

Incisive vManager User Guide

Product Version 15.20

February 2016

© 2016 Cadence Design Systems, Inc. All rights reserved.

Printed in the United States of America.

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

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

This product contains third party software. Refer to *<install_dir>/doc/thirdpartyinfo/IMCthirdpartyinfo.txt* for details on copyright and licensing terms.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

All other trademarks are the property of their respective holders.

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

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

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

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

Contents

1

Getting Started with vManager 11

<u>1.1 Launching vManager Client</u>	11
<u>1.1.1 Launching the vManager RCL Client</u>	12
<u>1.1.2 Launching the vManager Full Client</u>	13
<u>1.2 vManager Interface</u>	14
<u>1.2.1 Menu Bar</u>	15
<u>1.2.2 Activity Centers</u>	25
<u>1.2.3 Toolbar</u>	26
<u>1.2.4 View Area</u>	26
<u>1.2.5 Messages Log</u>	27
<u>1.3 Tasks Common to All Centers</u>	28
<u>1.3.1 Managing Table Columns</u>	28
<u>1.3.2 Sorting Data in Tables</u>	31
<u>1.3.3 Searching Data in Tables</u>	31
<u>1.3.4 Filtering Table Data</u>	33
<u>1.3.5 Detaching Panes</u>	39
<u>1.3.6 Hiding Panes</u>	41
<u>1.3.7 Defining and Organizing Views</u>	43
<u>1.3.8 Exporting Table Data (Sessions, Runs, Failures, Tests, vPlan, Metrics, Snapshots) to a CSV File</u>	49
<u>1.3.9 Generating Summary Report</u>	51
<u>1.3.10 Configuring Configurable Items</u>	54
<u>1.3.11 User-Defined Actions</u>	71

2

Loading and Viewing Sessions 77

<u>2.1 Regression Center Interface</u>	78
<u>2.1.1 Menu Bar</u>	79

Incisive vManager User Guide

<u>2.1.2 Toolbars</u>	80
<u>2.1.3 Sessions Pane</u>	85
<u>2.1.4 Attributes Pane</u>	86
<u>2.2 Loading Sessions</u>	87
<u>2.2.1 Launching a Session (VSIF)</u>	87
<u>2.2.2 Importing Sessions</u>	89
<u>2.3 Exporting Sessions</u>	91
<u>2.4 Exporting Merged Coverage of Sessions</u>	92
<u>2.5 Collecting Runs</u>	93
<u>2.6 Refreshing Sessions</u>	96
<u>2.6.1 Configuring Session Refresh Options</u>	96
<u>2.7 Opening Session Directory in Terminal</u>	97
<u>2.8 Viewing Session Information</u>	98
<u>2.9 Stopping Sessions</u>	100
<u>2.10 Stopping Session Based on Conditions</u>	100
<u>2.11 Suspend / Resume Sessions</u>	104
<u>2.12 Deleting Sessions</u>	105
<u>2.13 Relocating Session</u>	105
<u>2.14 Handling Inaccessible Sessions</u>	106
<u>2.15 Editing Session Details</u>	108
<u>2.15.1 Edit each</u>	108
<u>2.15.2 Edit all at once</u>	109
<u>2.15.3 History</u>	109
<u>2.16 Associating Merged Coverage with Sessions</u>	110
<u>2.17 Creating Compact Sessions</u>	111
<u>2.18 Rollup Attributes to Session Level</u>	117
<u>2.18.1 Write a Calculation Script File</u>	118
<u>2.18.2 Define User-defined Attributes in the CSV File</u>	120
<u>2.18.3 Import the CSV File</u>	121
<u>2.18.4 Recalculate UDA</u>	122
<u>2.19 Generating Charts</u>	123
<u>2.20 Taking Snapshots</u>	125
<u>2.21 Generating Sessions Report</u>	127
<u>2.22 Invoking Analysis Center</u>	131

3

Analyzing Runs and Failures	133
3.1 Launching Analysis Center	133
3.2 Analysis Center Interface	134
3.2.1 Menu Bar	135
3.2.2 Toolbars	137
3.2.3 Analysis Navigation Panel	152
3.2.4 View Area	154
3.3 Analyzing Runs	154
3.3.1 Navigate Through Various Test Groups	156
3.3.2 Launch Separate Runs Page for Analysis	156
3.3.3 View Runs within the Selected Test or Test Group	157
3.3.4 Launch SimVision Waveform Window for Analysis	157
3.3.5 View Errors, Warnings, and Properties Associated with Selected Item	158
3.3.6 View Attributes and Logs Associated with the Selected Item	160
3.3.7 Generate Report on Runs Page	162
3.4 Analyzing Failures	165
3.4.1 Available Views for Failure Analysis	166
3.4.2 Generate Report on Failure Page	170
3.5 Analyzing Formal Properties	172
3.5.1 Launching the Formal Properties Analysis Page	173
3.5.2 Properties Views	174
3.5.3 Report on Formal Properties Page	177
3.6 Grouping Runs	180
3.6.1 Grouping by One Attribute	180
3.6.2 Grouping by Multiple Attributes	182
3.6.3 Ungrouping Runs	184
3.7 Filtering Runs	184
3.8 Ranking Runs	184
3.8.1 Rank runs	185
3.8.2 Report on Rank Page	190
3.8.3 Enable Parallel Ranking	192
3.9 Correlating Runs	195
3.9.1 Correlate Runs	196
3.9.2 Report on Correlation Page	200

<u>3.10 Generating Tests Report</u>	204
<u>3.11 Integrating vManager with Indago Debug Analyzer</u>	209
<u>3.11.1 Opening INDAGO GUI from vManager on Exact Location of Error</u>	209
<u>3.11.2 Setup Regression to Automatically Record IDA Data in Rerun of Failed Runs</u>	.
215	

4

Analyzing Metrics 217

<u>4.1 Launching Metrics Analysis From Regression Center</u>	217
<u>4.2 Launching the Metrics Analysis From Analysis Center</u>	218
<u>4.3 Metrics Summary Page Overview</u>	220
<u>4.3.1 Verification Hierarchy Pane</u>	220
<u>4.3.2 List tabs Pane</u>	225
<u>4.3.3 Details Pane</u>	226
<u>4.4 Analyzing Code Coverage</u>	229
<u>4.5 Analyzing FSM Coverage</u>	229
<u>4.6 Analyzing Functional Coverage</u>	229
<u>4.7 Analyzing Coverage Coming from Palladium</u>	235
<u>4.8 Analyzing Coverage Coming from JasperGold</u>	237
<u>4.9 Refining Metrics Data</u>	238
<u>4.10 Coverage Grading</u>	239
<u>4.10.1 Grade Calculation in vManager</u>	241
<u>4.11 Generating Metrics Report</u>	249

5

Analyzing Verification Plans 251

<u>5.1 Loading Verification Plans</u>	251
<u>5.1.1 Loading vPlan from Regression Center</u>	252
<u>5.1.2 Loading vPlan from Analysis Center</u>	253
<u>5.2 Creating a New vPlan</u>	258
<u>5.3 Editing vPlans</u>	259
<u>5.4 Organizing and Defining vPlan Views</u>	261
<u>5.5 Computing Grade of a Section</u>	264
<u>5.5.1 Section Grade Parameters</u>	264
<u>5.5.2 Cumulative Grade Calculation</u>	265

<u> 5.5.3 Goal-Relative Grade Calculation</u>	266
<u> 5.5.4 Mapped Elements Grade Calculation</u>	267
<u> 5.5.5 Assertion Status Grade Calculation</u>	269
<u>5.6 vPlan Configuration Options</u>	270
<u>5.6.1 Specify the vPlan to be Used in the case of Synchronization Conflicts</u>	271
<u>5.6.2 Specify if Coverage from Failed Runs Must be Included While Calculating Grades</u>	273
<u>5.6.3 Specify How to Merge Multiple Coverage Models</u>	274
<u>5.6.4 Enable Ordering of Coverage Models According to Timestamp</u>	275
<u>5.6.5 Enable or Disable Loading of vPlan Refinements File Automatically</u>	276
<u>5.6.6 Metrics Port of a vPlan to Show Zero or NA for No Mapped Elements</u>	276
<u>5.6.7 Specify Default vPlan File</u>	278
<u>5.6.8 Enable or Disable Showing the vPlan Extended Tree</u>	278
<u>5.6.9 Enable or Disable Showing Section Numbers</u>	280
<u>5.6.10 Enable Showing of Unmapped Perspective in vPlan</u>	281
<u>5.7 Looking for a Section or Port</u>	283
<u>5.7.1 Lookup Section in Hierarchy</u>	283
<u>5.7.2 Lookup Parent Section in Hierarchy</u>	284
<u>5.7.3 Lookup Port in Hierarchy</u>	285
<u>5.7.4 Lookup Entity in Metric Hierarchy</u>	286
<u>5.8 Filtering Items in vPlan Hierarchy</u>	288
<u>5.9 Launching Detailed Analysis Page for vPlan Entity</u>	289
<u>5.10 Refining vPlan Items</u>	290
<u>5.10.1 Excluding Sections or Metrics Ports</u>	291
<u>5.10.2 Adding Comments at the Time of Exclusion</u>	292
<u>5.10.3 Editing User Comments</u>	295
<u>5.10.4 Hide Excluded Items</u>	296
<u>5.10.5 Un-Excluding Excluded Items</u>	297
<u>5.10.6 Refining vPlan Attributes</u>	298
<u>5.10.7 Revert vPlan Refinements</u>	301
<u>5.10.8 Saving Refinements</u>	301
<u>5.10.9 Loading Saved Refinements</u>	302
<u>5.10.10 Reusing Refinements File</u>	304
<u>5.11 Generating vPlan Reports</u>	309
<u>5.12 vPlan Visitor Interface</u>	316
<u>5.12.1 Using the vPlan visitor Interface</u>	316

<u>5.12.2 Extending a vPlan visitor Interface</u>	317
6	
Project Tracking	319
<u>6.1 Launching the Tracking Center</u>	319
<u>6.1.1 Menu Bar</u>	321
<u>6.1.2 Toolbar</u>	322
<u>6.1.3 View Area</u>	324
<u>6.2 Creating a Configuration</u>	325
<u>6.2.1 Editing a Configuration</u>	331
<u>6.2.2 Deleting a Configuration</u>	331
<u>6.2.3 Refreshing a Configuration</u>	331
<u>6.3 Taking Snapshots</u>	331
<u>6.3.1 Deleting Snapshots</u>	338
<u>6.3.2 Editing Snapshots</u>	338
<u>6.4 Cumulative Data Tracking</u>	340
<u>6.4.1 Viewing Cumulative Data</u>	340
<u>6.4.2 Adding Cumulative Attributes to Chart</u>	341
<u>6.4.3 Recalculating Cumulative Data</u>	342
<u>6.5 Generating and Editing Charts</u>	342
<u>6.5.1 Generating Charts (Basic)</u>	343
<u>6.5.2 Generating Charts (Advanced)</u>	346
<u>6.5.3 Generating Charts (Tracking Run Groups Automatically)</u>	349
<u>6.5.4 Editing Chart Attributes</u>	352
<u>6.5.5 Removing Attributes from Charts</u>	354
<u>6.5.6 Removing Charts</u>	355
<u>6.5.7 Renaming Charts</u>	356
<u>6.5.8 Exporting Charts to CSV File</u>	357
<u>6.6 Generating Tracking Reports</u>	358
<u>6.7 Converting Regular Chart to Triage or Stacked Chart</u>	360
A	
List of Attributes	365
<u>A.1 Attributes associated with Session</u>	365
<u>A.2 Attributes associated with Test or Tests Group</u>	371

Incisive vManager User Guide

<u>A.3 Attributes associated with Run</u>	372
<u>A.4 Attributes associated with Failure</u>	379
<u>A.5 Attributes associated with Formal Properties</u>	380
 B	
<u>Invocation and Command-Line Interface</u>	383
<u>B.1 Launching vManager</u>	383
<u>B.1.1 Valid Modes for Launching vManager</u>	383
<u>B.2 Batch-Mode Commands</u>	404
<u>B.2.1 General Commands</u>	404
<u>B.2.2 Metrics-related Commands</u>	409
<u>B.2.3 vManager General Commands</u>	430
<u>B.2.4 Context-related Commands</u>	475
 C	
<u>Known Gaps in vManager</u>	545
<u>C.1 Known Gaps in vManager in Comparison with IEM</u>	545
<u>C.1.1 GUI</u>	545
<u>C.1.2 Runner</u>	545
<u>C.1.3 Analysis (Runs and Coverage)</u>	546
<u>C.1.4 vPlan Limitations</u>	546
<u>C.1.5 Ranking limitations</u>	546
<u>C.1.6 API and CLI Compatibility</u>	546
<u>Index</u>	547

Incisive vManager User Guide

Getting Started with vManager

vManager facilitates the basic, every-day tasks of the verification process. It provides you with a robust and interactive GUI using which you can:

- Launch thousands of runs in a single session using a distributed resource manager
- Facilitate failure analysis by automatically extracting, filtering, and grouping key information from all log files in one or more sessions
- Launch the rerun of failed tests and manage failure scenarios
- Facilitate metric analysis by presenting data gathered by multiple agents (Incisive Simulator and Specman) in the same window

The vManager GUI is structured into various activity centers.

This chapter discusses the following topics:

- [Launching vManager Client](#) on page 11
- [vManager Interface](#) on page 14
- [Tasks Common to All Centers](#) on page 28

1.1 Launching vManager Client

You can launch vManager client in any of the following modes:

- *Regression Center Launcher* (RCL) client — Supports only the session management and metrics analysis functionality.
- *Full* client — Supports complete vManager functionality.

This section describes the following topics:

- [Launching the vManager RCL Client](#)
- [Launching the vManager Full Client](#)

1.1.1 Launching the vManager RCL Client

The *RCL* client mode supports only the session management and the metrics analysis functionality. The *RCL* client uses the same license as IMC, which is the `Affirma_sim_analysis_env` license.

The following command line launches vManager *RCL* client in GUI mode:

```
vmanager -regr
```

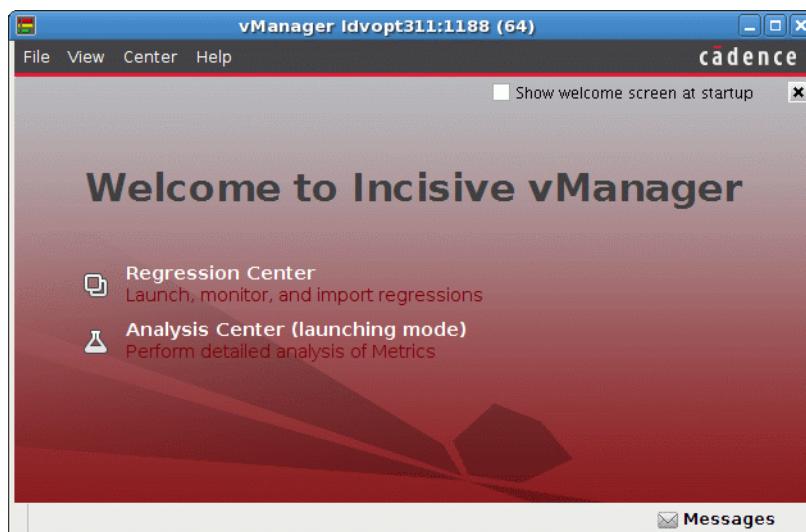
or

```
vmanager -regrcenter
```

Note: The `-regr` and `-regrcenter` options are the same (`-regr` is a shortcut). For more details on `-regr` and `-regrcenter` options, see [-regr or -regrcenter](#) on page 392.

The above commands launch vManager RCL client, and displays the *vManager* Welcome screen, as shown in [Figure 1-1](#) on page 12.

Figure 1-1 vManager Welcome Screen

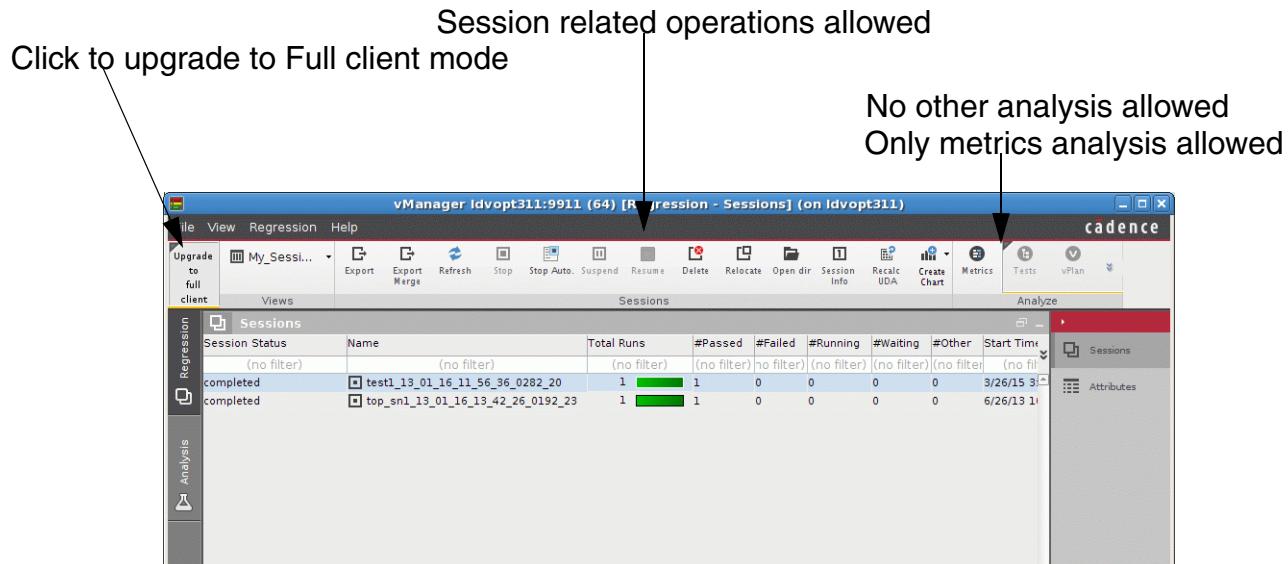


The *Welcome* screen shows the available activity centers in vManager. From the *Welcome* screen, you can launch only the *Regression* center. The *Analysis* center can be launched from the *Regression* center.

To proceed with the analysis tasks, click the *Regression Center* link on the Welcome screen.

[Figure 1-2](#) on page 13 shows the Regression Center in *RCL* client mode.

Figure 1-2 vManager Regression Center -- RCL Client Mode



In this mode, only the session management and metrics analysis functionality is available.

When you place the mouse over the buttons with unavailable functionality (for example, *vPlan* or *Tests* in the Analyze toolbar), a tooltip is shown to indicate that -- to enable this functionality, upgrade the license.

When in RCL client mode, you can upgrade the license and enable the *Full* client mode (which includes all the functionality) by clicking the *Upgrade to full client* button.

Note: When you click the *Upgrade to full client* button, the tool checks out for the *vmanager_client* license.

- If the license is available, then the *Affirma_sim_analysis_env* license is released and all the functionality of *Full* client is available.
- If the license is not available, an error is reported and the tool continues to run in *RCL* client mode.
- If the license is not available but the user launched the application with *-licqueue* option, then the application waits till the license becomes available.

1.1.2 Launching the vManager Full Client

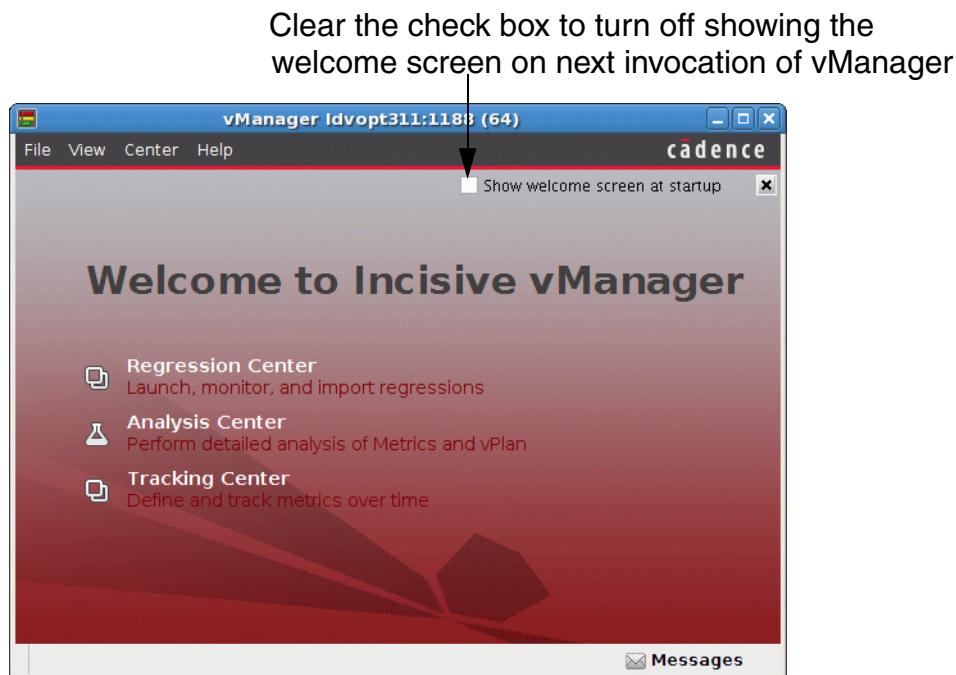
The *Full* client mode supports complete vManager functionality. It requires the *vmanager_client* license.

