IC Validator VUE User Guide

Version T-2022.03-SP3, September 2022

SYNOPSYS®

Copyright and Proprietary Information Notice

© 2022 Synopsys, Inc. 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 https://www.synopsys.com/company/legal/trademarks-brands.html.
All other product or company names may be trademarks of their respective owners.

Free and Open-Source Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

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.

www.synopsys.com

Feedback

Contents

New in This Release 8

Related Products, Publications, and Trademarks 8

Conventions 9

Customer Support 9

Statement on Inclusivity and Diversity 10

1. Introduction to IC Validator VUE 11

VUE Setup 12

Licensing and Resource Requirements 12

Required Files 12

Runset Requirements 13

VUE Execution 13

VUE Command-Line Options 13

Bindkey Setting 14

VUE Preference System 16

VUE Menu Overview 18

File Menu 18

View Menu 19

Tools Menu 19

VUE Options Dialog Box 20

General Options 20

DRC/Extraction Options 22

LVS Options 28

Classification Menu 30

Windows Menu 33

Help Menu 39

VUE DRC Toolbar 40

2. VUE Communications Interface Guide 44

Graphical User Interface 44

Main VUE GUI 45

Communication GUI 46

Tools Menu 47
IC Validator WorkBench 47
IC Validator WorkBench Command Line 47
Within IC Validator WorkBench 48
VUE Command-Line Options 48
Communication Window 48
Communicating With IC Validator WorkBench 49
IC Compiler 49
IC Compiler II 49
Custom Compiler 50
Laker Custom Layout System 50
Cadence Virtuoso Layout Editor 50
Running IC Validator VUE From Virtuoso 51
VUE Connections in Virtuoso 51
Autopopulating VUE Input 53
Exporting DFII or OpenAccess Capabilities and Usage Within VUE 57
Exporting CDL Capabilities and Usage Within VUE 62
Postprocessing CDL Netlists Within the IC Validator Run Stream 64
Limitations 65

3. Running IC Validator Within VUE 67
Executing IC Validator From VUE 68
Loading IC Validator Results Into VUE 72
Result File 73
Backward Compatibility 75
Run Summary 75
DRC Errors and Device Extraction Violations 76
Violation Browser Pane 78
Error List Pane 78
Violation Detail Pane 78
Error Classification Control 78
Unmatched Error Classification Browser 79
DRC Ease-of-Use Support 80
Predefined Keyboard Shortcuts 81
Predefined Mouse Keys 81
Filter Settings 83

Sorting the Error List 86
Sorting the Violation Browser 87
Copying the Error List 88
Searching the Error List 88
Selecting the Error List 89
Distinguishing Between User and cPYDB Classification 90
Waived Results 91
Detailed Information in Tooltips 92
Adjusting Column Location 93
Switching to Non-Detail Mode 93
Hiding Function Violations 94
Iterative Highlighting 95
The soft_check Graphical Violation Report 95
LVS Errors 100
Device Grouping 103
Netlist Browser 104
Highlight List 105
Details Tab 108
Net Trace Tab 109
Select Port Net 112
Netlist Visualizer 113
Netlist Visualizer Options 114
Netlist Visualizer Interface 115
Rendering Objects in the Graphical Canvas 116
Devices 116
Ports 117
Nets 118
Rendered Color for Matched and Unmatched Objects 118
Selected Objects 119
Highlighted Objects 120
Progress Bar 122
Adding Objects to the Graphical Canvas 123
Adding Objects From the Text Window 123
Adding Objects From the Graphical Canvas 126
Adding Objects From the Equivalence List 128
Interacting With Objects in the Graphical Canvas 128
Graphical Canvas Menu 129
Graphical Canvas Navigation 131
Bindkey Settings for Netlist Visualizer 131

Layer Debugger 132
Setup 132
Layer Debugger Execution Options 134
Layer Debugger With Selected Debug Objects 134
Running the Layer Debugger 138
Debugging the Run 138
Connect Debugger 139
Setup 140
Loading the Results 140
Connect Debugger Layout 140
Debugging a Connect Sequence 141

4. Resolving a Short 146
Short Static Output in Extraction Errors Tab 147
Configuration of Short Static Output 147
Loading Short Static Output 148
Interactive Short Finding in the Short Finder Tab 149
Text-Based Short Finder 149
Interacting With a Short Path 158
Configuration of the Text-Based Short Finder 161
LVS-Based Short Finder 161
Configuration of the LVS-Based Short Finder 161
Loading LVS-Based Short Finder Output 162

5. Introduction to IC Validator Error Heat Map 165
Executing the Heat Map 166
Heat Map Interface 167
Violation Heat Map 167
Violation Category 170
Violation Selection 170
Color Definitions Window 171
Sliding Bar 171
Overlaying the Error Heat Map With the Layout Editor 172
Highlighting Error Markers From the Heat Map 173
Highlighting Cell Outlines 174
Selecting Related Violations 175
Searching and Filtering Violations 177

Contents

Top-Cell Only Heat Maps 178
Heat Map Grid Statistics 179
Saving the Heat Map to a Disk 179
Restrictions and Limitations 181
Licenses 182

About This User Guide

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

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.

This preface includes the following sections:

- New in This Release
- Related Products, Publications, and Trademarks
- Conventions
- Customer Support
- Statement on Inclusivity and Diversity

New in This Release

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

Related Products, Publications, and Trademarks

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

https://solvnetplus.synopsys.com

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

- Synopsys® Custom Compiler™
- Synopsys IC Compiler™
- Synopsys IC Compiler™ II
- Synopsys Fusion Compiler™
- Synopsys StarRC™

Feedback

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
`Courier`	Indicates syntax, such as `write_file`.
`Courier italic`	Indicates a user-defined value in syntax, such as `write_file` *`design_list`*
`Courier bold`	Indicates user input—text you type verbatim—in examples, such as `prompt>` **`write_file top`**
`Purple`	• Within an example, indicates information of special interest. • Within a command-syntax section, indicates a default, such as `include_enclosing =` **`true`** `| false`
[ ]	Denotes optional arguments in syntax, such as `write_file [-format` *`fmt`*`]`
...	Indicates that arguments can be repeated as many times as needed, such as *`pin1 pin2 ... pinN`*.
|	Indicates a choice among alternatives, such as `low | medium | high`
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Bold	Indicates a graphical user interface (GUI) element that has an action associated with it.
Edit > Copy	Indicates a path to a menu command, such as opening the **Edit** menu and choosing **Copy**.
Ctrl+C	Indicates a keyboard combination, such as holding down the Ctrl key and pressing C.

