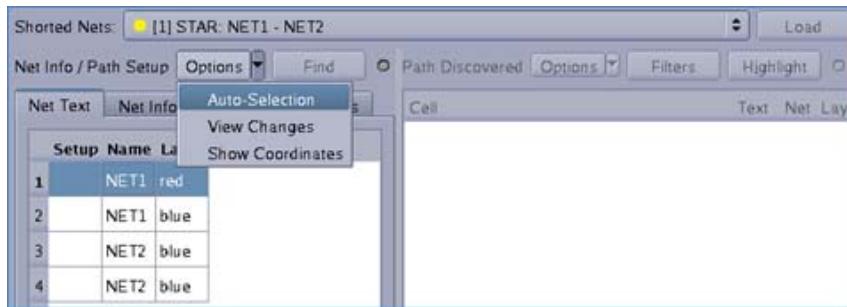
## Find Shortest Path Between Text Shorts

With this feature, you can configure the selection of shorts by using

- Text
- Coordinates

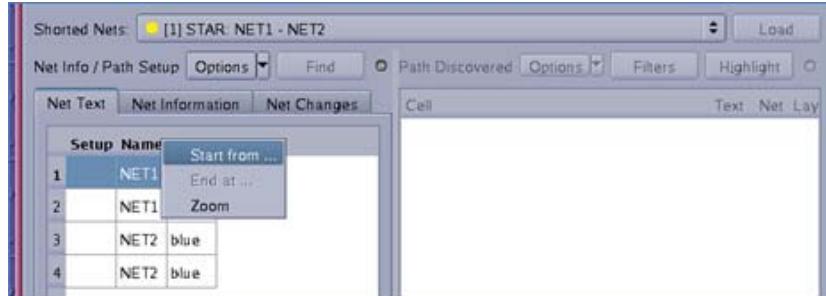
Selected paths are highlighted accordingly, shown in [Figure 4-19](#).

*Figure 4-19 Auto-Selection in Shorted Nets*



Net selection is shown in [Figure 4-20](#).

*Figure 4-20 Net Selection*

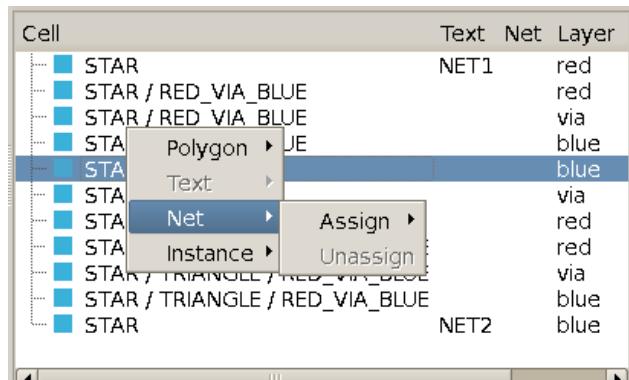


The highlighted net is shown in [Figure 4-21](#).

*Figure 4-21 Highlighted Net*

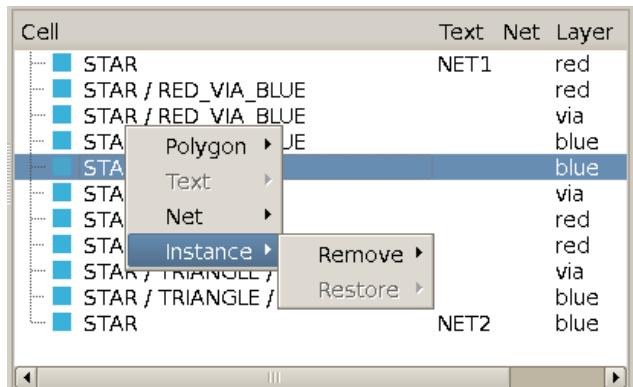
## Net

For a polygon without a net, you can assign a net to the polygon. You can also unassign the net from the polygon, shown in [Figure 4-22](#).

*Figure 4-22 Net Menu*

## Instance

For a polygon that is owned by an instance, you can remove the entire instance. You can also restore instances that have been previously removed, shown in [Figure 4-23](#).

*Figure 4-23 Instance Menu*

### **Hide or Unhide a Column**

To hide or unhide a column, right-click the Setup, Name, or Layer columns.