The following command line launches vManager *Full* client in GUI mode:

```
vmanager
```

The above command launches vManager *Full* client, and displays the *vManager* Welcome screen, as shown in [Figure 1-3](#) on page 14.

Figure 1-3 vManager Welcome Screen



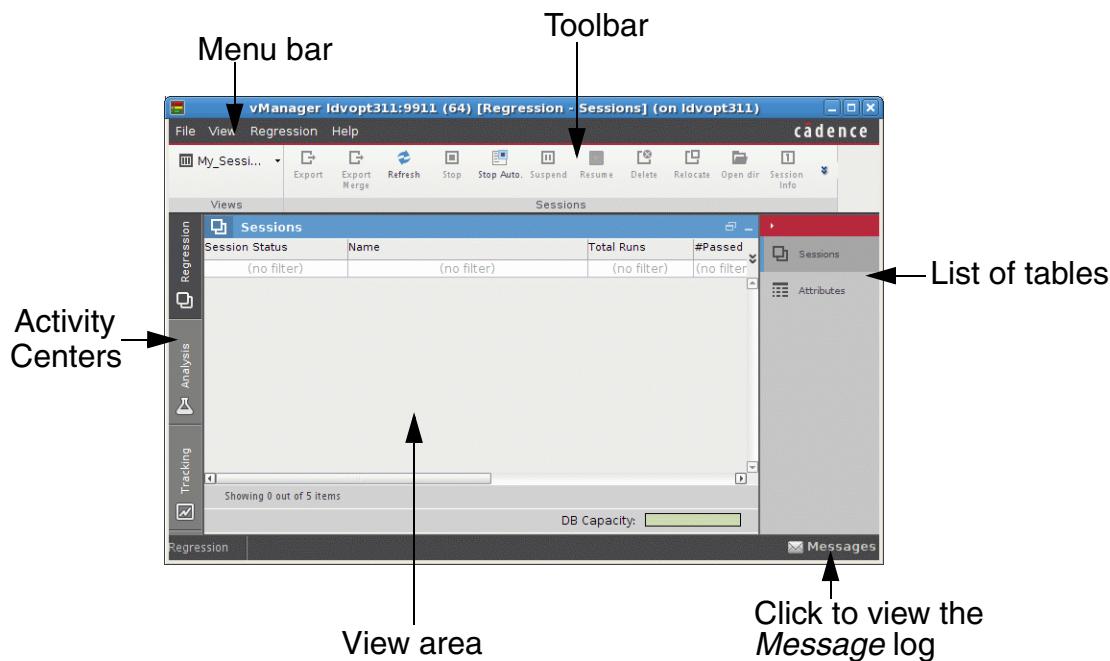
The *Welcome* screen shows the available activity centers in vManager. You can launch the *Regression* center and the *Tracking* center from the Welcome screen. The *Analysis* center can be launched from the *Regression* center.

Note: From the Welcome screen, you can also turn off the showing of the welcome screen on the next invocation of vManager. In case you turn off the welcome screen, then next time you launch vManager, by default, the *Regression* center will be shown.

1.2 vManager Interface

[Figure 1-4](#) on page 15 shows the Regression Center of vManager.

Figure 1-4 vManager Regression Center



The vManager window displayed in [Figure 1-4](#) on page 15 has the following components:

- **Menu Bar** — Shows the menu options related to currently active activity center. (In the above figure, options related to *Regression* center are shown)
- **Toolbar** — Shows options or buttons based on the currently selected activity center and pages. (In the above figure, toolbars related to *Regression* center are shown.)
- **Activity Centers** —Shows the activity centers available in vManager.
- **View Area** —Shows different panes depending on the currently selected activity center.
- **Messages Log** — Shows the list of warnings or errors that occurred during the current vManager invocation.

Note: Users that use KDE desktop, might have problems when minimizing a dialog. The application might become unresponsive and the dialog gets hidden. For more details, see:
https://bugs.kde.org/show_bug.cgi?id=277847 or
<http://permalink.gmane.org/gmane.comp.kde.devel.bugs/1498754>.

1.2.1 Menu Bar

[Figure 1-5](#) on page 16 shows the menu bar available in the *Regression* center.

Figure 1-5 Menu Bar



Menu options *File*, *View*, and *Help* are common to all activity centers. The *Regression* menu is related to the *Regression* center. When you are in *Analysis* center, the *Analysis* menu will be shown in place of *Regression* menu and when you are *Tracking* center, *Tracking* menu will show instead of *Regression* menu.

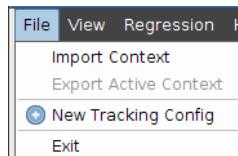
In this section, we will discuss the following menus because they apply to all the activity centers. The *Regression*, *Analysis*, and *Tracking* menus will be discussed in later chapters.

- [File](#)
- [View](#)
- [Help](#)

1.2.1.1 File

[Figure 1-6](#) on page 16 shows the *File* menu.

Figure 1-6 File Menu



From the *File* menu, you can select any of the following options:

- [Import Context](#)—To import a context from XML file.
- [Export Active Context](#)—To export an active context.
- [New Config](#)—To create a new configuration for tracking. For more details, see [Chapter 6, “Project Tracking.”](#)
- [Exit](#)—To exit the tool.

Import Context

vManager enables you to save and share context data, as required. You can export a context (which gets saved as an XML file) and when required, import the already saved XML file.

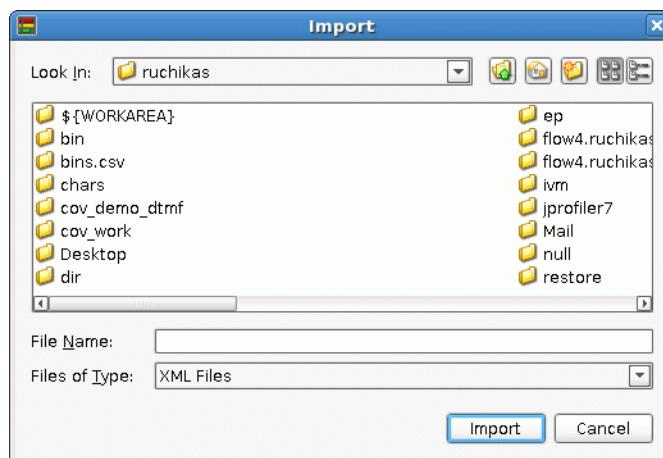
It saves a lot of time as you do not need to open all the pages again and you can start analysis immediately from the point where you left it. You can share the context with other users on the same server.

Note: A context must already be exported before you can import it. This section describes importing a context. For details on exporting a context, see [Export Active Context](#) on page 17.

To import a context:

1. Select *Import Context* from the *File* menu.
2. The *Import* dialog box appears, as shown in Figure 1-7 on page 17. Navigate to the location where the exported context (xml file) was saved. Select the xml file and click *Import*.

Figure 1-7 Import a Context



After you click import, the selected context is opened and the pages within the context are launched.

Note: At the time of launching vManager, you can also specify the context to be launched. For this, use the `-context` option as shown below:

```
vmanager -context <context_file> -local <dir> | -server <host:port>
```

For more details, see [-context <context_name>](#) on page 394.

Export Active Context

vManager enables you to save and share context data, as required. You can export a context (which gets saved as an XML file) and when required, import the already saved XML file.

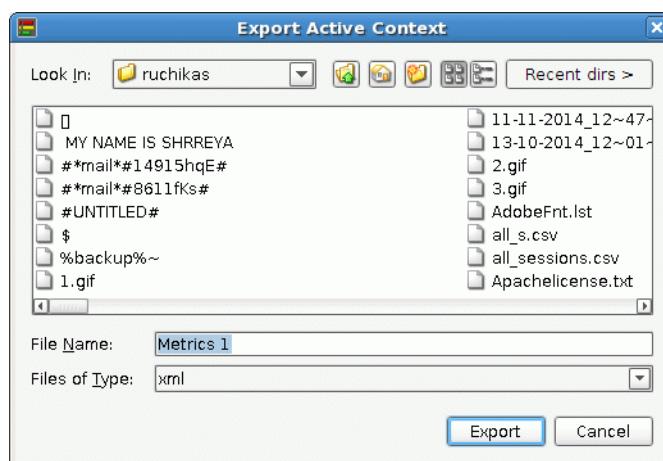
To export a context:

1. Select *Export Active Context* from the *File* menu.

Note: Alternatively, you can right-click the context in the *Navigation* pane, and select *Export*.

2. The *Export Active Context* dialog box appears, as shown in [Figure 1-8](#) on page 18. Navigate to the location where you want to save the exported context (xml file). Specify the name for the xml file in the *File Name* field and click *Export*. By default, the *File Name* field shows the current context name.

Figure 1-8 Export a Context



The exported context is then saved as an xml file. You can share this xml file with other users on the server. This xml file can be imported, as required. When the xml file is imported, it launches all the pages that were already open at the time of exporting the context.

For more details on importing a context, see [Import Context](#) on page 16.

1.2.1.2 View

The *View* menu contains the commands for enabling pop-ups, disabling pop-ups, organizing dialog choices, removing dialog choices, and so on. You can choose the following items from the *View* menu:

- [Toolbars](#)
- [Organize Message Popups](#)
- [Allow Showing All Message Popups](#)

- [Skip All Message Popups](#)
- [Organize Dialog Choices](#)
- [Allow Showing All Dialogs](#)
- [Auto Save Settings](#)
- [Save Settings](#)
- [Reset Settings](#)
- [Restore Current Layout](#)
- [Enable Anti-Aliasing](#)
- [Configuration](#)

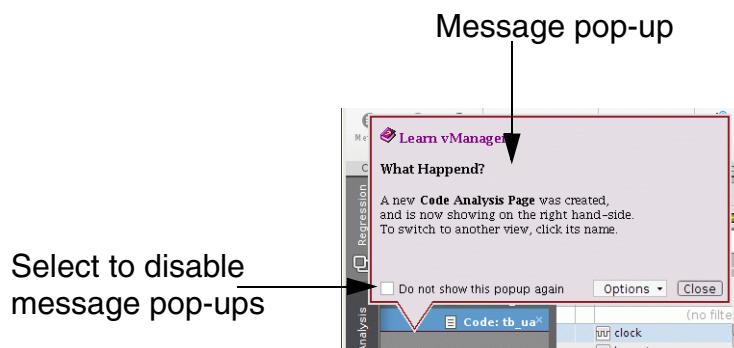
Toolbars

The *Toolbars* option of the *View* menu displays a list of toolbars that can be shown or removed from the current vManager window. The list of toolbars shown depends on the current activity center, page, and the sub-page. Using this option, you can enable or disable the showing of a specific toolbar. The selections and deselections you make are preserved for each page separately.

Organize Message Popups

In vManager, whenever you perform an action such as launching a new analysis page or selecting a specific view, a message pop-up window, as shown in [Figure 1-9](#) on page 19 is displayed. This pop-up describes what happened because of the action performed.

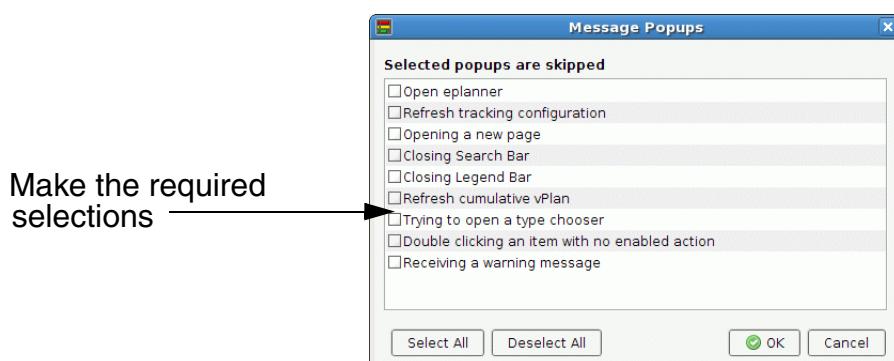
Figure 1-9 Pop-Up Window



You can disable such pop-ups by selecting the check box available on the message pop-up box.

The *Organize Message Popups* option of the *View* menu allows you to specify the scenarios for which message pop-ups must be displayed. [Figure 1-10](#) on page 20 shows the *Message Popups* dialog box that is displayed when you select the *Organize Message Popups* option from the *View* menu.

Figure 1-10 Message Pop-Ups



In the *Message Popups* dialog box, you can make the required selections or deselections. The message pop-up is disabled/skipped for the items you select in this dialog box.

Allow Showing All Message Popups

The *Allow Showing All Message Popups* option of the *View* menu enables showing all message pop-ups. It is the same as deselecting all of the items on the *Message Popups* dialog box.

Skip All Message Popups

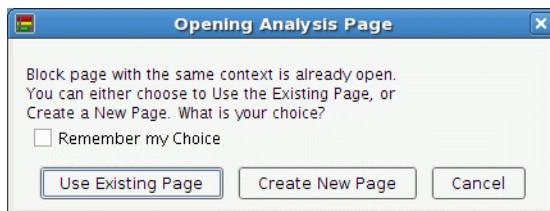
The *Skip All Message Popups* option of the *View* menu disables showing all message pop-ups. It is the same as selecting all of the items on the *Message Popups* dialog box.

Organize Dialog Choices

In the vManager GUI, when you perform certain actions, such as, opening an already-launched analysis page, a dialog box appears to indicate what happened.

Consider an example, where you already have an analysis page opened for an entity. Later, if you try to launch an analysis page again for the same entity, a dialog box, as shown in [Figure 1-11](#) on page 21 will be displayed.

Figure 1-11 Opening Analysis Page



This dialog box indicates that the page is already launched. It also prompts you to take an appropriate action, such as *Use Existing Page* or *Create New Page*.

This dialog box also enables remembering the choice you made in that scenario. If you select the check box, then the choice made on that dialog box is preserved for reuse in similar scenarios.

If you select the *Remember my Choice* check box and select *Use Existing Page*:

- A new page will not be launched and the already launched page will be used for further analysis, and
- Your action that is *Use Existing Page* will be preserved for reuse in similar scenarios.

Note: Next time, if you try to launch an already-launched page, you will not be prompted with this dialog box. The tool will automatically use the existing launched page because you selected the *Remember my Choice* check box. The choice made is reused if the same scenario is repeated, either in the same invocation of vManager, or in a different invocation.

The choices that you preserve in different dialog boxes can be selected or deselected using the *Organize Dialog Choices* option of the *View* menu.

When you select the *Organize Dialog Choices* option from the *View* menu, the *Dialog Choices* dialog box, as shown in [Figure 1-12](#) on page 22 is displayed.

Figure 1-12 Dialog Choices



The *Dialog Choices* dialog box displays a list of choices saved on different dialog boxes. This list keeps appending as you save more choices. The dialog boxes listed here are skipped, and the choice saved for the listed dialog boxes are used. You can remove an already saved choice by deselecting it from the list and clicking *OK*.

Allow Showing All Dialogs

The *Allow Showing All Dialogs* option removes all of the saved choices. It is the same as deselecting all of the choices on the *Dialog Choices* dialog box.

Auto Save Settings

By default, vManager saves the settings, such as the size of the vManager window, size of tables, that you set during the vManager session. These settings are automatically saved and reapplied to the next vManager invocation.

To disable this automatic saving of settings, deselect the *Auto Save Settings* check box in the *View* menu.

Save Settings

The *Save Settings* option allows you to preserve the settings, such as the size of the vManager window and size of tables, for reuse in the same as well as the next invocation of vManager.

Reset Settings

The *Reset Settings* option allows you to reset all of the settings and enable the default settings.

Note: This option does not enable the default settings immediately. Instead, the default settings take effect only on the next invocation of vManager.

This option hides the *Lookup* and *Settings* toolbars wherever applicable. For example, if the *Lookup* and *Settings* toolbars are shown in the *Analysis* center and you select the *Reset Settings* option, then next time when you invoke vManager, these toolbars will be hidden (not shown on the vManager screen). Later, when required, you can launch these toolbars from the *View* menu.

Restore Current Layout

While working in vManager, you might change the settings of view area, such as detach a pane, resize a pane, or hide a pane. The *Restore Current Layout* option helps you restore the original settings of the vManager view area.

For more details on detaching a pane, see [Detaching Panes](#) and for more details on hiding a pane, see [Hiding Panes](#).

Enable Anti-Aliasing

Anti-aliasing is a process where the edges of images and fonts have pixels added to blend the edge of the item into the background. The goal of anti-aliasing is to make the fonts and images appear smoother on a computer monitor.

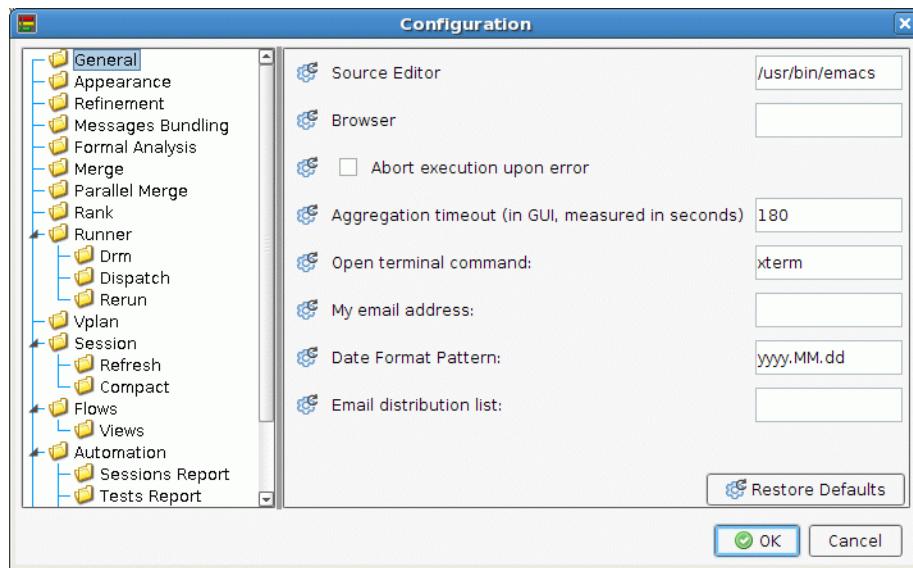
By default, this option is enabled. You can turn OFF anti-aliasing by clearing this check box.

Configuration

The *Configuration* option displays a dialog box using which you can configure options related to message logs, runner, vPlan, and sessions.

[Figure 1-13](#) on page 24 displays the *Configuration* dialog box.

Figure 1-13 Configuration



For more details, see [Configuring Configurable Items](#) on page 54.

1.2.1.3 Help

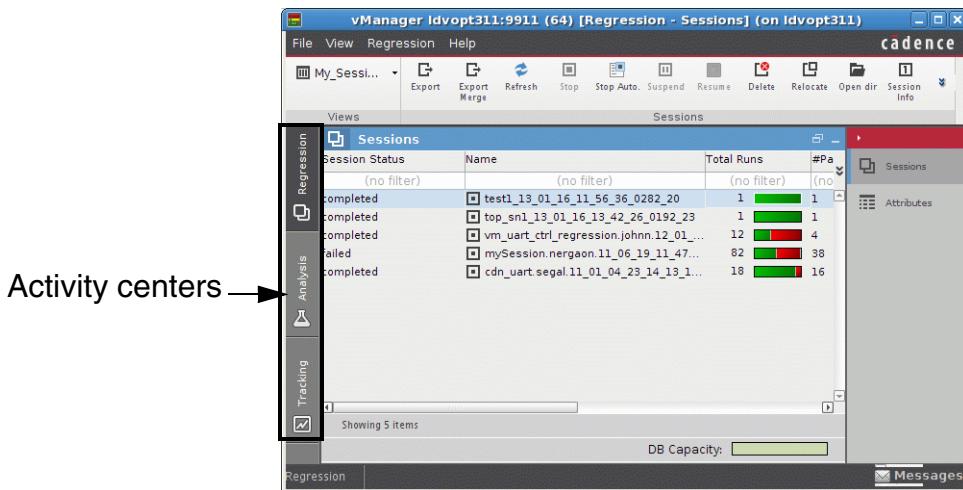
The *Help* menu provides you with access to the online help. You can choose:

- *User Guide*—To display the *vManager User Guide*
- *Quick Reference*—To display the *vManager Quick Reference Guide*
- *KPNS*—To display the *vManager Known Problems and Solutions* document
- *What's New*—To display the *vManager What's New* document
- *Cadence Help Library*—To launch *Cadence Help* with which you can view the currently installed documents
- *Cadence Online Support*—To open an Internet browser and launch the www.support.cadence.com page
- *About vManager*—To display the version and copyright information of the *vManager* GUI
- *Welcome*—To display the *Welcome* screen of the *vManager* GUI

1.2.2 Activity Centers

[Figure 1-14](#) on page 25 shows the activity centers available in vManager.