Customer Support

Customer support is available through SolvNetPlus.

Feedback

Accessing SolvNetPlus

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

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

https://solvnetplus.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 SolvNetPlus site, click REGISTRATION HELP in the top-right menu bar.

Contacting Customer Support

To contact Customer Support, go to https://solvnetplus.synopsys.com.

Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

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

Feedback

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, Elite, or Base licensing, and resource requirements, see Chapter 2, “Licensing and Resource Requirements,” in the *IC Validator User Guide*. When using Elite or Base licensing, ICValidator-Manager is the main license. With this license, IC Validator checks only for the presence of a valid license every time VUE is run.

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_INSTALL_DIRECTORY*), shown in Table 1:

Table 1 Required VUE Files

VUE file	Description
ICV_INSTALL_DIRECTORY/bin/systype/icv_vue	VUE binary
ICV_INSTALL_DIRECTORY/etc/VUE/*	VUE command files
ICV_INSTALL_DIRECTORY/etc/tcl-u/Icvwb Vue.tcl	Tcl code for IC Validator WorkBench connection
ICV_INSTALL_DIRECTORY/lib-u/systype/VUE/*	VUE connection filter files

Feedback

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 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 2.

Table 2 VUE Command-Line Options

Argument	Description
`-64`	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 `-64` command-line option.
`-cmdPrint`	Toggles the command print-out.
`-lic_elite`	Forces Elite licensing scheme.
`-lic_base`	Forces Base licensing scheme.
`-lay layTool`	Layout tool on which to connect during startup. Supports Custom Compiler™, IC Compiler™, IC Validator WorkBench, Laker™ Custom Layout System, and Cadence® Virtuoso® Layout Editor.

Table 2 VUE Command-Line Options (Continued)

Argument	Description
`-layArgs` *`argName argValue`*	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 Chapter 2, VUE Communications Interface Guide.
`-ld`	Runs VUE in Layer Debugger mode.
`-load vueFile`	Autoloads the information found in the *cell*.vue file. By default, no data file is loaded on VUE startup. For more information about where the .vue file is defined, see the “Result File” section in Chapter 3, Running IC Validator Within VUE.
`-schArgs` *`argName argValue`*	Assigns arguments and their values that are needed when VUE connects to the schematic editor.
`-sch` *`schTool`*	Schematic tool on which to connect during startup. Supports Composer and CosmosSE™.
`-usage | -h | -help`	Displays usage message and exit.
`-V`	Prints product version.
`-Version`	Prints 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.

Feedback

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 or 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.Metal1.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_INSTALL_DIRECTORY*/etc/VUE/.icv_vuerc
    - Disables certain VUE functionalities:
        - Supports editing by CAD or IT to disable certain VUE functionalities from third-party users
        - Overrides incrementally the same settings by previous setting files
        - 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 3 describes these file menu commands.

Feedback

Table 3 File Menu Commands

File menu command	Description
Load Run Settings	Loads a valid VUE run settings file. The default is `.icv_vue.runs`. See the VUE Preference System for more information about the run settings
Save Run Settings	Creates a VUE run settings file in the directory you specify. The default is `.icv_vue.runs`. See the VUE Preference System 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 4 describes the View menu commands.

Table 4 View Menu Commands

View menu command	Description
Run Result	*cell*.RESULTS file
Run Summary	run_details/*cell*.sum file
Layout Errors Summary	*cell*.LAYOUT_ERRORS file
TOP Layout Errors Summary	*cell*.TOP_LAYOUT_ERRORS file
LVS Errors Summary	*cell*.LVS_ERRORS 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.

Feedback

VUE Options Dialog Box

As shown in Figure 1, the VUE Options dialog box allows you to set how VUE displays errors or highlights objects.

General Options

Figure 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 After Highlighting**. 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 2 shows the Dockable Windows feature in the General Options window.

Figure 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 3 and Figure 4.

Figure 3 DRC/Extraction Options Page

Figure 4 DRC/Extraction Options Page Part II

Feedback

- **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 5 shows an example of the classification selections.

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

Figure 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 7 shows the “Row color by classification count” selection.

Figure 7 DRC/ Extraction: Row color by classification count

- **Customize columns**. Specifies whether the zero-count columns are displayed upon loading. The default is enable.
- **Show flat errors column upon loading**. Adds a column to the violation browser that shows the flat error count for each violation. The value of Flat Errors is defined as Error number * Cell flat placement count.

  Note:

  The error number is the value of the error column, and it changes under certain operations, such as error classification. For example, when you classify an error as Fixed, the error number decreases by 1. The Flat Errors count is adjusted accordingly. The cell flat placement count is the placement number of the cell where the error was reported. See Figure 8.

Figure 8 Flat Errors

- **Violation comment**. Specifies whether only the rule name is shown. The default is off.

  Figure 9 shows the “Show rule name only” selection.

Figure 9 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.

Feedback

Use the Filter Settings, as shown in Figure 62, to quickly set up the Violation or Classification filter settings.

- **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 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 a layout editor such as 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 10.

Figure 10 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. Alternatively, 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.

Figure 11, shows the Off and Per Highlight buttons from the LVS Error Info tab.

Figure 11 LVS Toolbar Options

The Off toolbar option provides the highlight color for each object type from the LVS Error Info tab. The Per Highlight toolbar option provides a highlight color button near the option that automatically changes colors when highlighting multiple objects.

Note:

The Auto Change > Per Highlight feature in DRC and LVS are independent of one another.

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 9, “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 12, 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 12 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 9, “DRC Error Classification” in 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 13 is an example of a classification report.

Figure 13 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 9, “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 the 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 14 shows the DRC errors in the Windows menu.

Figure 14 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 15 and Figure 16 show examples of docking and undocking a widget.

Feedback

Figure 15 Windows Menu: Undocking a Widget

Figure 16 Windows Menu: Docking a Widget

The removable windows feature enables you to remove and recover certain windows from the VUE interface. Figure 17 and Figure 18 show examples of removing and recovering a window.