*Figure 4-24 Hide or Unhide*

A table with columns 'Setup', 'Name', and 'Layer'. A context menu is open over the 'Coordinates' column, with 'UnHide' selected. Other options in the menu include 'Coordinates' and 'All'.

|   | Setup | Name | Layer |  |
|---|-------|------|-------|--|
| 0 |       | NET2 | blue  |  |
| 1 |       | NET2 | blue  |  |
| 2 |       | NET1 | blue  |  |
| 3 |       | NET1 | red   |  |

### **Sort a Column**

To sort a column, click the Setup, Name, or Coordinates columns.

*Figure 4-25 Sort a Column*

A table with columns 'Setup', 'Name', and 'Coordinates'. A red arrow points to the top of the 'Coordinates' column header, indicating it is being sorted. The data rows are numbered 1, 2, 0, 3.

|   | Setup | Name | Coordinates           |
|---|-------|------|-----------------------|
| 1 |       | NET2 | (-13.4540 , -9.9240)  |
| 2 |       | NET1 | (0.4200 , 274.7160)   |
| 0 |       | NET2 | (149.2710 , 52.5410)  |
| 3 |       | NET1 | (153.4040 , 212.6510) |

---

## LVS-Based Short Finder

During the layout and route process, you can create shorts between nets. Not all shorted nodes have text. The LVS-based Short Finder feature uses diagnostic information from LVS results to augment the shorting path displayed in the VUE interactive Short Finder. The endpoints of the shorting path are defined by the pins on matched devices or cell instances. The LVS information is added to the shorting path to directly report the short location.

```
DIAGNOSTIC: Shorted layout nets
The following unmatched nets are highly suspected to indicate
source of shorts in the layout netlist.
Group 1 of 1:
Schematic net : connections Layout net : connections
-----
OUT : 4           OUT : 9
$1N7 : 5
```

When the LVS-based Short Finder data is loaded, VUE displays the Short Finder tab.

## Configuration of the LVS-Based Short Finder

To use the LVS-based short output in the Short Finder tab, you must execute a full device extraction and LVS run. The following input files are required:

- LVS runset
- Schematic netlist
- Layout database
- Equivalence points (can be user-specified or system generated)

To enable the LVS-based short output from the Short Finder tab, you can do one of the following:

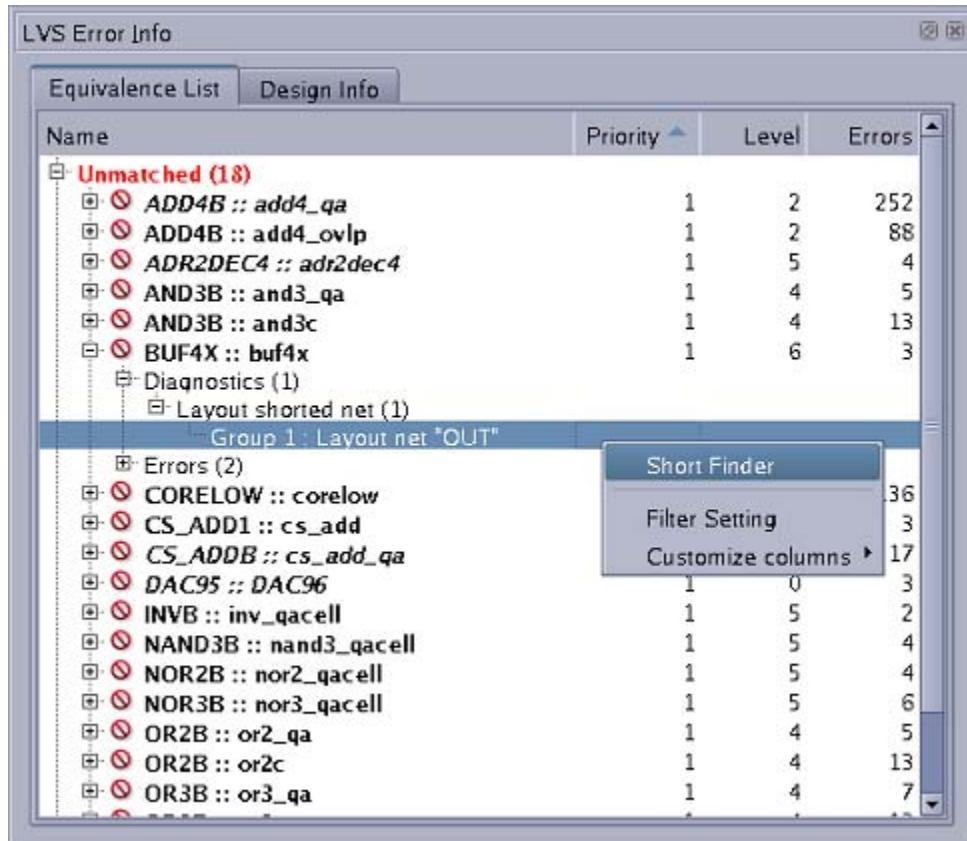
- Specify the `-create_lvs_short_output` command-line option.
- Set the `create_lvs_short_output = true` argument of the `text_net()` function.