Figure 1-14 Activity Centers



vManager provides you with the following activity centers:

- Regression Center
 - Analysis Center
 - Tracking Center

1.2.2.1 Regression Center

The *Regression* center of the vManager allows you to load sessions and view the sessions details such as, the name of the session, the number of runs, number of passed runs, failed runs, and so on.

For more details, see [Loading and Viewing Sessions](#) on page 77.

1.2.2.2 Analysis Center

The *Analysis* center of the vManager allows you to do detailed analysis of runs, failures, metrics, and verification plans.

For more details, see the following chapters:

- [Analyzing Runs and Failures](#) on page 133
- [Analyzing Metrics](#) on page 217
- [Analyzing Verification Plans](#) on page 251

1.2.2.3 Tracking Center

The *Tracking* center allows you to track the project status. For more details, see [Project Tracking](#) on page 319.

1.2.3 Toolbar

The buttons and options available on the toolbar depends on the currently selected activity center, the active page, and selected sub-page.

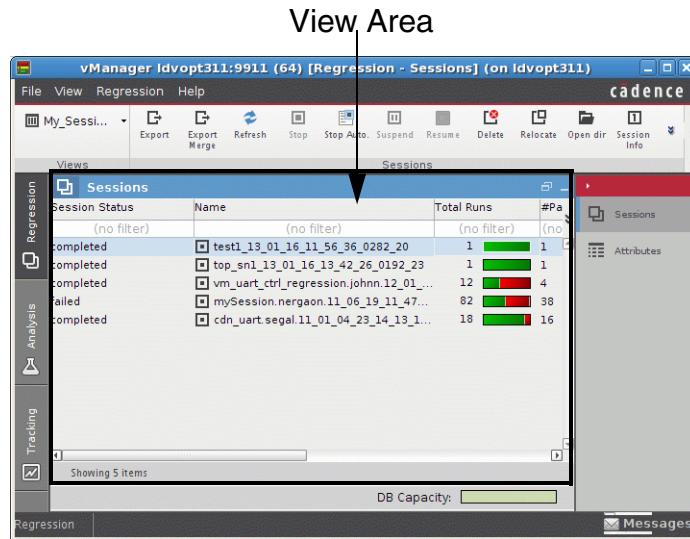
For details on toolbar options and buttons available in each activity center, see the respective chapters.

1.2.4 View Area

The view area of the vManager shows the various panes in which you can do detail analysis, navigate through various items, apply filters, organize items for analysis and so on. The panes shown in the view area depend on the currently selected activity center.

[Figure 1-15](#) on page 27 shows the view area of the *Regression* center.

Figure 1-15 View Area (Regression Center)

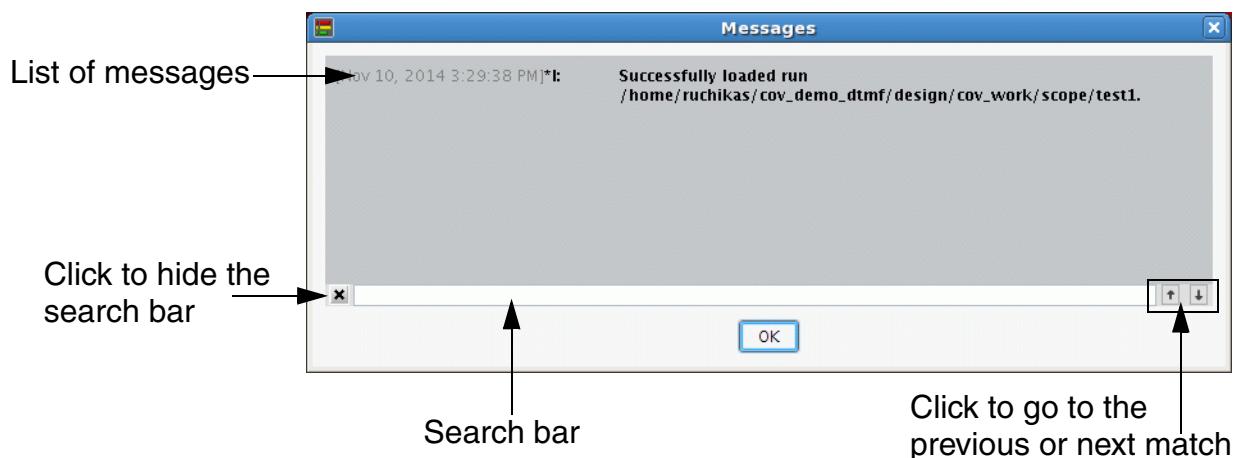


For more details on panes available in each activity center, see the respective chapters.

1.2.5 Messages Log

The vManager window displays *Messages* icon in the bottom left corner. When you click the *Messages* icon, the *Messages* dialog box is displayed, as shown in [Figure 1-16](#) on page 27.

Figure 1-16 Messages Log



The *Messages* dialog box shows the list of warnings or errors that occurred during the current vManager invocation.

In addition to viewing the message information, you can perform a search on the information displayed in the *Messages* dialog box.

To quickly search for a specific text, type the search text in the text box. As you type the text in the text box, the text gets highlighted in the messages area. You can use the *Up* and *Down* arrow keys to go to previous or next match. You can hide the search bar by clicking the *X* icon shown on the search bar. To reopen the search bar, press *Ctrl + F* keys.

1.3 Tasks Common to All Centers

This section covers the tasks that are common to all of the centers in vManager. This section covers the following topics:

- [Managing Table Columns](#) on page 28
- [Sorting Data in Tables](#) on page 31
- [Searching Data in Tables](#) on page 31
- [Filtering Table Data](#) on page 33
- [Detaching Panes](#) on page 39
- [Hiding Panes](#) on page 41
- [Defining and Organizing Views](#) on page 43
- [Exporting Table Data \(Sessions, Runs, Failures, Tests, vPlan, Metrics, Snapshots\) to a CSV File](#) on page 49
- [Generating Summary Report](#) on page 51
- [Configuring Configurable Items](#) on page 54
- [User-Defined Actions](#) on page 71

1.3.1 Managing Table Columns

In the vManager GUI, the data is shown in the form of tables. You can resize, reorder, remove columns, and sort data in different tables displayed in vManager GUI.

This section covers following topics:

- [Resizing Columns](#)
- [Reordering Columns](#)

- [Removing Columns](#)
- [Adding Columns](#)

Note: The maximum number of items that can be selected in a table or a tree is limited to 1000.

1.3.1.1 Resizing Columns

To resize a column, drag a separator in the header to a new location.

1.3.1.2 Reordering Columns

To reorder columns, drag the column header to its new location.

1.3.1.3 Removing Columns

To remove a column, right-click the column header and select *Remove Attribute*. When you select *Remove Attribute*, the selected column gets deleted from the table.

Note: You can also remove columns using the *Attribute Selector* dialog box. For more details, see [Adding Columns](#) on page 29.

1.3.1.4 Adding Columns

To add new columns:

1. Click the *Tools and Options* button and select *Select Attributes*.
2. The *Attribute Selector* dialog box opens, as shown in [Figure 1-17](#) on page 30. Select the attributes you want to display, and click *Add*.

Figure 1-17 Attribute Selector



In the *Attribute Selector* dialog box, you can add, remove, or define the placement of different attributes to be shown. The following buttons are available on this dialog box to perform these actions:

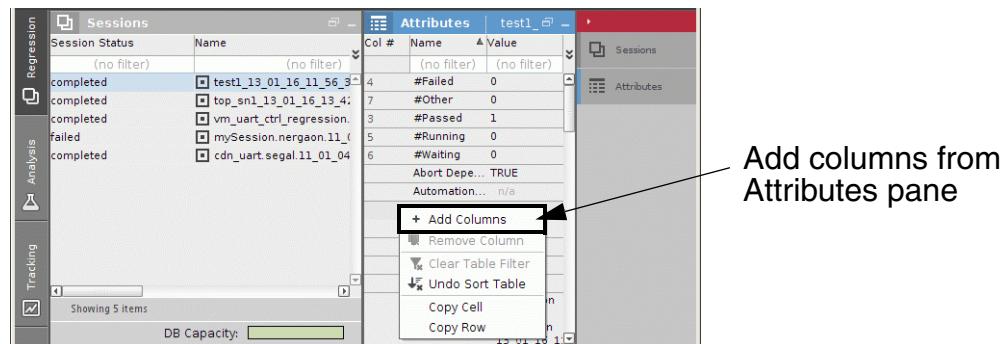
- Add—To add the selected attribute
- Top—To move the selected attribute to the top of the list
- Up—To move the selected attribute one step up in the list
- Down—To move the selected attribute one step down in the list
- Bottom—To move the selected attribute to the bottom of the list
- Remove—To remove the selected attribute
- Remove All—To remove all attributes present in the *Selected Attributes* list box.

After you make required selections, the selected attributes are displayed in the table.

Note: Alternatively, you can open the *Attribute Selector* by right-clicking the column header and selecting *Select Attributes*.

Note: You can also add columns from the *Attributes* tab page that appears on the right pane. To add columns from the *Attributes* tab page, right-click on a attribute and select *Add Columns*, as shown in [Figure 1-18](#) on page 31.

Figure 1-18 Adding Attribute



1.3.2 Sorting Data in Tables

To sort data in tables based on a column's values, click its column header. Click again to reverse the order.

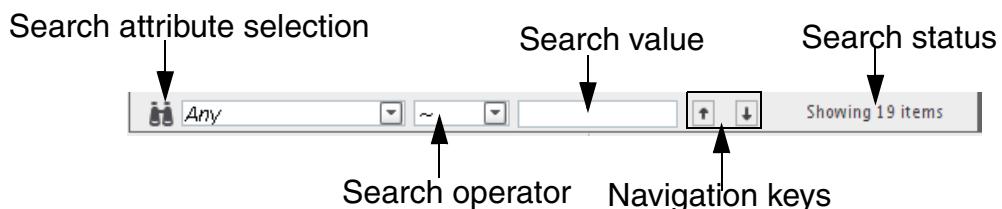
Note: You can unsort table data by selecting *Unsort Data* from the Tools & Options drop-down.

1.3.3 Searching Data in Tables

[Figure 1-19](#) on page 31 displays the search bar shown at the bottom of the Verification Hierarchy pane of the *Metrics* page. For more details on Verification Hierarchy pane, see [Analyzing Metrics](#) on page 217.

The search bar helps you to quickly navigate through the items that meet the specified search criteria.

Figure 1-19 Verification Hierarchy Pane—Search Bar



To navigate through the items based on a specified search criteria:

1. Select the attribute on which the search must be performed by clicking the search attribute selection drop-down. For example, to quickly navigate through the design

hierarchy for instances where overall local grade is <50, select *Overall Local Grade* from the search attribute selection drop-down.

Figure 1-20 on page 32 shows the search bar with *Overall Average Grade* selected.

Figure 1-20 Verification Hierarchy Pane—Search Bar (Attribute Selected)



2. For example, to set the criteria as Overall Local Grade <50, select < from the search operator drop-down.

Figure 1-21 on page 32 shows the search bar with search operator specified as <=.

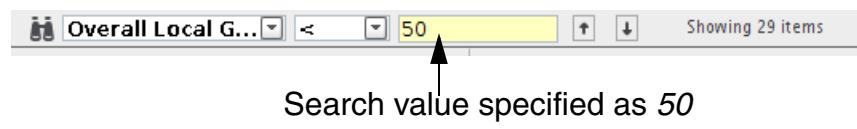
Figure 1-21 Verification Hierarchy Pane—Search Bar (Operator Selected)



3. Specify the search value and press Enter. For example, to set the criteria as Overall Local Grade <50, specify the value as 50 in the text box and press Enter.

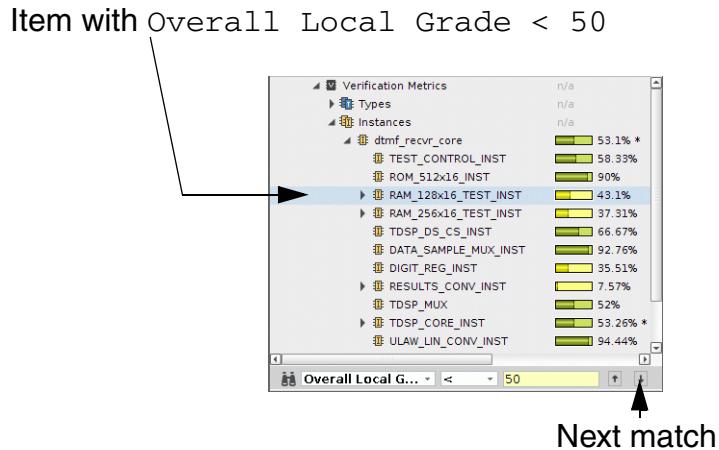
Figure 1-22 on page 32 shows the search bar with search value specified as 50.

Figure 1-22 Verification Hierarchy Pane—Search Bar (Value Specified)



After specifying the search value, when you press the `Enter` key, the item that meets the specified search criteria gets selected in the verification hierarchy, as shown in Figure 1-23 on page 33.

Figure 1-23 Verification Hierarchy Pane



4. To navigate directly to the next instance that meets the search criteria, click the *Next Match* arrow key.

Similarly, you can quickly navigate to other items, and also specify a different search criteria, as required.

1.3.4 Filtering Table Data

You can filter the data in tables to display only the required items. The row below the column header is the Quick Filter Bar, which allows you to add the filtering criteria, as shown in [Figure 1-24 on page 33](#).

Figure 1-24 Filtering Data

Index	Name	Status
(filter)	(no filter)	(no filter)
1	/cdn_uart_apbe_tests/data_poll	passed
2	/cdn_uart_apbe_tests/data_poll	passed
3	/cdn_uart_apbe_tests/data_poll	passed
4	/cdn_uart_apbe_tests/data_poll	passed
5	/cdn_uart_apbe_tests/data_poll	passed

This section covers the following topics:

- [Filtering Textual Values](#)
- [Filtering Numerical Values](#)
- [Filtering Enumeration Values](#)
- [Filtering Date and Time Values](#)

- Advanced Filtering

1.3.4.1 Filtering Textual Values

To filter textual values, specify the search string as any of the following:

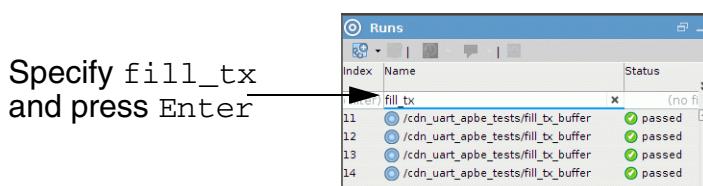
- `~ value` to filter table data to show data that match the specified `value`. For example, `~trans` in `Name` column will show data where `trans` appears in the `Name`.
- `!~ value` to filter table data to show data that do not match the specified `value`. For example, `!~trans` in `Name` column will show data where `trans` does not appear in the `Name`.

Note: The default operator is `~`.

- `one_of value1,value2,...,valuen` to filter table data to show data that matches the comma separated values specified along with the `one_of` operator. For example, `one_of regs,dgb` in `Name` column will show data where `Name` is either `regs` or `dgb`.

For example, to filter the list of runs to display runs that include `fill_tx` in the run name, specify `fill_tx` or `~fill_tx` in the filter text box, as shown in [Figure 1-25](#) on page 34, and press `Enter`.

Figure 1-25 Filter Textual Values



Notice that the list now displays only the runs that include `fill_tx` in the run name.

Note: When filtering string attributes, you can also use the `$ME()` filter, which filters the column by matching values against the user name.

1.3.4.2 Filtering Numerical Values

To filter numerical values, specify the search string as any of the following:

- `== value` to filter values that are equal to the specified `value`
- `!= value` to filter values that are not equal to the specified `value`
- `> value` to filter values that are greater than the specified `value`

- `>= value` to filter values that are greater than or equal to the specified value
- `< value` to filter values that are less than the specified value
- `<= value` to filter values that are less than or equal to the specified value

Note: The default operator is `==`. In addition, the above mentioned search strings apply to date as well as enumeration values also.

For example, to filter runs what include `fill_tx` in the run name and the duration is more than 20, specify `>20` in the filter text box, as shown in [Figure 1-26](#) on page 35, and press Enter.

Figure 1-26 Filter Numerical Values

Specify `>20` and press Enter

The screenshot shows a table titled "Runs" with columns: Index, Name, Status, and Duration (sec.). A filter is applied to the Duration column with the value "`>20`". The table contains two rows: index 11 with duration 22 and index 13 with duration 25. Both rows have a green checkmark in the Status column, indicating they passed the filter.

Index	Name	Status	Duration (sec.)
filter	fill_tx	(no filter)	>20
11	/cdn_uart_apbe_tests/fill_tx_b...	passed	22
13	/cdn_uart_apbe_tests/fill_tx_b...	passed	25

The only runs listed now are the ones with a duration greater than 20 seconds. In addition, notice that if multiple filters are applied, then the result is the intersection of matches. In this case, only runs that include `fill_tx` in the run name with a duration of more than 20 seconds are listed.

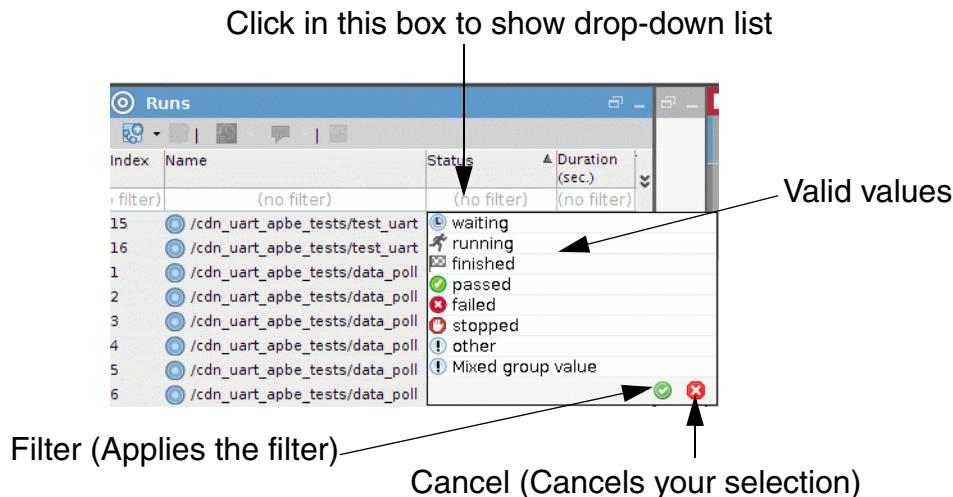
1.3.4.3 Filtering Enumeration Values

To apply filters to values in enumeration columns, perform the following steps:

1. Click in the Filter text box of an enumeration field. For example, click in the Filter text box of the *Status* column.

A drop-down is displayed with valid filter values, as shown in [Figure 1-27](#) on page 36.

Figure 1-27 Filtering Enumeration Values



Note: The status value can be Mixed group value, when you group by other attribute.

2. Select an appropriate value from the list. You can also select multiple values by holding the **Ctrl** key. For example, to show only the failed runs, select *failed* from the list.
3. Click the *Filter* icon to confirm the filtering action.

After applying the filter, only the failed runs are shown in the table, as shown in [Figure 1-28](#) on page 36.

Figure 1-28 Filtered Values in Status Column

Filtered values

Index	Name	Status	Duration (sec.)
(no filter)	(no filter)	(no filter)	(no filter)
15	/cdn_uart_apbe_tests/test_uart	failed	19
16	/cdn_uart_apbe_tests/test_uart	failed	17

Similarly, you can apply more filters based on your requirements.