Figure 17 Windows Menu: Removing a Window

Figure 18 Windows Menu: Recovering a Window

The removable windows feature also enables you to remove and recover certain widgets from the VUE DRC interface. Figure 19 and Figure 20 show examples of removing and recovering a widget.

Figure 19 Windows Menu: Removing a Widget

Figure 20 Windows Menu: Recovering a Widget

Figure 21 and Figure 22 show examples of merging multiple widgets into one table and splitting widgets from the table.

Figure 21 Windows Menu: Merging Multiple Widgets Into One Table

Figure 22 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 23 shows an example of the Top/Bottom and Left/Right window layouts.

Figure 23 Windows Menu: Resetting the Window Layout

Figure 24 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 24 Violation Details

For more information about the DRM link setting, see the VUE Preference System on page 16.

Help Menu

The *IC Validator VUE User Guide*, *IC Validator User Guide*, *IC Validator Deck Creation and Validation User Guide*, *IC Validator Launch User Guide*, *IC Validator LVS User Guide*, *IC Validator Distributed Processing Systems Quick Start Guide*, and *IC Validator Reference Manual* are available exclusively from Documentation on the Web, which provides HTML and PDF documents and is available on SolvNetPlus.

To access the SolvNetPlus site:

1. Go to the web page at https://solvnetplus.synopsys.com.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register.)

If you need help using the site, click **Help** on the menu bar.

Feedback

To enable the IC Validator VUE-based help system, download the icvVUEug.pdf, icvug1.pdf, icvlvsug.pdf, icvdp.pdf, icvlaunchug.pdf, icvdcvug.pdf, and icvrefman.pdf files from SolvNetPlus and place them into *ICV_INSTALL_DIRECTORY*/doc/icv/pdf/.

VUE DRC Toolbar

As shown in Figure 25, 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 26.

Figure 25 VUE DRC Toolbar With Four Groups

Figure 26 Relocating Toolbar Groups Within Other Groups

Figure 27, Figure 28, and Table 5 show the toolbar functions for each of the four groups.

Figure 27 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.

Feedback

Figure 28 VUE DRC Highlight Color and Fill

Table 5 defines the VUE DRC options.

Table 5 VUE DRC Toolbar Options

Icon	Group	Option	Description
Search (Ctrl+E)	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 5 VUE DRC Toolbar Options (Continued)

Icon	Group	Option	Description
	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. 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: Off: No view change.
	Highlight	Pan	Highlight zoom settings: Pan: Pan to highlights.
	Highlight	Zoom ratio	Highlight zoom settings: Zoom: Zoom to highlight with ratio.
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.

Table 5 VUE DRC Toolbar Options (Continued)

Icon	Group	Option	Description
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 Validator WorkBench
- 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.

Feedback

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 29. The names of the connected layout and schematic tools are displayed with a tool tip.

Figure 29 Main VUE Window

Note:

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

Feedback

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 30.

Figure 30 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 29. The names of the connected layout and schematic tools are displayed with a tooltip.
  - Table 6 shows the layout and schematic communication status.

  See the VUE Communications Interface Guide for more information about the Communication window.

Table 6 Communication Status

Status	Description
L S	Layout is not connected; schematic is not connected.
L S	Layout is connected; schematic is not connected.
L S	Layout is not connected; schematic is connected.
L S	Layout is connected; schematic is connected.

IC Validator WorkBench

VUE communicates with IC Validator WorkBench through a socket or port connection. This socket must be opened by IC Validator WorkBench. You can open the communication socket manually upon IC Validator WorkBench startup, or within the IC Validator WorkBench 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 Validator WorkBench Command Line

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

```
icwbev -socket number
```

Feedback

Within IC Validator WorkBench

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

```
user_socket open 0
```

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

IC Validator WorkBench connects to VUE automatically when you set the ICVWB_USER environment variable:

- For newer versions of IC Validator WorkBench:

  ```
  setenv ICVWB_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 Validator WorkBench and VUE from the UNIX command prompt and connect directly to the existing IC Validator WorkBench 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 Validator WorkBench 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 Validator WorkBench 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 Validator 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 Validator WorkBench 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 Validator WorkBench

To communicate with another IC Validator WorkBench session, open the Communication dialog box and choose IC Validator WorkBench 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 in the Layout or Schematic window. 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 and tested in version v6.10 and expected to work with version 12.x and 18.x.

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") "/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.

To use a standalone VUE run with Virtuoso,

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

   ```
   setenv STANDALONE_VUE_FOR_VIRTUOSO 6678
   ```

2. Run Virtuoso.
3. Run `icv_vue -lay virtuoso -layArgs Port 6678`.

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 SkillVueMenu.il 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 31, the status of the connection is displayed by tool tip, which contains the editor type, window ID, library name, cell name, and view type.

Figure 31 Status of the Connected Window

Autopopulating VUE Input

Autopopulating the input boxes occurs when the information is provided within the IC Validator runset file `library()` function or when the Layout or Schematic is connected. In Figure 32, both the Layout and Schematic are connected.

Figure 32 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 33.

Figure 33 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 34.

Figure 34 Communication

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 35 and Figure 36.

Figure 35 VUE Communication - Layout Tool and Window Configuration

Figure 36 VUE Communication - Schematic Tool and Window Configuration

Figure 37 VUE Communication - Layout and Schematic Editor Tooltips

You can fill in or load runset information from the browse button. After the information is populated, click Load Runset, as shown in Figure 38.

Figure 38 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 39.

Figure 39 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 40 and Figure 41.

Figure 40 DFII Export Options

Figure 41 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, object 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 40, 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 41, 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 42 shows an example of the .vueDFII_export file. This file can be changed manually and is automatically loaded with VUE.

Figure 42 .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                      ""
    '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/`

Feedback

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 43 shows the Schematic Library Information dialog box.

Figure 43 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 44.

Figure 44 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 42. 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 45.

Figure 45 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

Feedback

Executing IC Validator From VUE

The Execution tab, as shown in Figure 46, 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 that is executed and defaults to `icv -vue`. It can also be used to execute a script instead of directly calling IC Validator.

  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 46.

Figure 46 Execution Tab With Options

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 47 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 47, 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 48 shows the log window when DP is selected.

Figure 48 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 49 shows the log window when DRC is selected.

Figure 49 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 single schematic netlist or library, or multiple netlists for use in the IC Validator run. Figure 50 shows the log window when LVS is selected. The log

Feedback

window default is for a single netlist or library. For multiple netlists, select the Multiple Input option from the Source menu, shown in Figure 51.

Figure 50 Options Tab With LVS Options for Single Input

Figure 51 Options Tab With LVS Options for Multiple Input

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.

Result File

The result file (*.vue) is an ASCII file that contains the information VUE needs to correctly display results. The IC Validator tool automatically generates the file 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 Result File box is automatically populated if the *.vue file exists in the same directory from which VUE was started. Otherwise, you can manually browse for *.vue.

Figure 52 shows an example of the Load Results tab where the Result File is automatically populated.

There are three ways to quickly access the *.vue file.

1. Select the file from the file browser icon and either click open or double-click the file.
2. Enter the *.vue file name in the edit line and press Enter.
3. Double-click *.vue in the file list.

Figure 52 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 *.vue file is relative to the run directory. If 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 *.vue file to retrieve the information.

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

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

The IC Validator version that generates the result file, and the VUE version 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 53 is an example of the Run Summary tab.

Figure 53 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 54 is an example of the DRC Errors tab.

Figure 54 DRC Errors Tab

Feedback

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 9, “DRC Error Classification” in the *IC Validator User Guide* for more information about this flow.

Table 7 includes descriptions of each error classification.

Table 7 Error Classification

Icon	Classification	Description
	Error	Unclassified error (default classification).

Table 7 Error Classification (Continued)

Icon	Classification	Description
	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 `density(pydb_output = true)` 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 9, “DRC Error Classification” in the *IC Validator User Guide*for more information.

Figure 55 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 55 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

Feedback

Predefined Keyboard Shortcuts

Table 8 includes descriptions of each predefined keyboard shortcut.

Table 8 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 56.
- Apply in other than Violation Browser title. See Figure 57.
  - Cell, Violation, Function, and Error List
  - Unmatched and Debug
  - Count columns

Figure 56 shows RMB: Right-Mouse-Button support for the most useful function within RMB.

Figure 56 RMB: Apply in Violation Browser Title

Feedback

Figure 57 RMB: Apply Other Than Violation Browser Title

Filter Settings

Figure 58 and Figure 59 show you how to quickly filter and unfilter a classification error by right-clicking the titles of the classification count.

Figure 58 Example of Filtering Certain Classifications by RMB

Figure 59 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 60.

Figure 60 Example of Filtering Certain Violations by RMB

Figure 61 shows you how to unfilter violations. This capability can be accessed only from the RMB menu in the first column.

Figure 61 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 62.

Figure 62 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 63.

Figure 63 Example of Filtered Violation Settings Saved to Run Settings File

Sorting the Error List

Figure 64 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 64 Example of Sorting the Error List

Sorting the Violation Browser

Figure 65 shows how you sort the count in the Violation Browser by left-clicking the category title directly.

Figure 65 Example of Sorting the Violation Browser

Copying the Error List

Figure 66 shows how to copy the Error List information to other text editor, such as an e-mail or in an Excel file.

Figure 66 Example of Copying the Error List

Searching the Error List

Figure 67 shows an example of searching the Error List, as well as how to search limited data from the Error List.

Figure 67 Example of Searching the Error List

Selecting the Error List

Figure 68 shows how to quickly select a series of error lists from the currently selected error.

Figure 68 Example of Selecting the Error List

Figure 69 shows how to quickly select a series of error lists from the currently selected multiple errors.

Figure 69 Example of Selecting the Next Error List

Distinguishing Between User and cPYDB Classification

Figure 70 shows the first four error classifications from the input error classification database (cPYDB) that are specified in the runset file by the `error_option()` function. When you load the run results for the first time, the first four errors are marked as “From cpydb.” When you waive the fifth and sixth errors as Waive, they are marked as “From vue.”

Figure 70 Example From Input cpydb

Waived Results

If the original error is from the cPYDB and you classify it again using a different error type, the error classification comes from VUE rather than the cPYDB. If you reclassify them using the same error type, the “From” column shows “cpydb.” In Figure 71 after Watch is selected, the type of error becomes “Watch” and the “From” column changes to “vue.”

Figure 71 Example of Waived Result

Detailed Information in Tooltips

For error classification from the cPYDB, the full path of the cPYDB is displayed in the tooltip, as shown in Figure 72.

Figure 72 Error Classification From cPYDB

For error classification from VUE, the user ID and timestamp are displayed in the tooltip, as shown in Figure 73.

Figure 73 Error Classification From VUE

Adjusting Column Location

Figure 74 shows how to quickly adjust the column location for the Violation Browser.

Figure 74 Example of Adjusting the Violation Browser Columns

Switching to Non-Detail Mode

Figure 75 shows RMB on the error list and quickly switches from detail mode to non-detail mode.

Figure 75 Example of Switching From Detail Mode to Non-Detail Mode

Hiding Function Violations

Figure 76 shows RMB on the title and quickly hides the function violations.

Figure 76 Example of Hiding Function Violations

Iterative Highlighting

Figure 77 shows iterative highlighting in the detail and non-detail modes.