If a short is found using either of these settings, VUE results are loaded and LVS-based shorts are shown in the Short Finder tab.

## Loading LVS-Based Short Finder Output

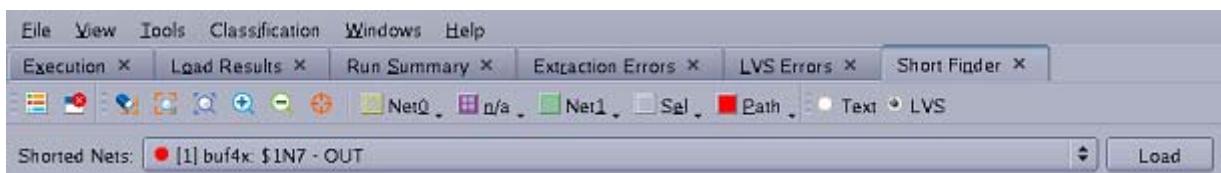
You can launch the interactive LVS-based Short Finder feature when you debug LVS shorts from within the LVS Errors tab, shown in [Figure 4-26](#).

Figure 4-26 LVS-Based Short Finder Feature Launched in LVS Errors Tab



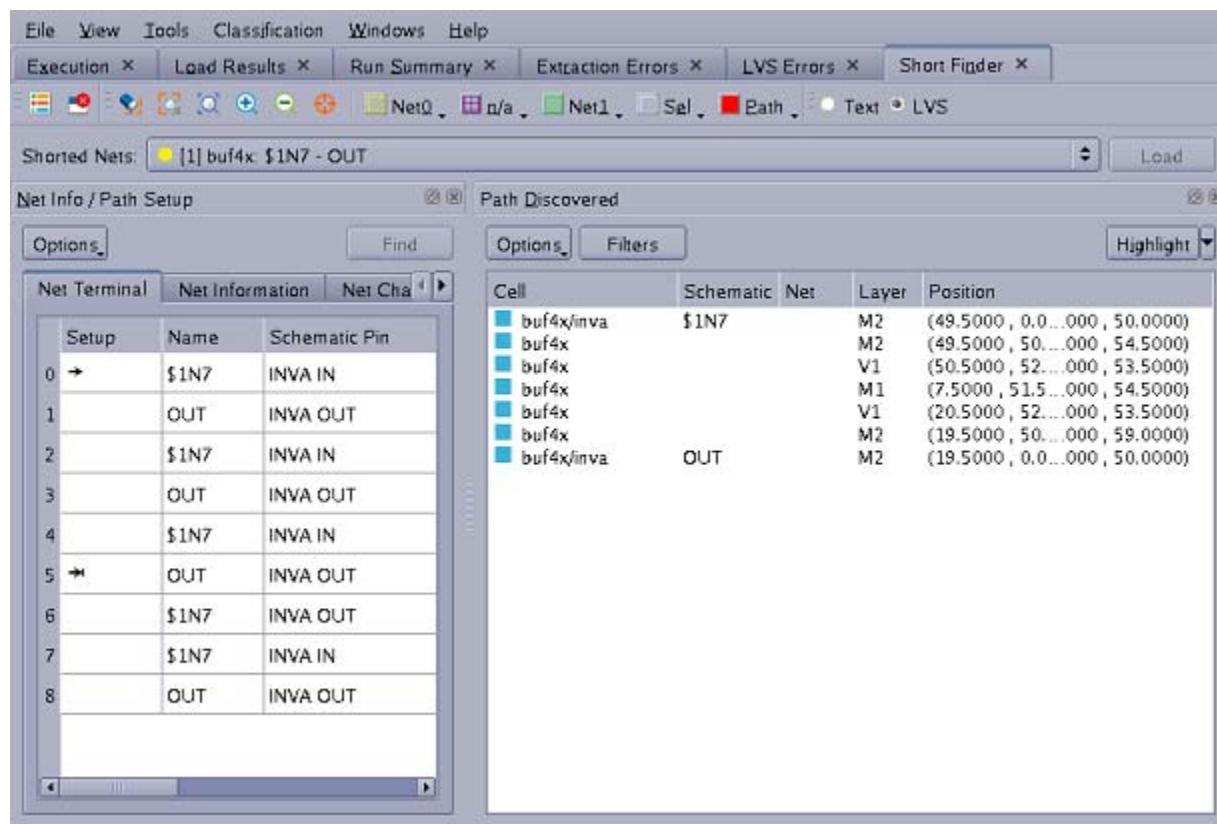
You can see all of the LVS shorts by switching between the Text and LVS options, shown in [Figure 4-27](#).