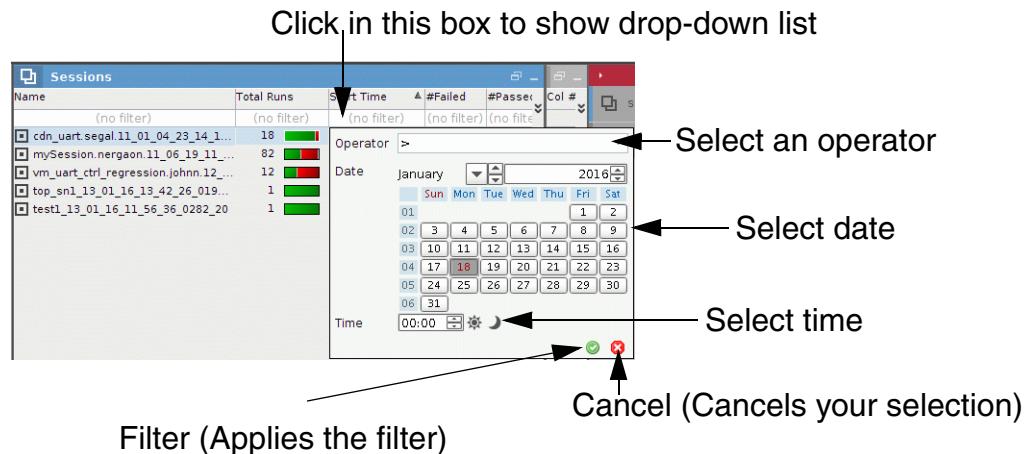
1.3.4.4 Filtering Date and Time Values

To apply filters to values in date and time columns, perform the following steps:

1. Click in the Filter text box of the date and time field. For example, click in the Filter text box of the *Start Time* column.

A drop-down is displayed with valid filter values, as shown in [Figure 1-29](#) on page 37.

Figure 1-29 Filtering Date and Time Values



2. Select an appropriate operator from the *Operator* drop-down list.
3. Select the *Date*.
4. Specify *Time*.
5. Click the *Filter* icon to confirm the filtering action.

Similarly, you can apply more filters based on your requirements.

1.3.4.5 Advanced Filtering

The advanced filtering feature enables you to add a complex filter to a table or to tree table.

It is useful when you want to apply more than one filter on the same attribute using AND or OR relationship and also useful in situations where you want to apply filters on attributes that are not displayed in the table at that time.

Note: Only basic filters work in batch mode. Advanced filters with AND or OR operations do not work in batch mode.

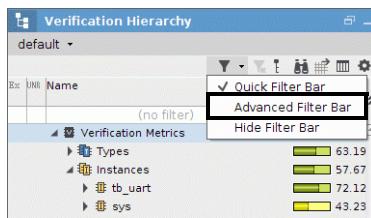
Displaying the Advanced Filter Bar

By default, the Quick filter bar is displayed in the different panes of vManager.

To apply advanced filters, you need to first open the Advanced filter bar. For this, perform the following steps:

1. Click the Tools & Options double arrow button. This will show additional buttons in the pane.
2. Now click the *Filter table* drop-down and select *Advanced Filter Bar* option, as shown in [Figure 1-30](#) on page 38.

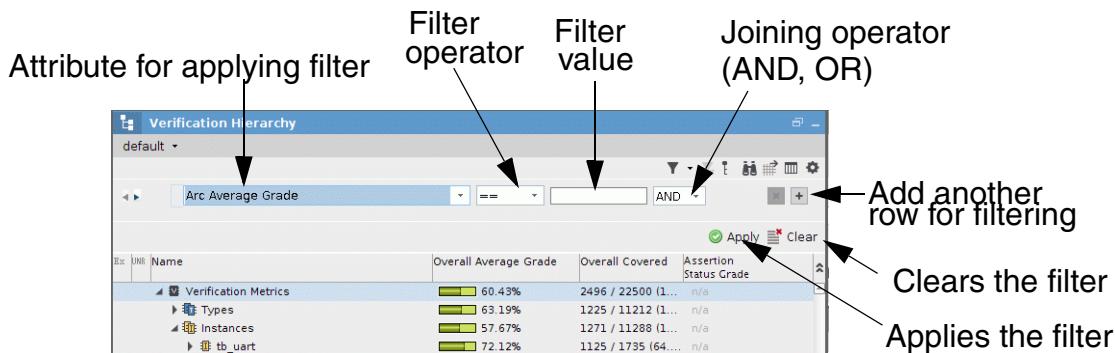
Figure 1-30 Advanced Filter



This will hide the Quick filter bar and show the Advanced filter bar.

[Figure 1-31](#) on page 38 shows the Advanced filter bar.

Figure 1-31 Advanced Filter Bar



Applying Advanced Filters

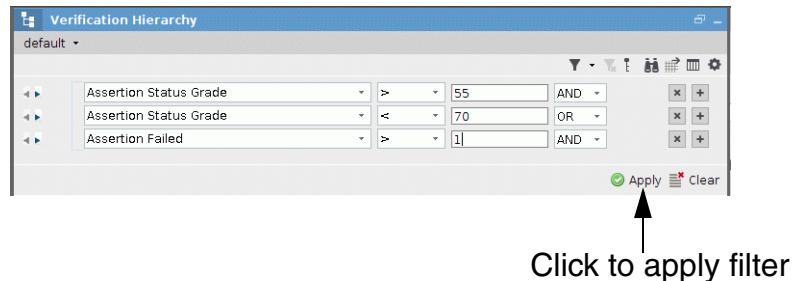
After displaying the Advanced filter bar, you can add complex filters, as required.

For example, you want to apply filter such that:

- Assertion Status Grade is >55 and <70 , or
- Assertion Failed >1 .

[Figure 1-32 on page 39](#) shows the Advanced filter bar with required values.

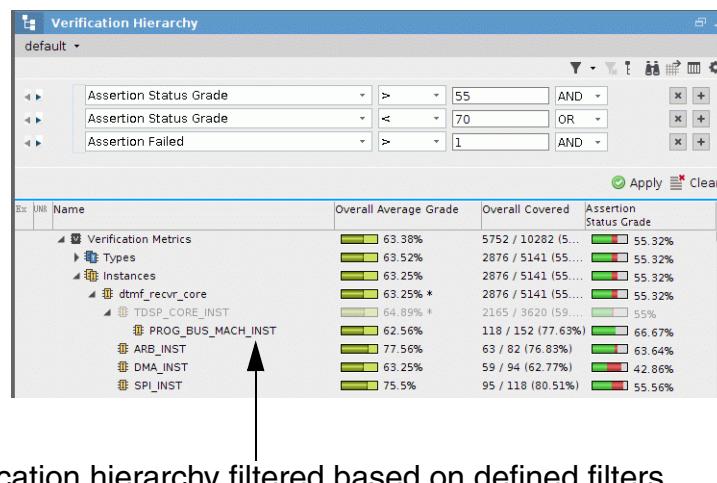
Figure 1-32 Advanced Filter Example



Click the *Filter* button to apply the filter.

[Figure 1-33 on page 39](#) shows the Verification hierarchy filtered based on the defined filter.

Figure 1-33 Advanced Filter Example



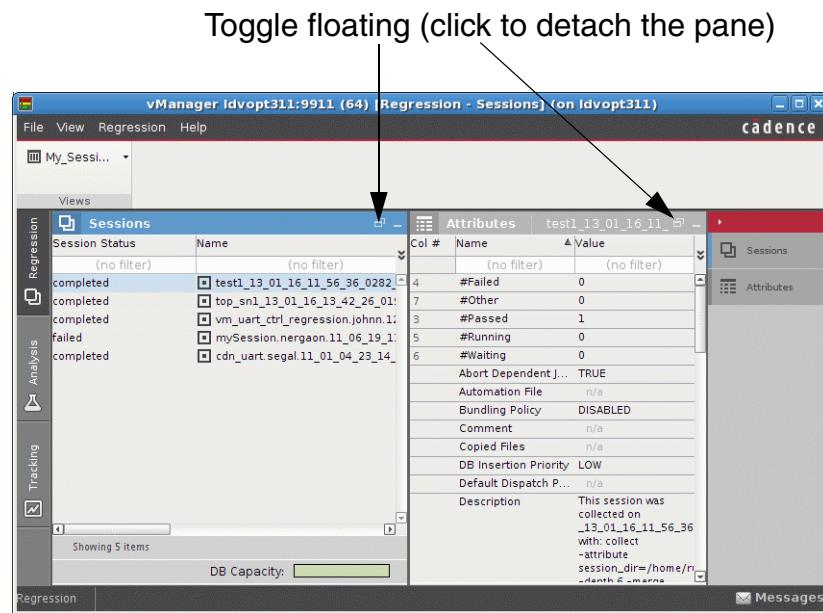
Similarly, you can apply more filters based on your requirements.

1.3.5 Detaching Panes

In vManager, you can detach panes from the view area and move it, as required.

[Figure 1-34 on page 40](#) displays the *Toggle floating* icon (for detaching a pane) that appears on the title bar of different panes in vManager.

Figure 1-34 Toggle Floating

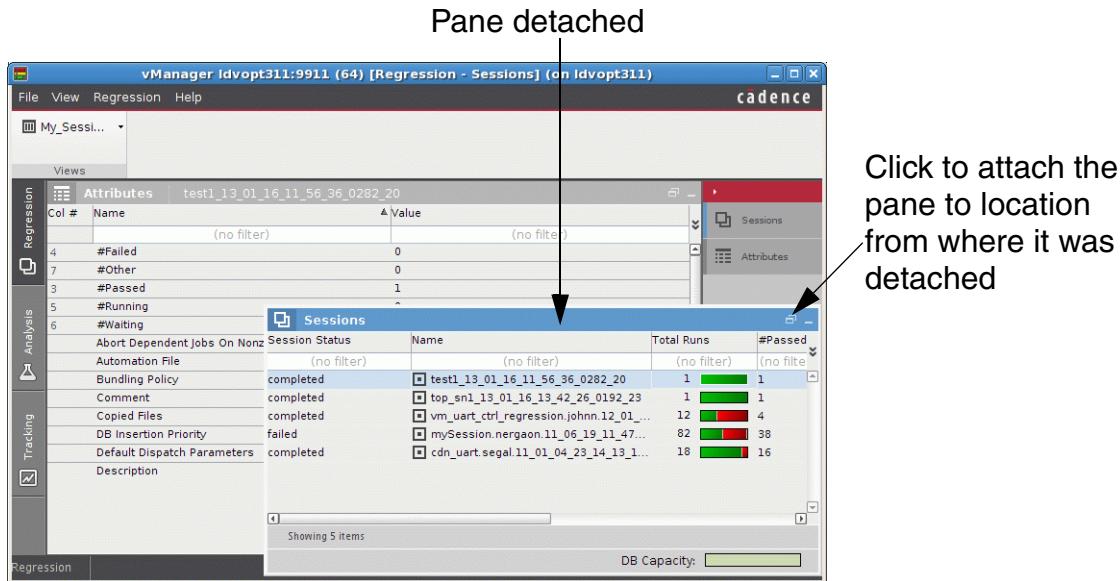


The *Toggle floating* icon is used to detach the pane from the view area and move it to a new location.

Note: You can also detach a pane by clicking on the title bar of the pane and then with the left-mouse button pressed, moving it to a new location.

For example, to detach the *Sessions* pane, click the *Toggle floating* icon in the title bar of the *Sessions* pane. After you click *Toggle floating* icon, the pane is detached, as shown in [Figure 1-35](#) on page 41.

Figure 1-35 Toggle Floating (Detached Pane)



The above figure shows the *Sessions* pane detached. You can attach it back to the location from where it was detached by clicking the *Toggle floating* icon in the title bar of the pane.

Note: You can also restore the detached pane back to its original location by selecting *Restore Current Layout* option in the *View* menu.

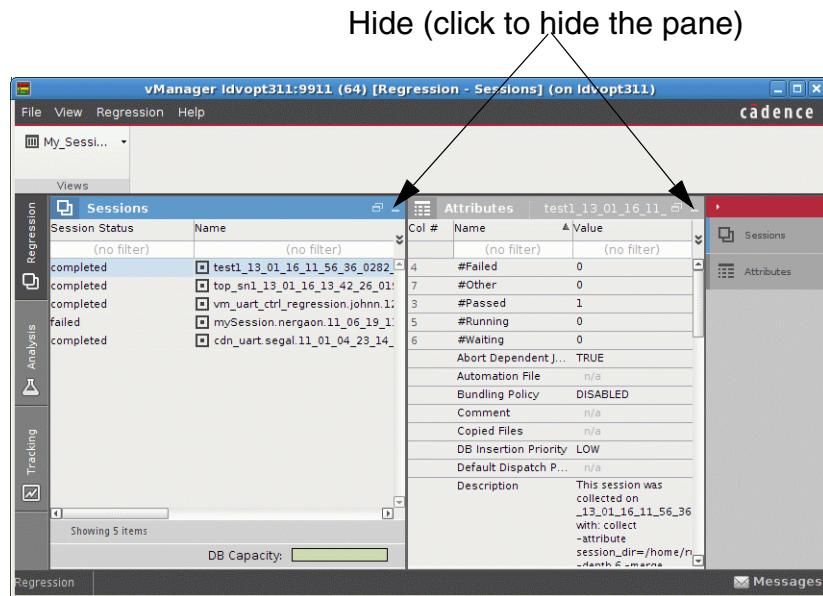
Note: The *Restore Current Layout* option sets the default layout for all the panes in all the opened views.

1.3.6 Hiding Panes

In vManager, you can hide panes, as required.

[Figure 1-36](#) on page 42 displays the *Hide* icon that appears on the title bar of different panes in vManager.

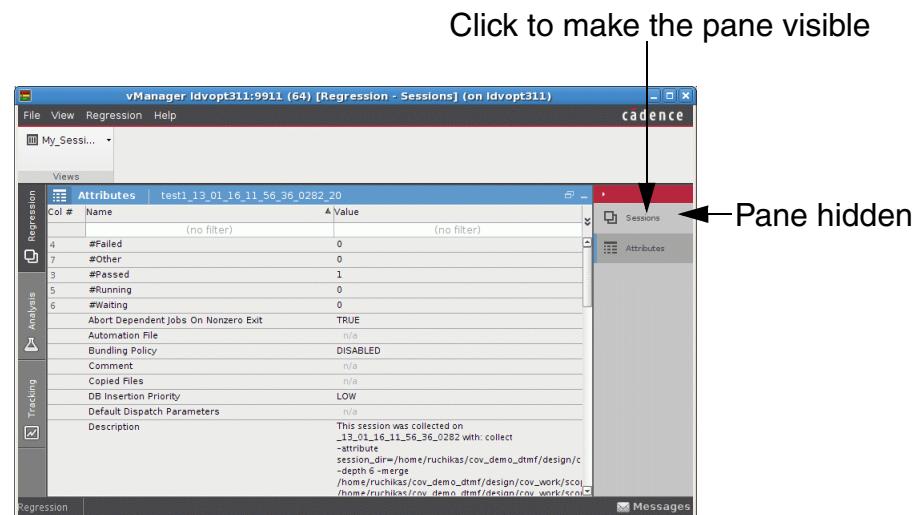
Figure 1-36 Hide



The *Hide* icon is used to hide the pane. For example, to hide the *Sessions* pane, click the *Hide* icon in the title bar of the *Sessions* pane.

[Figure 1-37](#) on page 42 shows the vManager screen with the hidden pane.

Figure 1-37 Hide (Hidden Pane)



The above figure shows the *Sessions* pane hidden. You can click the tab to restore the hidden pane to its original location.

Note: You can also restore the hidden pane to its original location by selecting *Restore Current Layout* option in the *View* menu.

Note: The *Restore Current Layout* option sets the default layout for all the panes in all the opened views.



You can double-click on the title bar of a pane to maximize the pane.

1.3.7 Defining and Organizing Views

vManager allows you to define views such that you include only the attributes of your interest. You can also apply sorting, filtering, and grouping as per your requirement and save it as a view.

Note: After you save a view, it is available (listed in the *Views* drop-down) for all the analysis pages of same type. However, it is not applied automatically when you launch a new analysis page until you set the view as the default view. For example, you remove attributes, or add attributes on a block analysis page of an instance and save it as a view. When you open the block analysis page for another instance or type, then the saved view will be listed in the available views but it will not be set automatically. In case, you want a particular view to be used as the default view when you open a new analysis page, set that view as the default view.

This section covers following topics:

- [Defining Views](#)
- [Saving Views](#)
- [Deleting Views](#)
- [Renaming Views](#)
- [Setting a Default View](#)
- [Organizing Views Under Different Folders](#)

1.3.7.1 Defining Views

For example, you can define a view named `my_failures` such that:

- It includes sessions associated with your name.

For this, you can filter the table data to show only the sessions where the *Owner* for instance is <your-name>. For more details on filtering, see [Filtering Table Data](#) on page 33.

- It does not include attributes, such as #Waiting, #Running, and #Other.

For this, you can remove the specified attributes by right-clicking the column header and selecting *Remove Attribute*.

After defining a view you must save it so that is available to you in later vManager invocations.

1.3.7.2 Saving Views

To save the view:

1. Click the drop-down in the *Views* toolbar.
2. Select *Save As View* from the drop-down menu.

The *Save As View* dialog box is displayed, as shown in [Figure 1-38](#) on page 44.

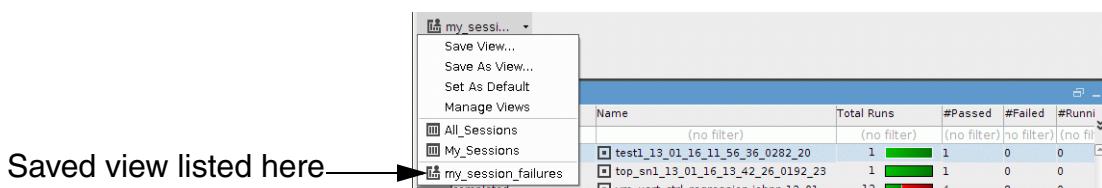
Figure 1-38 Save View



3. Specify the name of the view and click *OK*. For example, to save the view as *my_session_failures*, specify *my_session_failures* in the text box and click *OK*.

The view is created and listed, as shown in [Figure 1-39](#) on page 44.

Figure 1-39 Saved View



The views are stored in the database and are available in the next vManager invocation. It saves your time as you do not need to create views in each vManager invocation.

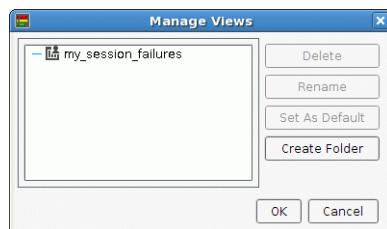
1.3.7.3 Deleting Views

To delete a view:

1. Click the drop-down in the *Views* toolbar.
2. Select *Manage Views* from the drop-down menu.

The *Manage Views* dialog box is displayed, as shown in [Figure 1-40](#) on page 45.

Figure 1-40 Manage Views



3. Select the view you want to delete and click *Delete*.
4. Click *OK*.

The selected view is then deleted from the list of views.

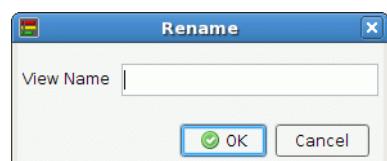
1.3.7.4 Renaming Views

To rename a view:

1. Click the drop-down in the *Views* toolbar.
2. Select *Manage Views* from the drop-down menu.
3. Select the view you want to rename and click *Rename*. For example, select *my_session_failures* and click *Rename*.

The *Rename* dialog box is displayed, as shown in [Figure 1-41](#) on page 45.

Figure 1-41 Renaming Views



4. Specify the new name for the view and click *OK*. For example, specify *my_failed_sessions* and click *OK*.

The new name shows in the *Manage Views* dialog box.

5. Click *OK*.

1.3.7.5 Setting a Default View

Using vManager you can set a particular view as the default view. The view that is set as the default view in one of the vManager sessions is automatically set in subsequent vManager sessions.

For example, you set *my_failed_sessions* as the default view and exit out of vManager. Later, when you again launch vManager, by default, *my_failed_sessions* will already be set.

You can set a view as the default view using any of the following methods:

- [From the All Sessions drop-down in the toolbar](#)
- [From the Manage Views Dialog Box](#)

From the All Sessions drop-down in the toolbar

To set a view as the default view:

1. Launch the view that you want to set as the default view. For this, select that view from the drop-down in the *Views* toolbar.
2. After the view is launched, select *Set as Default* from the *All Sessions* drop-down in the toolbar.

This will set the selected view as the default view.

From the Manage Views Dialog Box

To set a view as the default view:

1. Click the drop-down in the *Views* toolbar.
2. Select *Manage Views* from the drop-down menu.
The *Manage Views* dialog box is displayed.
3. Select the view that you want to set as the default view.

4. Click the *Set As Default* button.
5. Click *OK*.

This will set the selected view as the default view.