Figure 77 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 78 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 78 Loading a soft_check layer2 Violation

Figure 79 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 79 Loading a soft_check layer1 Violation

Figure 80 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 80 Cumulative Layer Highlight Zoom Control

Feedback

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 81.

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 81 LVS Errors Tab

When you select an equivalence cell pair, shown in Figure 82, diagnostic messages and equivalence errors are displayed in the Summary view. You can expand or collapse all display levels in this pane.

Figure 82 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 83, 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 83 Summary Tab

Device Grouping

The color of the device grouping highlight is set by the Device Highlight color selection. The groups of devices are provided in the layout window with the color displayed. See Figure 84.

Figure 84 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.

The instance pin (C in Figure 84), device pin, and cell port can be highlighted in the layout editor.

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 85 for more information.

Feedback

Figure 85 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 86. 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)

Feedback

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 86.

Figure 86 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 87, the More Actions list in the Highlight List pane displays a list of actions you can perform on LVS errors.

Figure 87 Highlight List Options for LVS Errors

Table 9 describes the actions that are available from the More Actions list.

Table 9 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.

Feedback

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 88.

Figure 88 Details Tab for LVS Errors

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 89. This can be very useful when working with isolated untexted shorts or device connection issues.

Figure 89 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 90.

Figure 90 Layout Netlist Pane

Next, and as shown in Figure 91, 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 91 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.

Feedback

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 92, 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 92 Zoom Context Section

Feedback

Select Port Net

The Select Port Net utility, shown in Figure 93 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 93 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.

Feedback

To use this utility, click the Select Port Net button in the Highlight pane, shown in Figure 94. Click a device within the Summary tab or Netlist browser.

Figure 94 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.

Feedback

Netlist Visualizer Options

The Netlist Visualizer Options dialog box is shown in Figure 95.

Figure 95 Netlist Visualizer Options

The following options are available in the Netlist Visualizer Options 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.

Feedback

- **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 96.

Feedback

Figure 96 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 97.

Figure 97 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 98.

Figure 98 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 99.

Figure 99 Drawing a Net

Rendered Color for Matched and Unmatched Objects

Matched devices and nets are rendered in the canvas in gray, shown in Figure 100. Unmatched devices and nets are rendered in the canvas in red.

Figure 100 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 101.

Figure 101 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 12 and Table 13.

After you select the objects, a menu is available for actions on the selected objects, described in Table 10.

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 102 and Figure 103, 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 earlier in Figure 86.

In Figure 102, the nets are highlighted in yellow and the devices are highlighted in green in the canvas and text netlist.

Figure 102 Highlighted Objects

In Figure 103, the nets are highlighted in magenta and the devices are highlighted in red in the canvas and text netlist.

Figure 103 Highlighted Objects

Figure 104 shows the layout window with multiple nets and devices highlighted.

Figure 104 Multiple Nets and Devices Highlighted

Progress Bar

The progress bar, shown in Figure 105, provides information about the progress of the drawing task.

Figure 105 Progress Bar

Feedback

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 106.

Figure 106 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 107.

Figure 107 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 108.

Figure 108 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 108. 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.

Feedback

Figure 109 Unmatched Pin

By clearing the yellow highlight, shown in Figure 109, 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 110.

Figure 110 Add Fanout Selection

After selecting a pin, net, or port, you can right-click and choose Add Fanout, shown in Figure 111. 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 Graphical Canvas Menu section for more information about the functions you can select.

Figure 111 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 112 Equivalence List Diagnostics

As an example, Figure 112 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.

Graphical Canvas Menu

When you right-click a selected object, a menu is displayed. The functions of the menu depend on the objects you select, shown in Figure 113 and Figure 114.

Figure 113 Menu Selections

Feedback

Figure 114 Alternative Menu Selections

Table 10 shows the commands you can choose to change your design.

Table 10 Graphical Canvas 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

Feedback

Table 10 Graphical Canvas Menu Selections (Continued)

Command	Description	Applies to
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 11 describes the commands you can use to zoom and pan.

Table 11 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 12 lists the bindkey settings for the mouse.

Table 12 Bindkey Settings for the Mouse

	Left button	Middle button	Scroll wheel	Scroll wheel
	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 13 lists the bindkey settings for the keyboard.

Table 13 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 for visually debugging a runset. It displays all of the intermediate layers created during the run to guide you to the source of the problem.

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 Layer Debugger from the File menu, as shown in Figure 115.

Figure 115 Layer Debugger Command on the File Menu

After you choose Layer Debugger 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.

Feedback

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 116.

Figure 116 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.

Feedback

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 Runset 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.

   To select a debug object, you can

4. Select objects from the available debug objects pane and move them to the selected pane using the Right Arrow button between the two panes.
5. Select debug objects from within the runset view, shown in Figure 117.
6. Manually enter a debug object into this list. This feature is disabled and the value is ignored.
7. From the Execution tab, click Execute.

Figure 117 Debugging Objects

Feedback

The Runset dialog box, shown in Figure 118, 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 118 Runset Dialog Box

There are six 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.

- Connect Database (CDB)

  You can select a connect database to debug. When selected, a utility is created in the output that allows you how to debug a connect sequence to find issues with connected layers. See the Connect Debugger section for more information.

Feedback

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, as shown in Figure 119, displays 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 Validator WorkBench, 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 119 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, as shown in Figure 120.

Figure 120 Non-Empty and Empty Layers

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, as shown in Figure 121.

Figure 121 Rollover Help

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, as shown in Figure 122.

Figure 122 Rollover Help

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.

Connect Debugger

The VUE Connect Debugger, which is part of the Layer Debugger, allows you to highlight and debug nets in a connect database.

Feedback

Setup

In the Layer Debugger setup on the File menu, the connect database is available as a new object type for selection. The Connect Debugger is enabled only when a CDB object type is selected from the Debug Objects tab or from the runset view, as shown in Figure 123. Only one CDB can be debugged at a time.

Figure 123 CDB Object Type From Debug Objects

Loading the Results

After the IC Validator run completes, set VUE to layer debugger mode and load the results. From the layer debugger results, a connect database for debugging is selectable, as shown in Figure 124.

Figure 124 CDB Object Type From Debug Objects

VUE creates a new Connect Debugger tab upon selection.

Connect Debugger Layout

The Connect Debugger is organized into three sections: the list of nets being debugged, the full connect sequence being debugged, and a path finding utility. See Figure 125.

Figure 125 Connect Debugger Tab

Debugging a Connect Sequence

You can select a net to debug either by: selecting by name or by selecting a point in the layout.

Figure 126 shows a list of all texted nets is available for selection when you select a net to debug either by name or from the layout. A reduced list matches the expression in the search bar.

Figure 126 Selecting a Net to Debug by Name

Figure 127 shows how to select a net from the layout. First, click the probe point icon on the toolbar, then select the point in the layout of interest. The Connect Debugger provides a list of nets under that point to select.

Figure 127 Selecting a Net to Debug From the Layout

Feedback

Working With Nets

When you highlight a net, you can choose to have the highlight overlayed with all of the connect layers, or just show the highlight, as shown in Figure 128.

Individual layers in the connect can be filtered from the net highlight by selecting the layer in the connect layer list.

Figure 128 Working With Nets

When showing all connect layers, you can omit bulk layers. Uncheck the Show bulk layers check box in the Connect Debugger options, as shown in Figure 129.

Figure 129 Show Bulk Layers

Feedback

Path Finding