Figure 4-27 All LVS Shorts Listed by Text and LVS Options



The Net Info / Path Setup for the LVS-based Short Finder is similar to the setup for the text-based Short Finder. Unlike the text-based Short Finder, the LVS-based Short Finder selects two pins to use at the starting and stopping points for the path and for clicking Find. The Path Discovered pane populates with the path. The organization of the net and path information is similar to the text-based Short Finder, with a few additions to identify the LVS information, shown in [Figure 4-28](#).

Figure 4-28 LVS-Based Short Finder Output





# Index

---

## A

Auto Load Results option 3-5

## C

Cadence® Virtuoso® Layout Editor 2-7  
Case Sensitive 1-30  
cells violations, DRC violations 3-7  
changing color and fill  
    DRC errors, primary 1-32  
    DRC errors, secondary 1-32  
Classification 1-13  
Command button 3-2  
command-line options 1-3  
Communication GUI 2-3  
Composer 2-7  
constraints, numerical values 3-12  
conventions for documentation 3-ix  
CosmosSE 1-4  
CREATE\_VUE\_OUTPUT option 1-3  
cross-probing 3-33  
customer support 3-x

## D

Details tab 3-36

Device Grouping 3-31

Disable loading error list of exceeding a threshold value 1-17

displays errors, setting 1-10

drc errors 3-7

DRC errors, highlighting  
    primary 1-32  
    secondary 1-32

DRC tab 3-7

## E

Edit-In-Place mode 1-10  
Enable/Disable classifications 1-14  
Enable/Disable filter functions 1-17  
equivalence point 3-32  
Error Classification Control 3-9  
Error list 1-17  
Error List Pane 3-9  
error viewing section 3-29  
errors, viewing 3-7  
executing PrimeYield 3-2  
Execution tab 3-2  
exit status 3-5

## F

File menu 1-8  
files, required 1-2  
Find Shortest Path Between Text Shorts (FSPBTS) 4-18

## G

graphical user interface, VUE 2-2

## H

Help browser, VUE 1-28  
Highlight List window 3-33  
Highlight Next 1-31  
Highlight Previous 1-31  
highlights  
    clearing 1-31  
    setting 1-10

## I

IC Compiler 2-6  
IC WorkBench EV Plus 2-4  
.icv\_vuer file 1-8  
Import information from layout button 3-3

## K

Keep Existing Highlights 1-31

## L

Laker™ Custom Layout System 2-7  
layout tools, connecting to 2-3  
LVS errors 3-29  
LVS Errors tab 3-29

## M

Messages window 3-5

## N

Net Trace Tab 3-37  
Netlist Browser 3-32  
Netlist Visualizer 3-41  
numerical values, constraints 3-12

## O

Off/ Pan/ Zoom 1-31  
Options window 1-30

## P

preferences, VUE 1-8  
probe point 1-31

## R

Run Summary tab 3-6  
running VUE 3-2  
runset requirements 1-3

## S

schematic tools, connecting to 2-3  
Search 1-30  
Search Cell Name 1-30  
Search Violation Name 1-30  
Short Finder 4-4  
soft\_check Graphical Violation Report 3-24  
SolvNet  
    accessing 3-x  
    documentation 3-viii  
starting VUE 3-2  
Stop 1-31

**T**

toolbar, VUE 1-28  
Tools menu 1-9

Virtuoso 2-7

VUE Options Dialog Box 1-10  
VUE preferences, saving 1-8

**V**

View menu 1-9  
viewing errors 3-7  
Violation Browser Pane 3-9  
Violation Comment 1-16

**Z**

zooming  
highlight extents 1-31  
in 1-31  
open block 1-31  
out 1-31