1.3.7.6 Organizing Views Under Different Folders

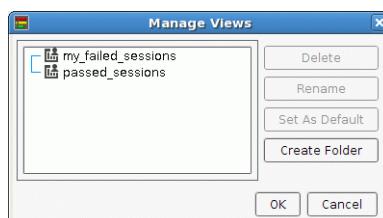
Using vManager, you can organize views under different folders. By default, all the views are listed at one level. In the case of multiple views, it is difficult to find the required view. You can create separate folders and organize your views in different folders.

To organize views in different folders:

1. Click the drop-down in the *Views* toolbar.
2. Select *Manage Views* from the drop-down menu.

The *Manage Views* dialog box is displayed, as shown in [Figure 1-42](#) on page 47.

Figure 1-42 Manage Views



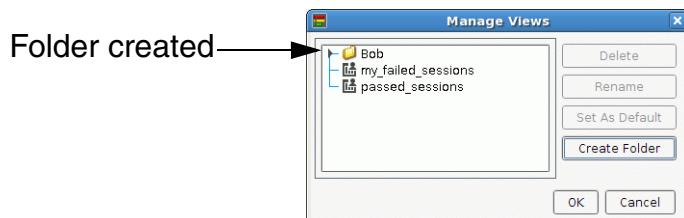
3. Click *Create Folder*.

The *Create Folder* dialog box is displayed.

4. Specify the name of the folder in the text box and click *OK*. For example, specify *Bob* in the text box and click *OK*.

The folder is created and shown in the *Manage Views* dialog box, as shown in [Figure 1-43](#) on page 48.

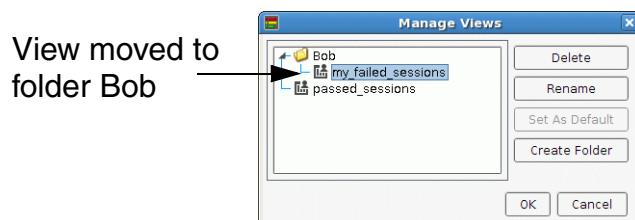
Figure 1-43 Create Folder



5. You can now move views to the folder. Select the view you want to move. Drag the view and drop it in the required folder. For example, select *my_failed_sessions*, drag it and drop it under the folder *Bob*.

The folder is created and shown in the *Manage Views* dialog box, as shown in [Figure 1-44](#) on page 48.

Figure 1-44 Move View

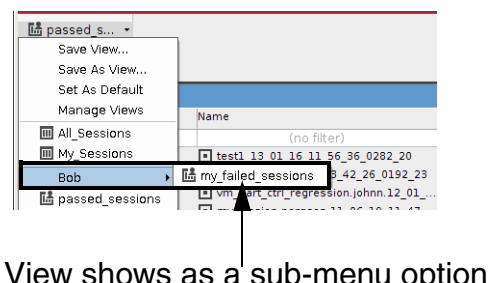


The view *my_failed_sessions* is now moved to the folder *Bob*.

6. Click *OK* to close the *Manage Views* dialog box.

[Figure 1-45](#) on page 48 shows how the view is organized.

Figure 1-45 Organized View



The view *my_failed_sessions* is now shown as a sub-menu of option *Bob*.

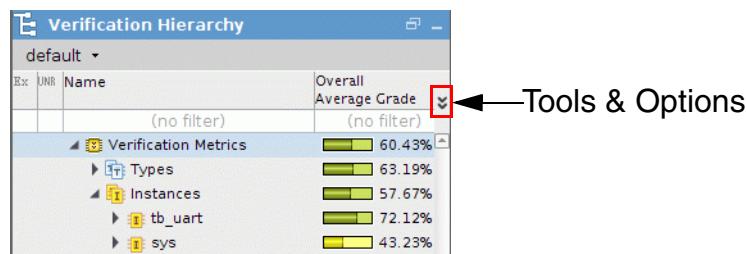
1.3.8 Exporting Table Data (Sessions, Runs, Failures, Tests, vPlan, Metrics, Snapshots) to a CSV File

You can export table data shown in various tables to a CSV file.

To export sessions, runs, tests, snapshots, vplan, and metrics data to a CSV file, perform the following steps:

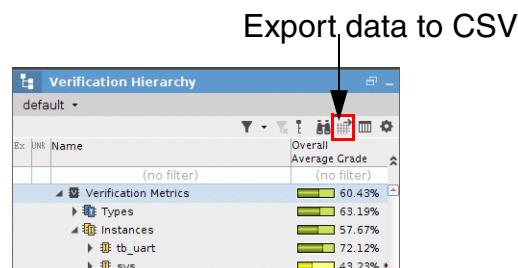
1. Click the *Tools and Options* double arrow, as shown in [Figure 1-46](#) on page 49.

Figure 1-46 Export to CSV File



2. Select *Export data to csv file* from list of advanced buttons, as shown in [Figure 1-47](#) on page 49.

Figure 1-47 Export to CSV File



3. The *Export to csv file* dialog box appears on the basis of table data.

- If the table data is flat structure (data as individual rows), then the *Export to csv file* dialog box appears as shown in [Figure 1-48](#) on page 50.

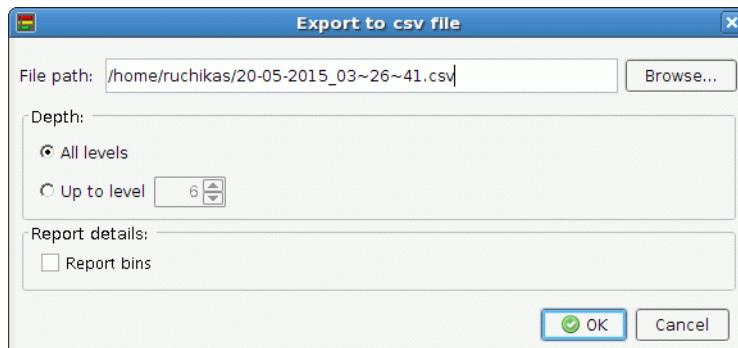
Figure 1-48 Export to CSV File



In the *File path* field, you can specify the location and name of the file where the exported file must be saved. By default, it shows the location from which vManager was invoked and the file name is shown as <timestamp>.csv. You can change it, as required and click *OK*.

- ❑ If the table data is as a hierarchy tree, then the *Export to csv file* dialog box has additional options, as shown in [Figure 1-49](#) on page 50.

Figure 1-49 Export to CSV File



You can specify the levels of hierarchy that must be exported to CSV file. You can select any of the following:

- All levels — To export all the levels of hierarchy. By default, this option is selected.
- Up to level — To specify the number of levels to be exported. By default, 6 is specified in this field. You can increase or decrease the levels, as required.

To include cover bins in the CSV file, select the *Report bins* check box.

After you click *OK*, the metrics tree is exported to the CSV file at the specified location.

Note: The data included in the exported CSV file is based on the view selected, filters, grouping, and sorting applied at the time of exporting to CSV file.

Note: If an attribute's value is n/a in GUI, then it is taken as an empty string in the exported CSV file.

You can also export the data from the command-line interface of vManager. For more details, see [csv_export](#).

1.3.9 Generating Summary Report

A summary report is a one-page report which includes different sections to list the details of sessions, tests, metrics, and vPlan (as specified in the *Summary Report* dialog box).

The summary report is generated as a single HTML file and if required, can be embedded in the email body.

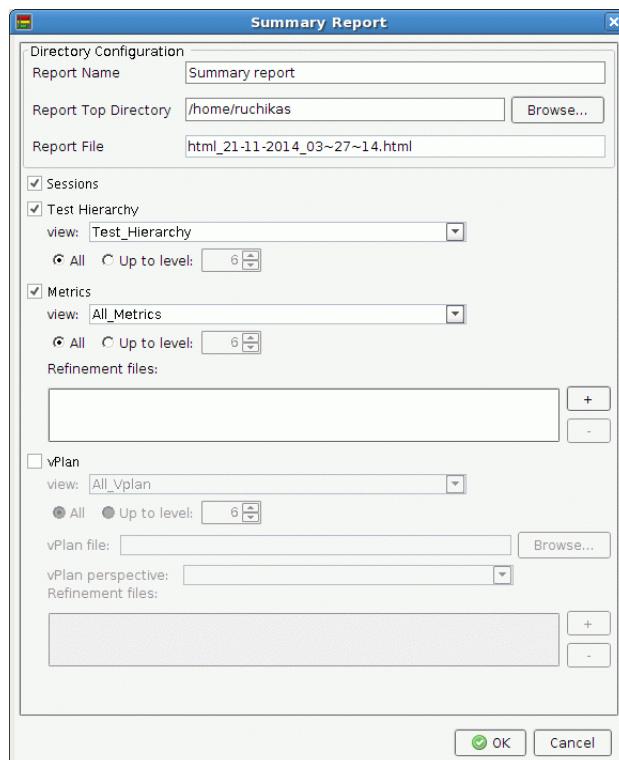
A summary report can be generated from any of the Activity Centers.

To generate a summary report:

1. Click the down-arrow next to the *Report* button in the *Report* toolbar.
2. Select *Summary Report*.

The *Summary Report* dialog box is displayed, as shown in [Figure 1-50](#) on page 51.

Figure 1-50 Summary Report



3. Specify the name of the report in the *Report Name* text box. By default, the name is specified as *Summary report*. This name appears on the HTML report.
4. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
5. Specify the name of the report file in the *Report File* text box. By default, it shows the name as `html_<timestamp>.html`. For example, change the name as `summary_rep.html`.
6. By default, *Sessions* check box is selected. This indicates that the Sessions information will be included in the summary report. Clear the check box, if you do not want the *Sessions* section to be included in the report.
7. By default, *Test Hierarchy* check box is selected. This indicates that the Tests Hierarchy information will be included in the summary report. Clear the check box, if you do not want the *Tests Hierarchy* section to be included in the report.
8. If required, you can select a Tests hierarchy view to be used from the *View* drop-down.
9. Next, specify the levels of hierarchy of the Tests hierarchy that must show in the report. You can select any of the following:
 - All— To show all the levels of hierarchy. By default, this option is selected.
 - Up to level — To specify the number of levels to be shown. By default, 6 is specified in this field. You can increase or decrease the levels, as required.
10. By default, *Metrics* check box is selected. This indicates that the Metrics information will be included in the summary report. Clear the check box, if you do not want the *Metrics* section to be included in the report.
11. If required, you can select a metrics hierarchy view to be used from the *View* drop-down.
12. Next, specify the levels of hierarchy for the Metrics hierarchy that must show in the report. You can select any of the following:
 - All— To show all the levels of hierarchy. By default, this option is selected.
 - Up to level — To specify the number of levels to be shown. By default, 6 is specified in this field. You can increase or decrease the levels, as required.
13. If the refinement files are loaded, then the files will show in the *Refinement files* field. In case you do not want to apply refinements to the metrics data while generating summary report, select the file and remove it. If no refinement files are already loaded and you want to apply refinements while generating report, click the + sign next to the *Refinement files* field and specify the refinement files to be specified.

14. If required, you can include the vPlan Hierarchy in the report. For this, select the *vPlan* check box. This will enable other options related to vPlan.
15. Select a vPlan hierarchy view to be used from the *View* drop-down.
16. Next, specify the levels of hierarchy for the vPlan hierarchy that must show in the report. You can select any of the following:
 - All— To show all the levels of hierarchy. By default, this option is selected.
 - Up to level — To specify the number of levels to be shown. By default, 6 is specified in this field. You can increase or decrease the levels, as required.
17. Specify the vPlan to be included in the *vPlan File* field. You can click the *Browse* button to navigate and select the vPlan to be included.
18. Select the perspective to be used from the vPlan perspective drop-down list.
19. In case you want to apply refinements to the vPlan data, you can specify the refinement files in the *Refinement files* field.
20. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 1-51](#) on page 53.

Figure 1-51 Report Done



This dialog box shows the location where the report is generated.

21. Click *OK*.

You can navigate through the location shown above, and then open the HTML file in a Web browser.

[Figure 1-52](#) on page 54 shows the generated HTML report.

Figure 1-52 Summary Report

The screenshot shows a Mozilla Firefox browser window with the title "Summary report - Mozilla Firefox (on Idvopt311)". The address bar displays the URL "file:///home/ruchikas/html_21-11-2014_03~27~14.html". The main content area is titled "Cadence Summary Report" and includes the following sections:

- Context Info:**
 - loaded vplan: not loaded
 - loaded refinements: not loaded
 - loaded vplan refinements: not loaded
- Sessions**

Pre Filter: (Owner MATCHES segal) Sorting: [Start Time (DESCENDING)]

No.	Session Status	Name	Total Runs	#Passed	#Failed	Start Time	Owner
1	completed	cdn_uart.segal.11_01_04_23_14_13_1797	18	16	2	Wed Jan 05 02:44:13 IST 2011	segal
- Tests Hierarchy**

View Name: Test_Hierarchy

Expand all Collapse all

Name	Overall Average Grade	Overall Covered	Test Status
Test-Case Model	75%	16 / 18 (88.89%)	88.89%
- Metrics**

View Name: All_Metrics

Verification Scope: default

Expand all Collapse all

Exclusion Rule Type	UNR	Name	Overall Average Grade	Overall Covered	Assertion Status Grade
None	none	Verification Metrics	60.43%	2496 / 22500 (11.09%)	n/a
- Coverage Color Legend**

0 | <25 | <50 | <75 | <100 | 100 | n/a | Not Scored

The summary report shows information such as, who generated the report and when the report was generated. The report includes sections, such as *Sessions*, *Tests Hierarchy*, *Metrics*, and *vPlan* (based on options specified at the time of generating the report). You can expand or collapse the hierarchy tree, as required.

Note: The report action considers the configuration option *Show extend metrics tree* while generating the report. If the *Show extend metrics tree* option is enabled, then the covergroups, cover items, FSMs, and assertions are also listed in the metrics hierarchy tree in the report. For more details on this option, see [Configuring Appearance Options](#) on page 56.

You can also generate this report from the CLI mode using the `report_summary` command.

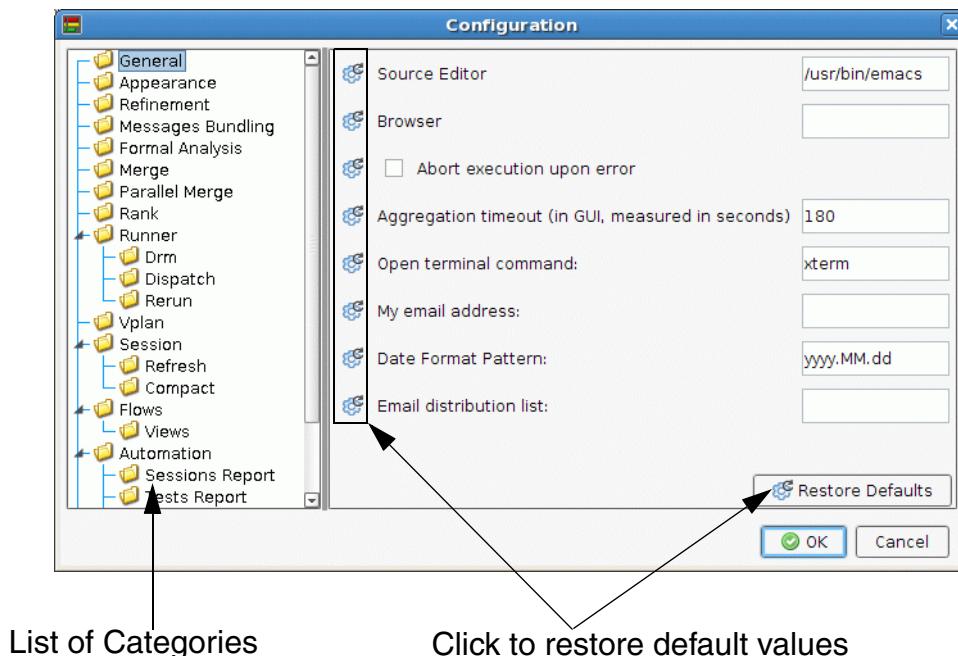
1.3.10 Configuring Configurable Items

The *Configuration* dialog box allows you to configure general options and options related to appearance, refinement, message logs, runner, vPlan, and sessions.

You can invoke the *Configuration* dialog box by selecting *Configuration* from the *View* menu.

[Figure 1-53 on page 55](#) displays the *Configuration* dialog box.

Figure 1-53 Configuration



The *Configuration* dialog box has different categories whose options can be configured.

You can click any of the categories in the left pane and its related options are shown in the right pane of the *Configuration* dialog box.

Note: For the numeric fields in the *Configuration* dialog box, there is a valid range of values that can be specified. In case you specify a value outside the range of valid values, the field becomes red to indicate that an invalid value is being added and tool by default, assigns the default value for that particular field to it.

1.3.10.1 Configuring General Options

When you invoke the *Configuration* dialog box, the *General* options are shown by default.

The *General* category shows the following option:

- Open terminal command —To specify the default terminal window. By default, *Open terminal command* is specified as `xterm`. In case you want to change the terminal, specify it in the text box and click *OK*.
- My email address—To specify the email address of the person to whom the emails generated by vManager will be sent.
- Date Format Pattern—To specify an alternate format for specifying dates in date related fields/attributes. By default, the format is specified as `yyyy.mm.dd`.
- Email distribution list —To specify the email addresses of persons to whom mails should be sent. You can specify more than one email address by separating the email addresses with a comma (,). The value you specify here will be set as the value of all other email configuration options, such as *Session Done Report (Email)*, *Metrics Report Email*, *Sessions Report Email*, *Tests Report Email*, and *vPlan Report Email*.
- Source Editor —To specify an alternate source editor for opening source files. By default, `emacs` is used as the source editor for opening source files.

Note: Unlike `emacs` and `gvim`, the `vim` editor does not open a new terminal. As a result, if you want to use `vim` as the source editor, specify `xterm -e /usr/bin/vim` in the *Source Editor* field.

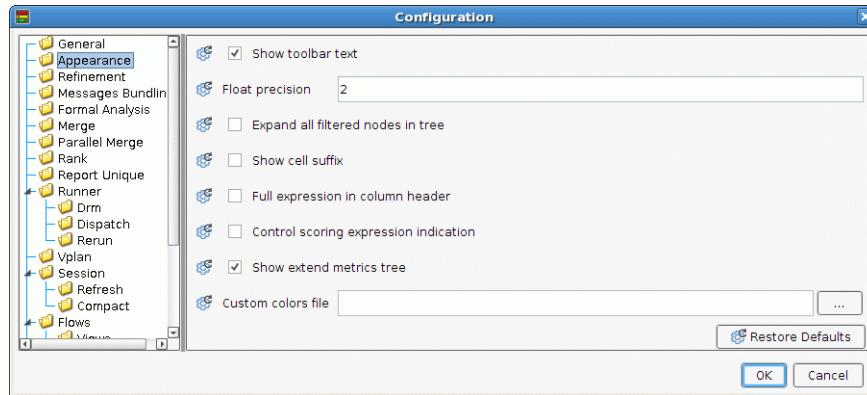
- Browser—To specify the path of the HTML browser. For example, you can specify the path as: `/usr/bin/firefox`.
- Abort execution upon error — By default, this check box is not selected. This indicates that execution is not aborted even if a command in the TCL file specified with the `-exec` option fails. When this check box is selected, then if any of the commands in this TCL file specified with the `-exec` option fails, the tool terminates without executing the remaining commands.
- Aggregation timeout (In GUI, measured in seconds)— To specify the aggregation timeout limit. By default, 180 seconds is specified. This indicates the aggregation calculation stops after 180 seconds. You can modify this limit, as required.

1.3.10.2 Configuring Appearance Options

To view and configure appearance related options, click *Appearance* in the left pane in the *Configuration* dialog box.

Figure 1-54 on page 57 displays the *Configuration* dialog box with appearance related options.

Figure 1-54 Configuration -- Appearance Options



The *Appearance* category shows the following options:

- Show toolbar text—To enable or disable showing of text in the toolbar. By default, in each toolbar, just below the icon button, text is also written. To hide the text that appear below the icon buttons on toolbar, clear the *Show toolbar text* check box.

[Figure 1-55](#) on page 57 shows the default setting of toolbar text.

Figure 1-55 Toolbar Text Enabled (Default Setting)



If you clear the *Show toolbar text* option in the *Configuration* dialog box, the toolbar will appear, as shown in [Figure 1-56](#) on page 57.

Figure 1-56 Toolbar Text Disabled



- Float precision —To specify the maximum number of decimal places that you want to see in the attributes that show values as decimals. By default, you see values with maximum 2 decimal places. You can increase or decrease the precision, as required. Valid range of values is 0 to 10. In case you specify a value outside the range of valid values, tool by default, assign the default value (which in this case is 2) to it.
- Expand all filtered nodes in tree —To enable or disable expanding of filtered nodes in the verification hierarchy tree. By default, expanding is turned off. To enable expanding of filtered nodes in the verification hierarchy tree select the *Expand all filtered nodes in tree* check box.