To find a path between two points that you might not to be connected, select the path finding button for that net. This activates the path finding widget. Select the starting and stopping points on the net from the layout and then select Start Net Trace to see that path, as shown in Figure 130, Figure 131, and Figure 132.

Figure 130 Path Finding

Figure 131 Path Finding Start the Trace

Figure 132 Path Finding Highlighted Path

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.

Feedback

Loading Short Static Output

- Figure 133 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 133 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 134 for details on the text-based Short Finder flow.

Figure 134 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 shorted texts. Polygons can be highlighted and zoomed, and you can use different colors to show pruning of polygons that are properly connected.

The Short Finder tab contains a list of all of the nets that can be debugged with the interactive Short Finder, as shown in the following figure.

The Short List table shows each cell and the text that is part of the short. Power and ground nets are given a special column to help identify these crucial nets. All other text is placed in the Nets column. Other basic information about each short is available:

- Layer # is the number of unique layers in the shorted net. A large number means that the net traverses through many layers in the connectivity stack and can be very complex.
- Cell # is the total number of unique cells the shorting path traverses. A large number indicates a deeply hierarchical short or one that traverses large sections of a design.
- Polygon # is the total number of polygons that are part of the shorted net.
- Text # is the total number of text labels applied to the short net.

- ID column is a unique number for each short. The number is consists of two parts. The first number represents the text net command that found a text short. The first text net is a 1, and the second text net has a 2. The next number is a unique number for each short found during that particular text net.

You can sort the short list in each of these columns by clicking on the column header. The Search Keyword field allows you to search for specific net names or cell names. All columns are searched for the specified keyword.

Note:

The Text and LVS buttons, as shown in the following figure switch between the text-based Short Finder (Text) and the LVS results-based Short Finder. When you select the short list, it is updated to show only the shorts found for the selected type.

After you select a short, double-click the row and a new short path window is generated, as shown in the following figure.

The Highlight Path dialog box contains all of the information needed for the currently loaded short. You can minimize the main VUE window to help optimize screen usage. To do this, click the Hide VUE button at the bottom left of the dialog box.

The Highlight Path dialog box is separated into two major sections.

- The Discovered Path section shows a list of each cell and layer in the shorting path.
- The Highlight Filter provides quick access to different highlight controls.

The Highlight Path options are:

- Physical Layer: Defines a unique color for each layer in the shorting path. All polygons on metal1 are the same color regardless of which cell or text label.
- Net Type: Defines a unique color for any polygon that is determined to be power, ground, or signal regardless of layer or cell.
- Net (default): Defines a unique color for each text in the short, and all unknown polygons (not directly labeled) are given a third color.
- Cell: Defines a unique color for each cell in the path. All of the polygons in a cell are given the same color regardless of which layer or text label.

You can manipulate the drawn short path in several ways, and you can turn highlight layers on or off to help identify areas that might indicate a short.

If a row (or multiple rows) in the Discovered Path is selected, when you rehighlight the path, the selected polygons are given a unique color in the path. This makes it easy to identify those polygons in the short path.

To enable or disable a specific row highlight, enable or clear the check mark next to the row.

The highlight for row two is now cleared. All other highlights are still displayed.

Right-click a function in the Discovered Path widget to manage the check boxes more efficiently.

You can right-click to zoom to the selected polygons as well. This can be very useful when the path traverses large sections of the design.

The path highlight can also be manipulated with the Highlight Filters. By default, the entire path is displayed, but you can disable specific layers or cells by unselecting the choices in the appropriate tab.

All ME6 layers are disabled in the short path. These filters can be applied regardless of which highlight scheme is used.

Additional options change how the shorting path is highlighted. These options are available when you select Short Finder, as shown in the following figure.

Here, you can add columns to the discovered path table.

- Coordinate adds the polygon bounding box.
- Text adds any labels attached during the IC Validator run.
- Assigned nets add a column for any user-assigned labels while using the Short Finder.

Path Inside Polygon:

When you click Highlight, the shorted path is visible in the layout editor, as shown in Figure 135. The red line is the short path. VUE connects the center point of each polygon in the Path Discovered pane to its neighbor to form this path. See the Highlight Path dialog box later in this chapter for a description of the colors in the toolbar that are used to highlight the short path.

Figure 135 Short Path With Centerline Path

By default, the Short Finder clips polygons in the path to only the parts connecting the path-into the polygon and path-out to the next polygon in the path. This optimizes the highlight and allows for a more intuitive path. To see the entire polygon, select Whole polygons in the Short Finder options.

To add a label along the path, select the polygon or polygons in the Discovered path list that you want to add a label to. Right-click to select the label to assign to those polygons. Their highlight color changes to match that of the assigned net, if using the Net highlight scheme

Additional controls over what path is chosen are available via the Net Terminal widget.

When selected, the following Select Short Net dialog box appears:

This interface allows you to define your own starting and stopping points for the path. The Net Tree defaults to Auto Select the path. If Auto Select is unchecked, right-click in the Net Tree to set the “start from” and “end at” points for the path. The Net Tree is organized by net label, layer, and then coordinate.

You can filter the labels in the Net Tree using the Layer Priority Settings or Area Settings. Any text labels in the Net Tree that do not meet the criteria set are disabled.

The Net Info tab in the Select Short Net dialog box guides the Short Finder on where it is allowed to find the paths.

When a Cell or Layer is unselected from the lists, that tells the interactive Short Finder to avoid finding a path through the specified cell or layer. If no path is found with the filter applied, the Discovered Path widget remains empty. Add back the elements that were filtered to find the minimum set to get a shorting path.

Interacting With a Short Path

The IC Validator interactive Short Finder allows you to provide information into the short. You can remove polygons from the path, or disconnect them. Any changes made are kept as part of the net changes. All of the shorts can be found without rerunning IC Validator. This greatly reduces the turnaround time in debugging your design.

You can attach virtual labels to provide references when highlighting. All of the information is annotated into the short database for the path. When information is provided that might change the short, you can reanalyze the shorting net to determine if there are any other shorting paths.

To manipulate the short path, use the polygon action buttons in the Highlight Path ribbon.

Table 14 Polygon Actions

Action	Description
Disconnect	The selected polygon is disconnected from the previous or next polygon.
Remove	The selected polygon is removed from the short path.
Undo	All changes for the selected polygons are undone.
Verify	Any changes are applied to the short out database. The shorting net is then reanalyzed and, if another path is found, it is displayed. Changes are added to the Net Change pane in the short finder tab.

Note:

All changes are virtual. The interactive Short Finder does not alter the physical design.

In the Net Change table, you can double-click a row to highlight the polygons involved in the changes.

If there is not another path short found with the applied changes, the Highlight Path GUI remains blank and the row for that short in the Short List is green.

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.

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 136.

Figure 136 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 137.

Figure 137 All LVS Shorts Listed by Text and LVS Options

ID	Cell	PWR	GND	Nets	Layer#	Cell#	Polygon#	Text#
0 - 1	DESIGN1_DRC1:DESIGN1_DRC1			N_1024...	13	9	113	3
0 - 10	DESIGN1_DRC1:DESIGN1_DRC1	VDD		XTPSRA...	11	13	67	2
0 - 11	DESIGN1_DRC1:DESIGN1_DRC1	VDD		XTPSRA...	11	13	57	2
0 - 12	DESIGN1_DRC1:DESIGN1_DRC1			N_3398...	13	6	52	2
0 - 13	DESIGN1_DRC1:DESIGN1_DRC1			N_1852...	15	17	152	5
0 - 14	DESIGN1_DRC1:DESIGN1_DRC1			N_1049...	13	8	42	3
0 - 2	DESIGN1_DRC1:DESIGN1_DRC1	VDD	VSS	N_1034...	21	300	16975	159
0 - 3	DESIGN1_DRC1:DESIGN1_DRC1	VDD		XTPSRA...	11	13	57	2
0 - 4	DESIGN1_DRC1:DESIGN1_DRC1	VDD	VSS	XTPSRA...	11	11	37	4
0 - 5	DESIGN1_DRC1:DESIGN1_DRC1	VDD	VSS	XTPSRA...	11	11	37	3
0 - 6	DESIGN1_DRC1:DESIGN1_DRC1		VSS	XTPSRA...	11	11	35	6

The Highlight Path GUI 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 device or instance pins to use at the starting and stopping points for the path.

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 138 LVS-Based Short Finder Output

5

Introduction to IC Validator Error Heat Map

The IC Validator Error Heat Map is a graphical interface used to visualize error distributions.

For more information, see the following sections:

- Executing the Heat Map
- Heat Map Interface
- Overlaying the Error Heat Map With the Layout Editor
- Highlighting Error Markers From the Heat Map
- Highlighting Cell Outlines
- Selecting Related Violations
- Searching and Filtering Violations
- Top-Cell Only Heat Maps
- Heat Map Grid Statistics
- Saving the Heat Map to a Disk
- Restrictions and Limitations
- Licenses

Feedback

Executing the Heat Map

Ensure that the *ICV_INSTALL_DIRECTORY* installation directory is set properly.

Add the `-hm`, and optionally, the `-hm_cell` command-line option to your execution command line.

To generate only the top-cell heat map