- Show cell suffix—To enable or disable showing the suffix of the attributes of type string. This option actually right justifies the content in the column. By default, this option is turned off.
- Full expression in header column—To enable or disable showing the expression in the column header in the Expression analysis page. By default, it shows the term numbers, such as, T1, T2, and so on. If you enable this option, then instead of T1, T2, the corresponding expression from the source code is shown.
- Control scoring expression indication — To enable or disable showing a * notation in front of the controlling terms in all expression coverage tables. By default, a * notation is not shown for identification of controlling terms. After you enable this option, a * notation appears next to the controlling terms in expression tables in GUI, HTML reports, and ASCII reports.

Note: This feature is enabled only if `set_expr_scoring -control` CCF command is used during simulation run. For more details on this command, see the *ICC User Guide*.

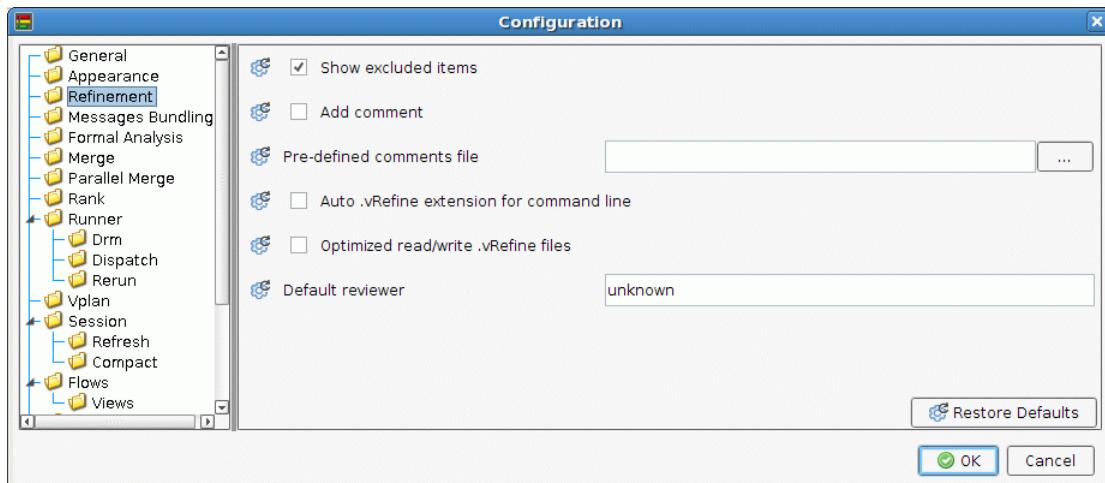
- Show extend metrics tree — To enable or disable showing extended metrics tree. By default, only instances and types are listed in the metrics hierarchy tree. After you enable this option, covergroups, cover items, FSMs, and assertions are also listed in the metrics hierarchy tree. In addition, only Relative Elements, Bins, and Toggle tab pages are shown in the *List tabs* pane of the *Metrics* page. By default, the *List tabs* pane of the *Metrics* page shows Relative Elements, Cover groups, Items, FSMs, Assertions, and Toggle tab pages.
- Custom colors file— To specify a custom color file that will be used for specifying chart colors.

1.3.10.3 Configuring Refinement Options

To view and configure refinement related options, click *Refinement* in the left pane in the *Configuration* dialog box.

Figure 1-57 on page 59 displays the *Configuration* dialog box with refinement related options.

Figure 1-57 Configuration -- Refinement Options



The *Refinement* category shows the following options:

- Show excluded items—To enable or disable showing of excluded items in the hierarchy. By default, excluded items appear in the hierarchy (though the coverage for that item is excluded from overall coverage). You can hide the excluded items by clearing the *Show excluded items* check box. For more details, see [Hide Excluded Items](#) on page 296.
- Add comment —To enable or disable adding comments at the time of exclusion. By default, the *Add comment* check box is not selected. This indicates that you will not be prompted to add comments at the time of exclusion. After you select the *Add comment* check box, you will be prompted to add comments at the time of exclusion. A dialog box will be displayed which will prompt you to add comments. For more details, see [Adding Comments at the Time of Exclusion](#) on page 292.
- Pre-defined comments file—To enable adding comments to exclusion from a pre-defined list of comments. A comments file is created as a .csv file in the following format:

```
NAME,COMMENT
<comment_name>,<comment_text>
<comment_name>,<comment_text>
```
- You can specify the path to the comments file in the *Pre-defined comments file* field. After this, at the time of applying exclusion, (in the *Exclusion* dialog box) you will see an option that will allow you to select the comment from the pre-defined comments that were defined in the comments file. For more details, see [Adding Comments at the Time of Exclusion](#) on page 292.
- Auto .vRefine extension for command line —To automatically add .vRefine as the file extension if the user has not explicitly specified it at the time of saving the refinement file.

Note: This option affects only the batch mode. In GUI, the .vRefine extension is always added automatically, even if the user has not explicitly specified it.

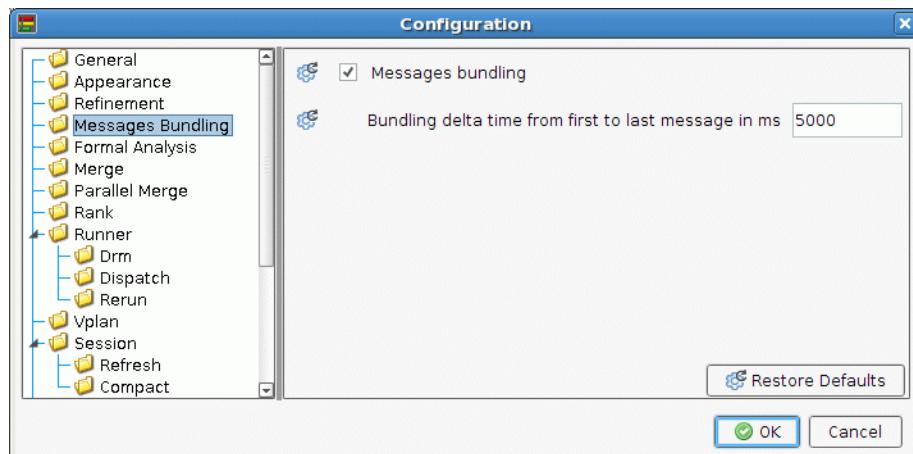
- Optimized read/write .vRefine files —To optimize the tool performance for read/write operation on a large .vRefine file. By default, this option is not selected. It is recommended that if the refinement file is relatively large, then you select this check box to optimize tool performance.
- Default reviewer—To specify the default reviewer for adding comments. By default, the reviewer is specified as unknown. You can change the reviewer, as required. The name you specify here will appear in the Exclusion dialog box when you want to add comments at the time of exclusion. For more details, see [Adding Comments at the Time of Exclusion](#) on page 292.

1.3.10.4 Configuring Messages Bundling Options

To view and configure message bundling related options, click *Messages Bundling* in the left pane in the *Configuration* dialog box.

[Figure 1-58](#) on page 60 displays the message bundling related options.

Figure 1-58 Configuration -- Messages Bundling Options



The *Messages Bundling* category shows the following options:

- **Messages bundling**—To bundle messages of same type into a single message. By default, message bundling is turned on.
- **Bundling delta time from first to last message in ms**—To specify the maximum gap between messages for them to be bundled.

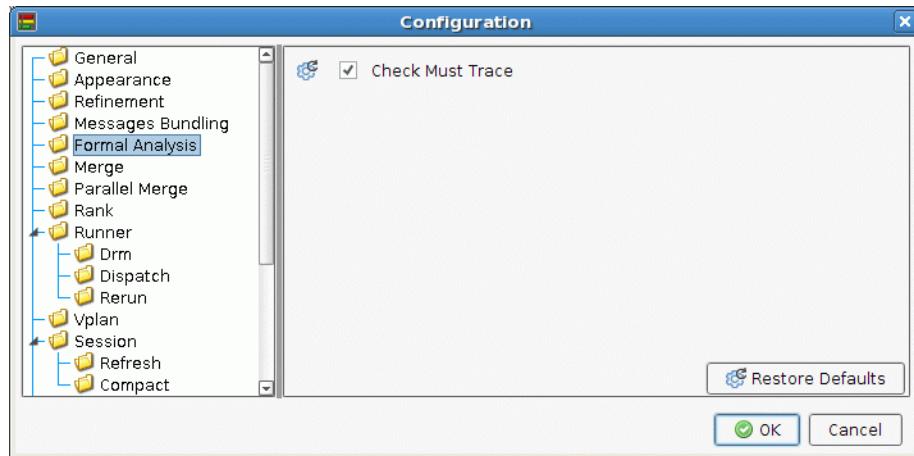
Note: These two options are related to the *Messages* dialog box that can be opened by clicking the envelope button at the bottom right.

1.3.10.5 Configuring Formal Analysis Options

To view and configure formal analysis related options, click *Formal Analysis* in the left pane in the *Configuration* dialog box.

[Figure 1-59](#) on page 61 displays the *Configuration* dialog box with *Formal Analysis* related options.

Figure 1-59 Configuration -- Formal Analysis Options



The *Formal Analysis* category has following options:

- **Check Must Trace** —This option indicates if the assertion properties must have a trace in order to be marked as *Proved*. By default, this option is enabled.
 - If enabled, the assertions must have a trace for the status to be marked as Proved.
 - If disabled, the assertion status can be marked as Proved without a trace.

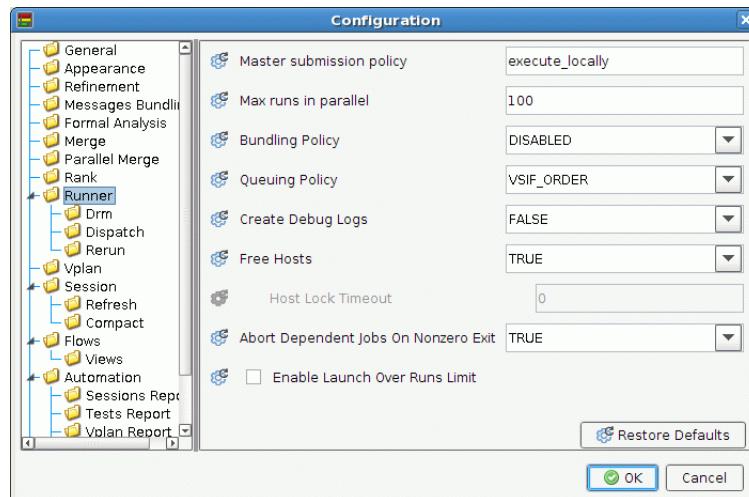
Note: This option impacts the *Formal Status Grade* calculation.

1.3.10.6 Configuring Runner Options

To view and configure runner related options, click *Runner* in the left pane in the *Configuration* dialog box.

[Figure 1-60](#) on page 62 displays the *Configuration* dialog box with runner related options.

Figure 1-60 Configuration -- Runner Options



For more details, see [Incisive vManager Installation and Configuration Guide](#).

Below the Runner folder in the left pane, you would see following subfolders:

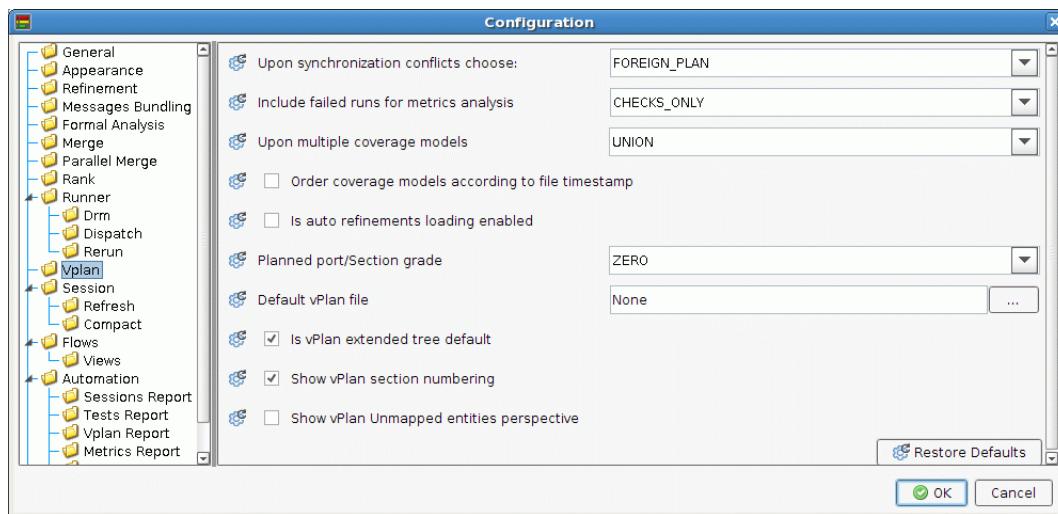
- Drm —To set DRM related options. For more details, see [Incisive vManager Installation and Configuration Guide](#).
- Dispatch—To set dispatch related options. For more details, see [Incisive vManager Installation and Configuration Guide](#).
- Rerun—To set rerun related options. For more details, see [Configuring Rerun Options](#) on page 148.

1.3.10.7 Configuring Vplan Options

To view and configure vPlan related options, click *Vplan* in the left pane in the *Configuration* dialog box.

[Figure 1-61](#) on page 63 displays the *Configuration* dialog box with vPlan related options.

Figure 1-61 Configuration -- Vplan Options



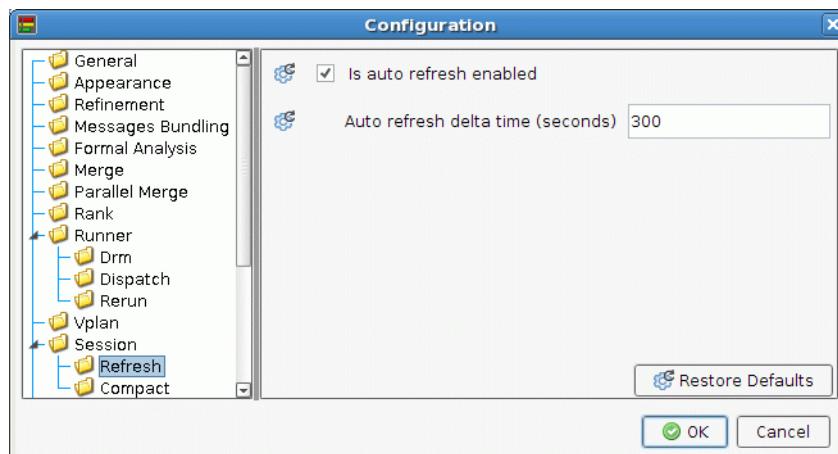
For more details, see [vPlan Configuration Options](#) on page 270.

1.3.10.8 Configuring Session Refresh Options

To view and configure session refresh options, click *Refresh* under the *Session* folder in the left pane in the *Configuration* dialog box.

[Figure 1-62](#) on page 63 displays the *Configuration* dialog box with refresh options.

Figure 1-62 Configuration -- Session Refresh Options



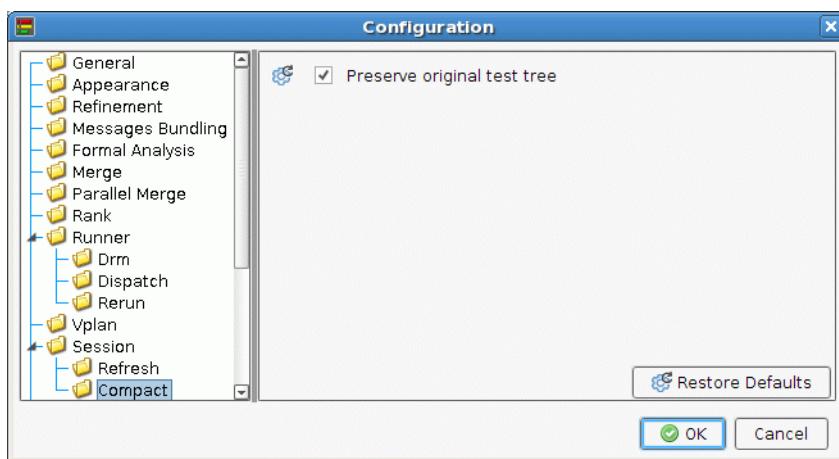
For more details, see [Refreshing Sessions](#) on page 96.

1.3.10.9 Configuring Session Compact Options

To view and configure session compaction options, click *Compact* under the Session folder in the left pane in the *Configuration* dialog box.

[Figure 1-63](#) on page 64 displays the *Configuration* dialog box with compaction options.

Figure 1-63 Configuration -- Session Compact Options



The *Compact* category has following options:

- **Preserve original test tree** —This option is selected by default. It indicates that the original test tree is preserved when sessions are compacted. In case you do not want to preserve the original test tree upon compaction, clear the *Preserve original test tree* check box in the *Configuration* dialog box.

For details on session compaction, see [Creating Compact Sessions](#) on page 111.

1.3.10.10 Configuring Flows Views Options

To view and configure flows views options, click *Views* under the Flows folder in the left pane in the *Configuration* dialog box.

[Figure 1-64](#) on page 65 displays the *Configuration* dialog box with flows views options.

Figure 1-64 Configuration -- Flows Views Options



By default, when you launch analysis pages for vPlan, metrics, tests, runs, failures, and incomplete runs, and so on the default view to be used while launching the page is already set. You can change the default view by selecting the appropriate option in the *Views* options page of the *Configuration* dialog box and click *OK*.

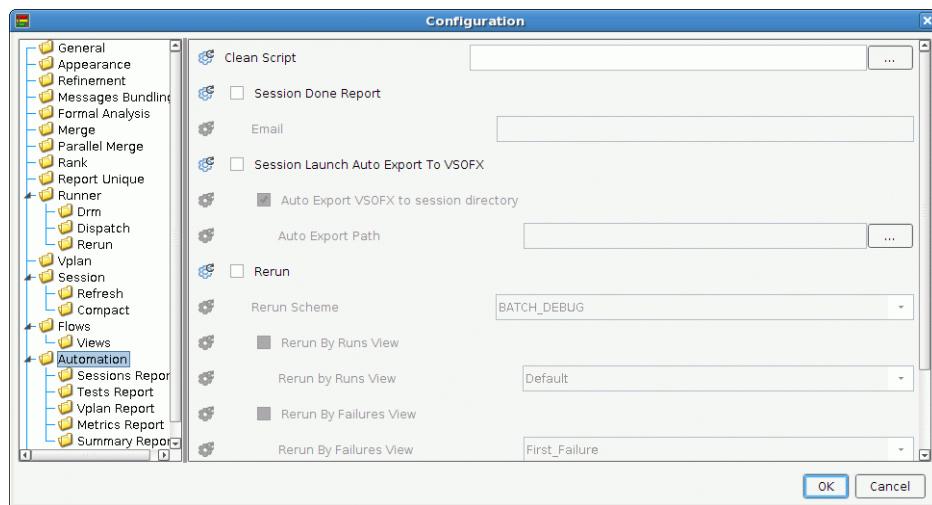
This will set the desired view as the default view when you launch the analysis page.

1.3.10.11 Configuring Automation Options

To view and configure automation related options, click *Automation* folder in the left pane in the *Configuration* dialog box.

[Figure 1-65](#) on page 66 displays the *Configuration* dialog box with automation related options.

Figure 1-65 Configuration -- Automation Options



The *Automation* folder also has subfolders for specifying automation of metrics, sessions, tests, and vPlan reports.

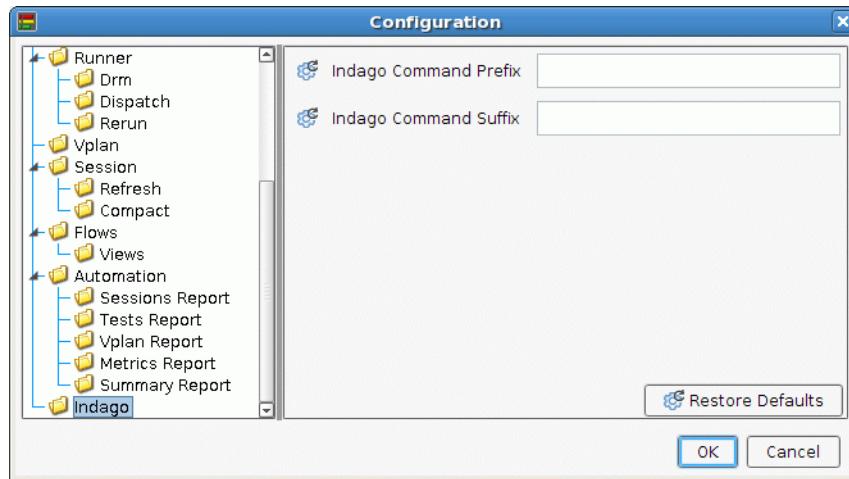
For more details, see [Incisive vManager Installation and Configuration Guide](#).

1.3.10.12 Configuring INDAGO Options

To view and configure INDAGO related options, click *Indago* folder in the left pane in the *Configuration* dialog box.

[Figure 1-66 on page 67](#) displays the *Configuration* dialog box with INDAGO related options.

Figure 1-66 Configuration -- Indago Options



The Indago folder has following options:

- Indago Command Prefix—To specify a prefix of the `debug_analyzer` command. This option can be used to launch debug analyzer in LSF.
- Indago Command Suffix —To specify a suffix of the `debug_analyzer` command. This option can be used to specify additional options to the `debug_analyzer -db <db>` command.

Note: With the above Indago related options, the command to invoke `debug_analyzer` would be:

```
<INDAGO Command Prefix> debug_analyzer -db <db> <INDAGO Command Suffix>
```

For example:

```
INDAGO Command Prefix = bsub  
INDAGO Command Suffix = -sim_env
```

When the user opens Indago from vManager, the invoked command would be:

```
bsub debug_analyzer -db <db> -sim_env
```

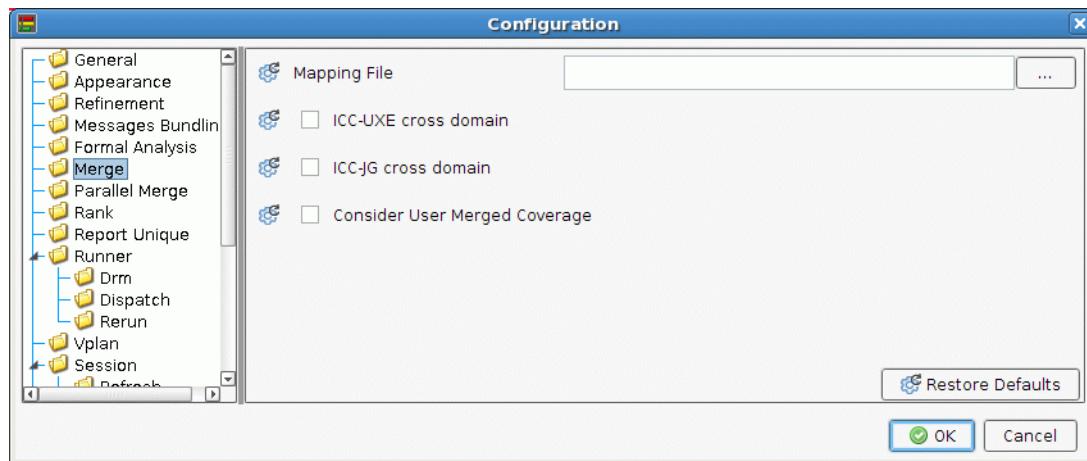
For more details on Indago, see [Integrating vManager with Indago Debug Analyzer](#) on page 209.

1.3.10.13 Configuring Merge Options

To view and configure merge related options, click *Merge* folder in the left pane in the *Configuration* dialog box.

[Figure 1-67](#) on page 68 displays the *Configuration* dialog box with merge related options.

Figure 1-67 Configuration -- Merge Options



The *Merge* folder has following options:

- **Mapping File**—Allows you to specify the entities mapping file for merge operations. This file may include one or more `merge_config` commands. For more details on the `merge_config` commands, see the [Incisive Metrics Center User Guide](#). The file that you specify here is read and considered while performing the merge operation. If `merge_config` command is run from the command line and a mapping file is also specified here, then both are considered at the time of merge.
- **ICC-UXE cross domain**—Allows you to merge ICC and UXE runs into the same hierarchy. For more details, see [Analyzing Coverage Coming from Palladium](#) on page 235.
- **ICC-JG cross domain**—Allows you to merge ICC and JG runs into the same hierarchy. For more details, see [Analyzing Coverage Coming from JasperGold](#) on page 237.
- **Consider User Merged Coverage**—Allows you to enable considering of user-specified merged coverage in operations that require a merge. By default, vManager uses an internal automatic merge mechanism for any operation that requires merge. Instead of automatic merge, you can associate/attach previously merged data with a session and use it for analysis. However, the data that you attach is not used until you select the *Consider User Merged Coverage* check box. For more details on how to associate already merged data with a session, see [Associating Merged Coverage with Sessions](#) on page 110.

Merge Messages in vManager

In vManager, merge operations are not initiated directly by the user; however, they are executed as part of wider operations such as, analyze metrics, collect runs, and so on.

During a merge operation, following type of messages are printed:

- Type 1: Info messages which describe the current scope of the merge operation.
- Type 2: Merge conflict messages which are originated by the unicov library and are registered using unicov API code.
- Type 3: Summary messages which show the number of conflicts during model union merge (target model creation), and during data projection.
- Type 4: Merge fatal error messages that are printed when merge cannot be done.

Merge Messages in Log Files

Messages mentioned in *Type 1* and *Type 2* are written in log files created under:

<current_working_dir>/vmgr_merge_logs/pid_<VMGR-process-id>

Note: The `vmgr_merge_logs` directory or any of its sub-directories are not automatically deleted by vManager. You can delete them, as required.

Merge Messages in Messages Window or STDOUT

The following messages are printed in the *Messages* window in GUI or on stdout in batch mode:

- Messages mentioned in *Type 3* that is the merge summary messages to show any merge conflicts or errors (not printed in case there are not any warnings or errors).
- Messages mentioned in *Type 4* that is the merge fatal messages when merge cannot be done. The message also points to the merge log file under `vmgr_merge_logs`. For example:

Merge operation (<source origin of merge>) is aborted:

<Merge error message as appears in IMC>

Merge operation log file: - vmgr_merge_logs/pid_<VMGR-process-id>/<file>.log

Note: These error messages are added to the log files also (without the portion that indicates the pointer to the log file).

- A message that notifies that a merge is being done, and the reason for that (not printed in case there are not any merge conflicts). The message also points to the merge log file under `vmgr_merge_logs`. For example:

Merge operation (<source origin of merge>) is done.

Number of conflicts during target model creation:1

Number of conflicts during data projection:1

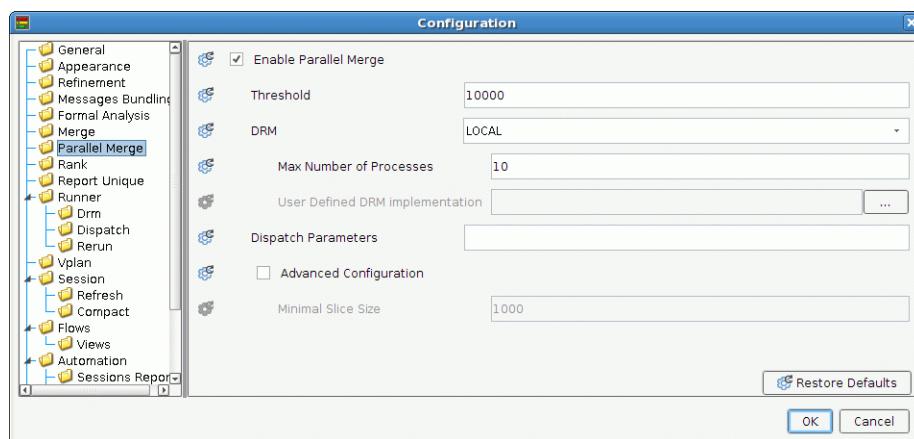
Merge operation log file - vmgr_merge_logs/pid_<VMGR-process-id>/<file>.log

1.3.10.14 Configuring Parallel Merge Options

To view and configure parallel merge related options, click *Parallel Merge* folder in the left pane in the *Configuration* dialog box.

[Figure 1-68](#) on page 70 displays the *Configuration* dialog box with parallel merge related options.

Figure 1-68 Configuration -- Parallel Merge Options



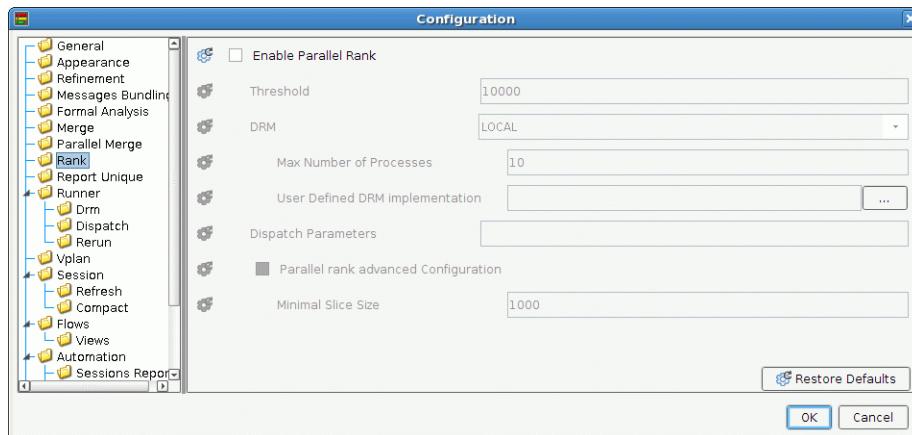
For more details, see [Incisive vManager Installation and Configuration Guide](#).

1.3.10.15 Configuring Parallel Rank Options

To view and configure parallel ranking related options, click *Rank* folder in the left pane in the *Configuration* dialog box.

[Figure 1-69](#) on page 71 displays the *Configuration* dialog box with parallel rank related options.

Figure 1-69 Configuration -- Rank Options



For more details, see [Enable Parallel Ranking](#) on page 192.

1.3.11 User-Defined Actions

vManager GUI can be extended to allow performing of user-defined actions.

The user-defined actions are defined in a TCL script and once they are registered, they appear as a button in GUI on the *Scripts* toolbar. These buttons are supported for following entities/pages:

Page	Entities
Regression Center	Session
Tests Analysis Page	Run, Warning, Error
Runs Analysis Page	Run, Warning, Error

This section covers the following topics:

- [Create a TCL Script with User-defined Action](#) on page 71
- [Launch GUI and Register the TCL script](#) on page 72

1.3.11.1 Create a TCL Script with User-defined Action

You must first create a TCL script with the required user-defined action. The TCL script must contain a procedure named `execute` with a single parameter named `attributes`.

For example, to create a user action for compressing a session directory, write the following code in a TCL file:

```
# A vManager action script to compress session dir
proc execute {attributes} {
    upvar attributes attrs;
    #compression is done in compress.sh which accept session dir arg
    exec compress.sh $attrs(session_dir);
}
```

You can now save the TCL file as `compress.tcl`.

In addition to the `execute` procedure, two additional optional procedures can be defined in the TCL file: `startAction` and `endAction`. These procedures accept no arguments. You can use these procedures to perform calculation based on all of the selected entities.

Note: The `startAction` procedure is called first, then the `execute` procedure is called for any selected entity, and finally the `endAction` procedure is called.

For example, you can use the following TCL code to define a user action named *Sum Runs* to calculate the sum of all runs for sessions. The relevant entity in this case would be: *Session*.

```
# A vManager action script to file bug based on a message
set sum 0;
proc execute {attributes} {
    upvar attributes attrs;
    set ::sum [ expr $::sum + $attrs(RUNS) ];
}
proc endAction {} {
    puts $::sum;
}
```

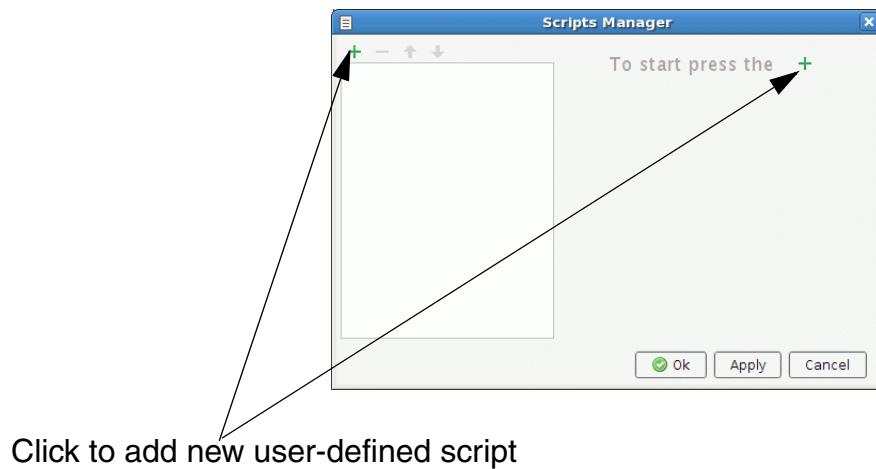
1.3.11.2 Launch GUI and Register the TCL script

After you have created a user-defined action in a TCL file, you need to register it using vManager GUI. To register the TCL script:

1. Launch vManager GUI.
2. Go to Regression center or Analysis center.
3. In the *Scripts* toolbar, click the *Scripts Manager* button.

The *Scripts Manager* dialog box is displayed, as shown in [Figure 1-70](#) on page 73.

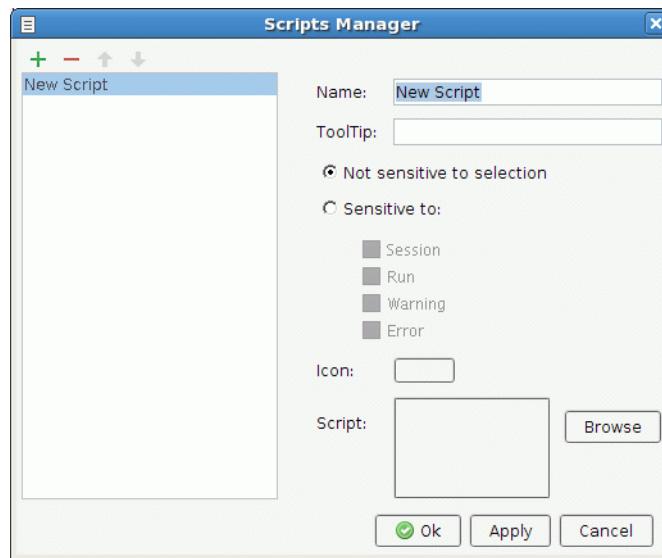
Figure 1-70 Scripts Manager



4. Click the + button to add a new user-defined script.

Figure 1-70 on page 73 shows the *Scripts Manager* dialog box with the required fields for registering the script.

Figure 1-71 Scripts Manager



5. In the *Name* field, specify the name of the action. The name you specify will appear in the GUI. For example, specify the name as *Compress*.
6. In the *ToolTip* field, specify some description of the action. The text you specify here, will appear as the tooltip when you move the cursor over the action button that will be created

in GUI. For example, specify tooltip text as *-- Compress session dir and delete it from file system.*

7. By default, the *Not sensitive to selection* radio button is selected. This indicates that the user-defined action that you are registering is not sensitive or dependent on selection of any entity (Session, run, warning, error).
8. If you want the user-defined action to be relevant on all or limited entities, select the *Sensitive to* radio button. This will enable the following check boxes:

- Session
- Run
- Warning
- Error

Select the required check boxes. For example, select *Session* and *Run* check boxes.

9. To specify the icon that would be shown in the GUI for this action, click the *Icon* button.
10. The *Choose Icon* dialog box appears. Navigate through the hierarchy and select the relevant file. For example, select `compress.png` and click *OK*.

Note: The script icon file must be of type, PNG, JPG, or GIF format. In addition, if the size of the icon is bigger than 16x16, then vManager automatically minimizes it to 16x16.

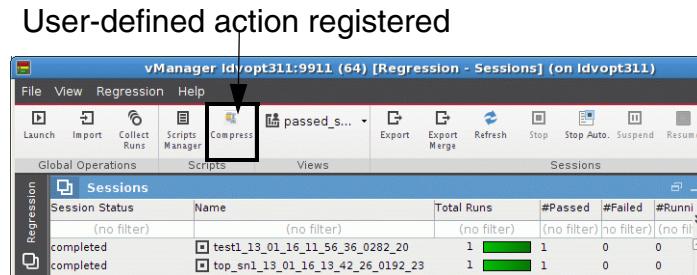
11. You will return to the *Scripts Manager* dialog box. Now, select the *Browse* button next to the *Script* field.
12. The *Choose Script* dialog box appears. Navigate through the hierarchy and select the TCL file in which the user-defined action is defined. For example, select `compress.tcl` and click *OK*. After you click OK, the script text shows in the *Script* field.

Note: If there are any syntax errors in the TCL file, then the errors are reported and registration fails.

13. Click *OK* to close the *Scripts Manager* dialog box.

The user-defined action that you registered will now be displayed as a new button on the *Scripts* toolbar, as shown in [Figure 1-72](#) on page 75.

Figure 1-72 User-defined Action Registered



When you click the newly created *Compress* button, then user-defined TCL script *compress.tcl* will be invoked.

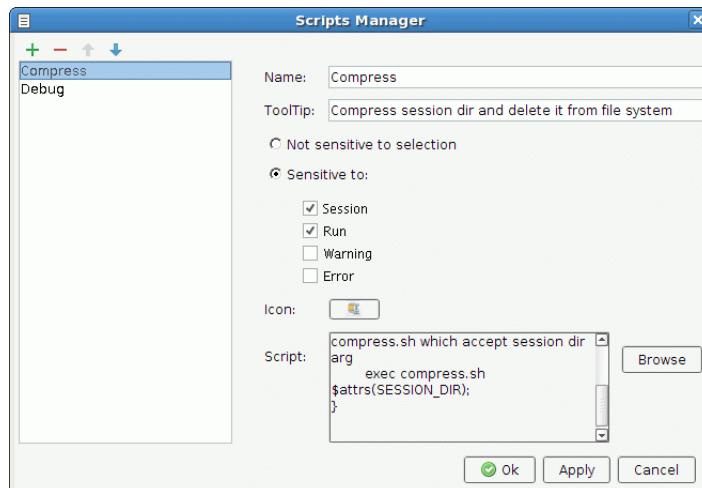
Similarly, you can define and register more user-defined actions.

Note: If you select multiple entities and then click the user-defined action button, then the TCL script is executed for each of the selected entities. If grouped entities are selected (grouping was done in the table), then the TCL script will be executed for each entity in the selected group.

Other Actions on the Scripts Manager Dialog Box

[Figure 1-73 on page 75](#) shows the *Scripts Manager* dialog box with few registered scripts.

Figure 1-73 Scripts Manager



- Click the “–” button to delete the selected script.
- Click the Up arrow and Down arrow buttons to move the selected script accordingly.

Incisive vManager User Guide

Note: Changing the order of scripts in the *Scripts Manager* dialog box will also change its location in the *Scripts* toolbar.

You can also edit the properties of the selected script.

Loading and Viewing Sessions

The *Regression* center of the vManager allows you to load sessions and view the details of various runs and failures.

This chapter covers the following topics:

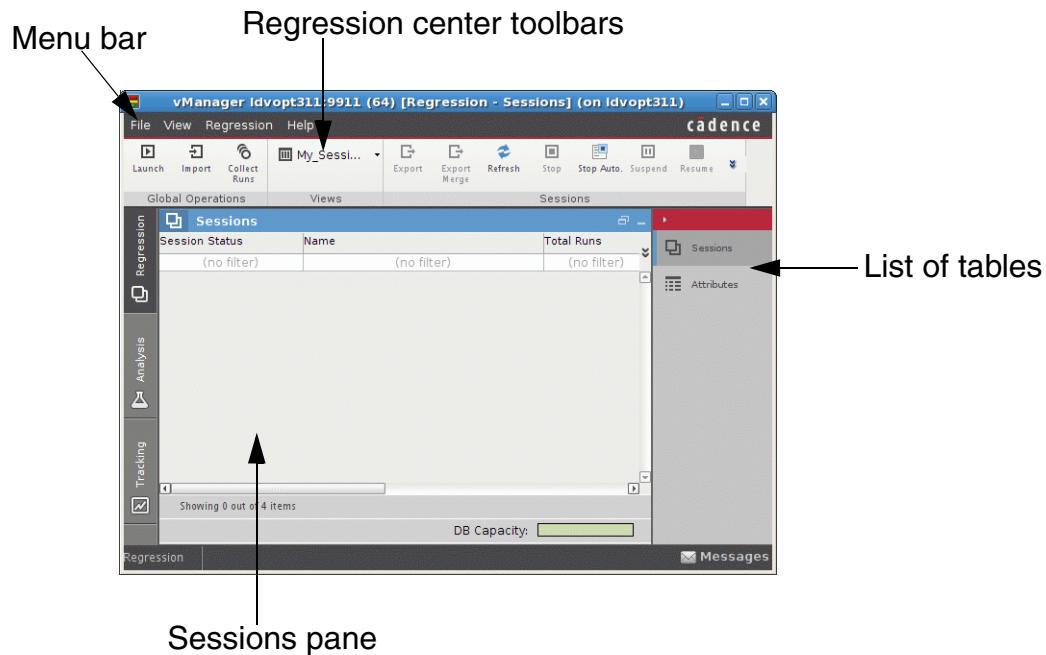
- [Regression Center Interface](#) on page 78
- [Loading Sessions](#) on page 87
- [Exporting Sessions](#) on page 91
- [Exporting Merged Coverage of Sessions](#) on page 92
- [Collecting Runs](#) on page 93
- [Refreshing Sessions](#) on page 96
- [Opening Session Directory in Terminal](#) on page 97
- [Viewing Session Information](#) on page 98
- [Stopping Sessions](#) on page 100
- [Stopping Session Based on Conditions](#) on page 100
- [Suspend / Resume Sessions](#) on page 104
- [Deleting Sessions](#) on page 105
- [Relocating Session](#) on page 105
- [Handling Inaccessible Sessions](#) on page 106
- [Editing Session Details](#) on page 108
- [Associating Merged Coverage with Sessions](#) on page 110
- [Creating Compact Sessions](#) on page 111
- [Rollup Attributes to Session Level](#) on page 117

- [Generating Charts](#) on page 123
- [Taking Snapshots](#) on page 125
- [Generating Sessions Report](#) on page 127
- [Invoking Analysis Center](#) on page 131

2.1 Regression Center Interface

[Figure 2-1](#) on page 78 displays the *Regression* center.