```
% icv -hm other DRC options
```

To generate the subcell heat map in addition to the top-cell heat map

```
% icv -hm -hm_cell sub_cell_name other DRC options
```

Note:

The IC Validator Error Heat Map is automatically invoked when running IC Validator Explorer.

Note:

The `-hm` command-line option must also be provided for any heat maps to be generated.

After the Heat Map information is generated, VUE automatically reads it and creates the Heat Map tab in the VUE GUI, as shown in Figure 139.

Figure 139 Heat Map Tab

Heat Map Interface

The IC Validator Error Heat Map consists of five panes, which are described in the following sections.

Violation Heat Map

The Violation Heat Map shows the heat map errors for the selected violations. The Violation Heat Map toolbar is displayed in Figure 140.

Feedback

Figure 140 Violation Heat Map Toolbar

The Violation Heat Map toolbar buttons are defined in Table 15.

Table 15 Violation Heat Map Toolbar Buttons

Toolbar button	Description
Zoom Fit	Zooms to the chip extent
Zoom In	Zooms in to the Heat Map
Zoom Out	Zooms out of the Heat Map
Overlay Heat Map	Draws Heat Map grids on connected layout editor
Highlight Top-Cell Errors	Highlights top-cell errors in the current zoom window
Select Related Violations	Selects violations in the current zoom window
Match View	Changes zoom of layout editor to zoom window of Heat Map
Information	Toggle button for grid statistics (when zoomed in) and for cell name (when Show Cell Outlines is active)
Show Cell Outlines	Displays cell outlines in violation heat map window. Level depth can be filtered in the argument box.
Help	Launches the Help menu

When connected to a layout editor, the Violation Heat Map window displays a list of right-click options, as shown in Figure 141.

Figure 141 Violation Heat Map Right-Click Options

Table 16 shows the different right-click options that are available in the Violation Heat Map window. Use these options to highlight and draw error markers and heat maps in the connected layout editor.

Table 16 Violation Heat Map Right-Click Menu Options in the Connected Layout Editor

Menu options	Description
Highlight top-cell errors in grid	Draws top-cell errors that are in the Heat Map grid
Highlight top-cell errors in current window	Draws top-cell errors that are in the current Heat Map zoom window
Overlay Heat Map	Draws the colored heat map

Feedback

Violation Category

The Violation Category window is the interface used for viewing all predefined (IC Validator or user) violation categories.

All categories are self-contained, that is, all errors are present in each individual category and divided into subsections. Table 18 shows the different violation categories and their descriptions.

Table 17 Violations Categories

Category	Subcategory	Description
All DRC errors		All DRC errors are available from the runset.
Explorer categorized errors		All DRC errors are available from the runset and Explorer methodology warnings.
	DRC Warnings: Explorer Methodology	Explorer methodology warnings aimed to flag potential design or performance issues.
	DRC Errors: Fill Overlap Diagnostic	DRC errors that are diagnosed as fill overlap issues
	DRC Errors: Explorer	DRC errors available from Explorer checks (runset selected basic rules)
	DRC Errors: Non-Explorer	DRC errors available from non-Explorer checks (runset rules after Explorer selection)

Note:

Categories and subcategories can vary based on run data.

Violation Selection

The Violation Selection window is the interface to select the violations used to draw the Heat Map. All violations selected contribute to the drawn Heat Map.

Table 18 shows the different columns available for the Violation Selection window.

Table 18 Violation Heat Map Selection Options

Menu options	Description
Violation/Function	Runset violation comment.
Hierarchical Errors	Hierarchical error count for the violation.

Feedback

Table 18 Violation Heat Map Selection Options (Continued)

Menu options	Description
Flat Error Count	Flat error count for the violation.

Color Definitions Window

The Color Definitions window displays the Heat Map color definitions ranging from red (maximum error count) to dark blue (minimum error count). Color definitions adjust dynamically based on the selection or number of errors. You can select or unselect individual color bins to be displayed in the Violation Heat Map window.

An additional black color represents regions with 0 errors by default. You can customize the upper threshold of the black color to be greater than 0 by double-clicking on 0 and entering the desired error count, as shown in Figure 142.

Figure 142 Color Definitions Window

Sliding Bar

This sliding bar adjusts the size of Heat Map grids. Slide the bar to left to see larger grids (coarser) and to the right to see smaller grids (finer). Finer grids are more appropriate at closer zoom levels. The number of levels of detail is determined dynamically based on the size of the design; larger designs have more levels.

Overlaying the Error Heat Map With the Layout Editor

The Error Heat Map does not have the ability to display design shapes or polygons. To view both of the Heat Maps and design shapes, overlay the Heat Map with the connected layout editor by following these steps:

1. Connect VUE to a compatible layout editor. See Chapter 1, Introduction to IC Validator VUE for compatible layout editors.
2. Select Overlay Heat Map from the Options list in the Violation Heat Map window. You can change the highlight fill pattern by using the pull-down next to the Heat Map (button/action/option).

Figure 143 shows the IC Validator WorkBench layout editor on the left overlayed with the Heat Map.

Figure 143 VUE Connected to IC Validator WorkBench With Overlay Heat Map

Highlighting Error Markers From the Heat Map

For ease of use, highlight the top-cell error markers directly from the Heat Map tab by following these steps:

1. Connect VUE to a compatible layout editor. See Chapter 1, Introduction to IC Validator VUE for compatible layout editors.
2. Right-click to open the Violation Heat Map list. The following top-cell error highlighting options appear:
3. Highlight top-cell errors in grid (only when right-clicked on a Heat Map grid). This option highlights the top-cell errors for violations in the selected grid, as shown in Figure 144.

Figure 144 VUE Connected to IC Validator WorkBench With Highlighted Errors in Selected Grid

4. Highlight top-cell errors in current window. This option highlights top-cell errors for violations in the current zoomed Violation Heat Map window, as shown in Figure 145.

Figure 145 VUE Connected to IC Validator WorkBench With Highlighted Errors in Current Window

The Highlight top-cell Errors option from the Heat Map tab is used only for the top-cell errors. If there are no top-cell errors in the given window, a notification message appears, as shown in Figure 146.

Figure 146 Notification Message When No Top-Cell Errors are Available to Highlight

Highlighting Cell Outlines

You can highlight cell outlines using the Show Cell Outlines option in the Violation Heat Map toolbar. Use this feature to analyze cells with errors and interface errors between cell boundaries. You can filter the depth level of cell outlines to be highlighted in the argument box. Cell name information can be toggled with the Information button in the toolbar. See Figure 148.

Figure 147 Show Cell Outlines

Selecting Related Violations

You can select all related violations in a zoomed window by following these steps:

1. Zoom to a desired window and select the Show related violations option in the Violation Heat Map window.
2. All violations in the zoomed window are selected in the Violation selection window, as shown in Figure 148.

Figure 148 Select Related Violation Option to Select All Violations in Current Zoomed Window

Feedback

Searching and Filtering Violations

You can quickly search and filter violations by doing the following:

- Click the search menu to select the search criteria. Two options are available:
  - Search Violation by Name. Performs a search based on the violation name. You can use the “*” wildcard character.
  - Search Violation by Layer. Performs a search based on the layers involved in a violation. You can use multiple layers (separated by “,”) and the “*” wildcard character, as shown in Figure 149.

Figure 149 Violation Filtering by Name or by Layer

Top-Cell Only Heat Maps

If you are interested in the Heat Map contribution from only the top-cell errors, you can view them by following these steps:

1. From the VUE Tools list, select Heat Map and check the Top-cell Errors (Top-cell only) option. There is an additional filter option to remove zero-count violations, that is, violations with a zero contribution to the top-cell only error count.
2. Click Apply to update the Heat Map tab with the top-cell only Heat Map, that is, the error distribution and hotspot contribution from only the top-cell errors, as shown in Figure 150.

Figure 150 Top-Cell Only Heat Maps

Heat Map Grid Statistics

Use the Information option in the Violation Heat Map toolbar to see heat map grids statistics when zoomed into the grids, as shown in Figure 151.

Figure 151 Heat Map Grid Statistics

Saving the Heat Map to a Disk

The heat map provides a visualization of the status of the design and the geographical distribution of DRC errors. Therefore, it is useful to capture an image of the heat map for reporting purposes. After the run completes, you can save the heat map image using the `-hmSaveImage` command-line option.

```
% icv_vue -load path_to_vue_file -hmSaveImage
```

You must use the `-load` command-line option to indicate the .vue file when using `-hmSaveImage[=subcell1,subcell2,…]`.

This command-line option accepts an optional, comma-separated list of subcells.

By default, VUE saves the heat map image of the top cell. When a subcell list is specified, heat map images for those subcells are saved in addition to the top cell.

A subcell heat map image is skipped if the subcell was not generated using the `-hm_cell` command-line option during the run, if that specified subcell did not exist in the design, or if the specified subcell was exploded during the run. The standard output notifies you when such cases occur.

The heat map images are saved as hm_*cell_name*.png in the current working directory.

The image saved to a disk shows the cell name, design path relative to the .vue file location, the heat map, and the color gradient legend, as shown in Figure 152.

Figure 152 Saved Heat Map Images

Restrictions and Limitations

The Heat Map shows a flattened representation of all errors in your design. However, error markers and highlights are available only for some of the errors (limited by the `error_limit_per_check` or `error_sampling_per_check` arguments of the `error_options()` function).

Feedback

Licenses

The Violation Heat Map requires the following licenses:

This feature requires Elite or Base IC Validator license: ICValidator-Manager.

See the “Licensing and Resource Requirements” chapter in the *IC Validator User Guide* for more information.