Figure 2-1 vManager Regression Center Interface



The main components of the *Regression* center are:

- [Menu Bar](#)
- [Toolbars](#)
- [Sessions Pane](#)
- [Attributes Pane](#)

Note: By default, when you launch Regression center, the *Attributes* pane/table is hidden. You can click Attributes table from the list of tables in the right pane to show the Attributes table in the view area.

2.1.1 Menu Bar

[Figure 2-2](#) on page 79 shows the menu bar available in the *Regression* center.

Figure 2-2 Menu Bar



Menu options *File*, *View*, and *Help* are common to all activity centers and are discussed in the [Chapter 1, “Getting Started with vManager”](#).

The *Regression* menu is related to the *Regression* center and is discussed here.

From the *Regression* menu, you can select any of the following options:

- [Launch](#)—To load a session (VSIF file).
- [Import](#)—To load an already generated session (VSOF or VSOFX file).
- [Collect Runs](#)—To create a new session (a single run vsop) by collecting runs result from the disk.
- [Scripts Manager](#)—To add user-defined actions on toolbar. For more details, see [User-Defined Actions](#) on page 71.
- [Export](#)—To export a session from the database to a VSOFX file.
- [Export Merge](#)—To export merged coverage of a session to a specified location.
- [Refresh](#)—To refresh the *Sessions* pane.
- [Stop](#)—To stop the currently running session.
- [Stop Automation](#)—To define rules for stopping sessions based on specific conditions.
- [Suspend](#)—To suspend the currently running session. Applicable only on sessions launched with new runner.
- [Resume](#)—To resume the selected suspended session.
- [Delete](#)—To delete the selected session.
- [Relocate](#)—To relocate the selected session.

- Open Session Directory—To open a terminal window and show the selected session as the current working directory.
- Session Info—To view the details of the selected session.
- Recalculate UDA Script—To recalculate values of computed user-defined attributes in the selected session. For more details, see [Recalculate UDA](#) on page 122.
- Create chart—To generate all sessions chart or selected sessions chart. For more details, see [Generating Charts](#) on page 123.
- Take snapshot—To take snapshot from Regression center.
- Edit each—To add comments or description to the selected session one by one.
- Edit all at once—To add comments or description to all the selected sessions together.
- Attribute change history—To view changes made to attributes of the selected session.
- Analyze Metrics—To invoke analysis center and open the Metrics page for analysis.
- Analyze Tests—To invoke analysis center and launch the tests tree.
- Analyze vPlan—To invoke analysis center and prompt for launching a verification plan.
- Analyze Failures—To invoke analysis center for analyzing session failures.
- Analyze All Runs—To invoke analysis center for analyzing all runs.
- Analyze Pass Runs—To invoke analysis center for analyzing passed runs.
- Analyze Failed Runs—To invoke analysis center for analyzing failed runs.
- Analyze Incomplete Runs—To invoke analysis center for analyzing incomplete runs.
- Analyze Formal Properties—To invoke analysis center for analyzing formal properties. For more details, see [Analyzing Formal Properties](#) on page 172.
- Reports—To generate session reports.

2.1.2 Toolbars

The following toolbars are available in the *Regression* center of vManager:

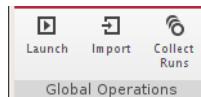
- Global Operations
- Scripts
- Views

- [Sessions](#)
- [Attributes](#)
- [Analyze](#)
- [Report](#)
- [Tracking](#)

2.1.2.1 Global Operations

[Figure 2-3 on page 81](#) shows the *Global Operations* toolbar.

Figure 2-3 Global Operations Toolbar



The *Global Operations* toolbar has the following options:

- [Launch](#)—To load a session (VSIF file)
- [Import](#)—To load an already generated session (VSOF or VSOFX file)
- [Collect Runs](#)—To create a new session (a single run vsop) by collecting runs result from the disk

2.1.2.2 Scripts

[Figure 2-4 on page 81](#) shows the *Scripts* toolbar.

Figure 2-4 Scripts Toolbar



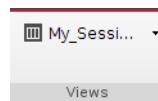
The *Scripts* toolbar has the following options:

- [Scripts Manager](#)—To add user-defined actions on toolbar. For more details, see [User-Defined Actions on page 71](#).

2.1.2.3 Views

[Figure 2-5 on page 82](#) shows the *Views* toolbar.

Figure 2-5 Views Toolbar



The *Views* toolbar allows you to define views, organize views, set views, and quickly launch a view. For more details, see [Defining and Organizing Views on page 43](#).

2.1.2.4 Sessions

[Figure 2-6 on page 82](#) shows the *Sessions* toolbar.

Figure 2-6 Sessions Toolbar



The *Sessions* toolbar has following options:

- [Export](#)—To export a session from the database to a VSOFX file.
- [Export Merge](#)—To export merged coverage of a session to a specified location.
- [Refresh](#)—To refresh the *Sessions* pane
- [Stop](#)—To stop the currently running session
- [Stop Automation](#)—To define rules for stopping sessions based on specific conditions.
- [Suspend](#)—To suspend the currently running session. Applicable only on sessions launched with new runner.
- [Resume](#)—To resume the selected suspended session.
- [Delete](#)—To delete the selected session
- [Relocate](#)—To relocate the selected session
- [Open dir](#)—To open a terminal window and show the selected session as the current working directory

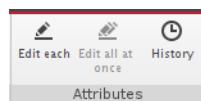
- Session Info—To view the details of the selected session
- Recalc UDA —To recalculate values of computed user-defined attributes in the selected session. For more details, see Recalculate UDA on page 122.
- Create chart—To generate all sessions chart or selected sessions chart. For more details, see Generating Charts on page 123.

Note: You can also select the above options from the *Regression* menu and the right-click menu.

2.1.2.5 Attributes

The *Attributes* toolbar, as shown in [Figure 2-7](#) on page 83 allows you to edit or add details, such as comments, description, editable user-defined attributes, and owner of the session. It also allows you to view attribute change history.

Figure 2-7 Attributes Toolbar



The *Attributes* toolbar has following options:

- Edit each—To add comments or description to the selected session one by one
- Edit all at once—To add comments or description to all the selected sessions together
- History—To view attribute change history

Note: You can also select the above options from the *Regression* menu.

For more details on editing sessions, see [Editing Session Details](#) on page 108.

2.1.2.6 Analyze

[Figure 2-8](#) on page 83 shows the *Analyze* toolbar.

Figure 2-8 Analyze Toolbar



The *Analyze* toolbar has following options that allow you to analyze the session:

- Metrics—To invoke *Analysis* center and open the *Metrics* page for analysis. This creates as separate context in the *Analysis* center.
- Tests —To invoke *Analysis* center and launch tests tree. This creates as separate context in the *Analysis* center with all the runs from the selected session.
- vPlan—To invoke *Analysis* center and prompt for launching a verification plan. This creates as separate context in the *Analysis* center.
- Failures—To invoke *Analysis* center for analyzing session failures. This opens the *First_Failure* view of *Analysis* center.
- All Runs—To invoke *Analysis* center for analyzing all runs. When you select this option, all the runs are shown in the *Analysis* center.
- Passed Runs—To invoke *Analysis* center for analyzing passed runs. When you select this option, only the passed runs are shown in the *Analysis* center.
- Failed Runs—To invoke *Analysis* center for analyzing failed runs. When you select this option, only the failed runs are shown in the *Analysis* center.
- Incomplete Runs—To invoke analysis center for analyzing incomplete runs. When you select this option, only the incomplete runs are shown in the *Analysis* center.
- Formal Properties—To invoke analysis center for analyzing formal properties. For more details, see [Analyzing Formal Properties](#) on page 172.

Note: You can also select the above options from the *Regression* menu.

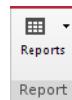
Note: Metrics, vPlan, Ranking, and correlation contexts will show disabled if the underlying coverage model is extremely large.

For more details, see [Analyzing Runs and Failures](#) on page 133, [Analyzing Metrics](#) on page 217, and [Analyzing Verification Plans](#) on page 251.

2.1.2.7 Report

[Figure 2-9](#) on page 84 shows the *Report* toolbar.

Figure 2-9 Report Toolbar



The *Report* toolbar has following option:

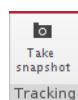
- Report—To generate session reports

Note: You can also select the above options from the *Regression* menu.

2.1.2.8 Tracking

[Figure 2-10 on page 85](#) shows the *Tracking* toolbar.

Figure 2-10 Tracking Toolbar



The *Tracking* toolbar has following option:

- Take snapshot—To take snapshot from Regression center.

Note: You can also select the above option from the *Regression* menu.

2.1.3 Sessions Pane

The *Sessions* pane of the *Regression* center shows the status of the session. It shows the details such as, the name of the session, the number of runs, number of passed runs, failed runs, and so on.

[Figure 2-11 on page 85](#) shows the *Sessions* pane.

Figure 2-11 Sessions Pane with the Loaded vsosf

Session Status	Name	Total Runs	#Pa
(no filter)	(no filter)	(no filter)	(no filter)
completed	test1_13_01_16_11_56_36_0282_20	1	1
completed	top_sn1_13_01_16_13_42_26_0192_23	1	1
completed	vm_uart_ctrl_regression.johnn.12_01_...	12	4
completed	cdn_uart_segal.11_01_04_23_14_13_1...	18	16

The *Sessions* pane shows the details such as, the name of the session, the number of runs, number of passed runs, failed runs, and so on.

2.1.4 Attributes Pane

The *Attributes* pane of the *Regression* center shows a complete list of attributes associated with the selected session. [Figure 2-12](#) on page 86 shows the *Attributes* pane.

Figure 2-12 Attributes Pane

The screenshot shows a table titled "Attributes" with a session name "test1_06_01_16_16_18_04_08" displayed above it. The table has three columns: "Col #", "Name", and "Value". The "Name" column contains attribute names like "#Failed", "#Other", "#Passed", "#Running", "#Waiting", and "Abort Dependent Jobs ...". The "Value" column contains their corresponding numerical values. Arrows point to specific parts of the table with labels: "Session name" points to the title bar, "Column number" points to the "Col #" header, "Attribute's name" points to the "Name" column header, and "Attribute's value" points to the "Value" column header.

Col #	Name	Value
	(no filter)	(no filter)
4	#Failed	0
7	#Other	0
3	#Passed	1
5	#Running	0
6	#Waiting	0
	Abort Dependent Jobs ...	TRUE

The *Attributes* pane shows all of the attributes associated with the selected session.

The *Col #* column of the *Attributes* pane shows the column number of the corresponding attribute. For example, a value 1 in the *Col #* column indicates that this column appears at the first place in the *Sessions* pane. No value in the *Col #* column indicates that the corresponding attribute is not displayed in the *Sessions* pane.

You can right-click on an attribute and select any of the following:

- Add Columns—To add the selected column in the *Sessions* pane
 - Remove Columns—To remove the selected column from the *Sessions* pane
- Note:** Alternatively, you can double-click on an attribute to add it to the *Sessions* pane, and then double-click it again to remove it from the *Sessions* pane.
- Unfilter Table—To release an already applied filter
 - Unsort Table—To unsort the table data
 - Copy Cell—To copy the data in the selected cell and paste it in any editor outside of vManager
 - Copy Row—To copy the data in the selected row and paste it in any editor outside of vManager

2.2 Loading Sessions

Using vManager, you can either launch a VSIF file or load an already generated session data (VSOF file).

- [Launching a Session \(VSIF\)](#)
- [Importing Sessions](#)

2.2.1 Launching a Session (VSIF)

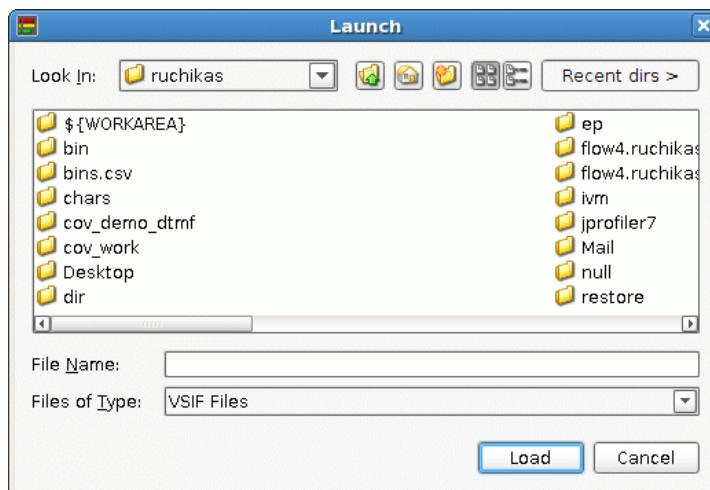
The *Launch* button allows you to load a session (VSIF file). When you load a session, the session is run and the results start showing in the *Sessions* pane.

To launch a session (VSIF file):

1. Click the *Launch* button on the *Global Operations* toolbar.

The *Launch* dialog box is displayed, as shown in [Figure 2-13](#) on page 87.

Figure 2-13 Launch VSIF File

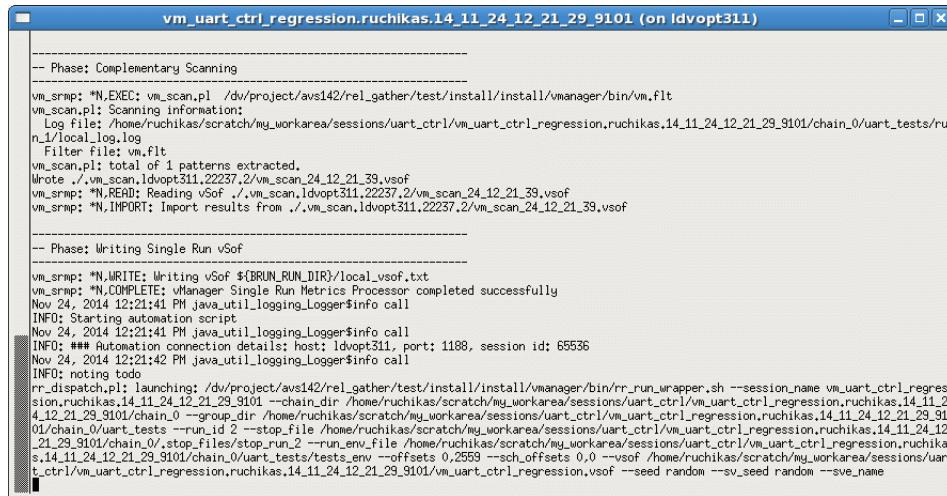


2. Navigate through the hierarchy and select the relevant VSIF file. The VSIF files has a VSIF icon so it can be recognized easily in the file chooser. For example, to launch the VSIF file named `uart_regress_demo.vsif` select `uart_regress_demo.vsif` and click *Load*.

Note: By default, the dialog box shows the last directory you visited. In order to navigate to a different directory, click the *Recent dirs* button. If you navigate to a new directory, it is saved automatically under the list of quick links and becomes the new default.

An additional window as shown in [Figure 2-14](#) on page 88 is displayed showing the progress of runs within the session.

Figure 2-14 Session Progress



```
-- Phase: Complementary Scanning
vm_smp: *NEXEC: vm_scan.pl /dv/project/avs142/rel_gather/test/install/install/vmanager/bin/vm.filt
vm_scan.pl: Scanning information:
  Log file: /home/ruchikas/scratch/my_workarea/sessions/uart_ctrl/vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101/chain_0/uart_tests/run_1/local.log.log
  Filter file: vm.filt
vm_scan.pl: total of 1 patterns extracted.
Wrote ./vm_scan.ldvopt311.22237.2/vm_scan.24_12_21_39.vsof
vm_smp: *NREAD: Reading vsof ./vm_scan.ldvopt311.22237.2/vm_scan.24_12_21_39.vsof
vm_smp: *NIMPORT: Import results from ./vm_scan.ldvopt311.22237.2/vm_scan.24_12_21_39.vsof

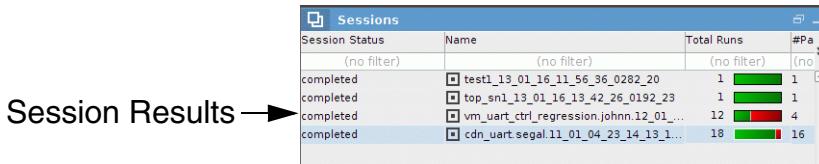
-- Phase: Writing Single Run vsof
vm_smp: *NWRITE: Writing vsof ${BRUN_RUN_DIR}/local_vsof.txt
vm_smp: *NCOMPLETE: vManager Single Run Metrics Processor completed successfully
Nov 24, 2014 12:21:41 PM java.util.logging.Logger$info call
INFO: Starting automation script
Nov 24, 2014 12:21:41 PM java.util.logging.Logger$info call
INFO: ### Automation connection details: host: ldvopt311, port: 1188, session id: 65536
Nov 24, 2014 12:21:42 PM java.util.logging.Logger$info call
INFO: noting todo
rr_dispatch.pl: launching: /dv/project/avs142/rel_gather/test/install/install/vmanager/bin/rr_run_wrapper.sh --session_name vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101 --chain_id /home/ruchikas/scratch/my_workarea/sessions/uart_ctrl/vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101/chain_0/stop_files/stop_run_2 --run_env_file /home/ruchikas/scratch/my_workarea/sessions/uart_ctrl/vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101/chain_0/uart_tests --run_id 2 --stop_file /home/ruchikas/scratch/my_workarea/sessions/uart_ctrl/vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101/chain_0/stop_files/stop_run_2 --run_env_file /home/ruchikas/scratch/my_workarea/sessions/uart_ctrl/vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101/chain_0/uart_tests/tests_env --offsets 0,2559 --sch_offsets 0,0 --vsof /home/ruchikas/scratch/my_workarea/sessions/uart_ctrl/vm_uart_ctrl_regression.ruchikas.14_11_24_12_21_29_9101/vm_uart_ctrl_regression.vsof --seed random --sv_seed random --svv_name
```

While the session run is in progress, the *Session Status* field in the *Sessions* pane shows the status as *in_progress*.

You can click the *Refresh* button in the *Sessions* toolbar to refresh the status while the session is being run in the background. Else, by default, a session is refreshed and configured every 300 seconds.

After the session run is complete, the *Sessions* pane shows the results, as shown in [Figure 2-15](#) on page 88.

Figure 2-15 Session Results in the Sessions Pane



Session Status	Name	Total Runs	#Pa
(no filter)	(no filter)	(no filter)	(no filter)
completed	test1_13_01_16_11_56_36_0282_20	1	1
completed	top_anl_13_01_16_13_42_26_0192_23	1	1
completed	vm_uart_ctrl_regression.johnn.12_01_...	12	4
completed	cdn_uart_segal11_01_04_23_14_13_1...	18	16

For more details on understanding session results, see [Sessions Pane](#) on page 85.

Using the Run Status Information

Whenever a run is completed, the run status is computed and stored in the vManager database in a file named `vm_run_status` in the run directory. This file includes a single line

text with a single word as `passed` or `failed` depending on the run's status. If the file does not exist, it indicates that the run is incomplete.

Note: The file is also updated when the status changes due to editing of message severity.

Based on the value (`passed` or `failed`) stored in this file, you can perform certain automation tasks, such as deleting log files from passed runs and so on. These tasks can be performed out of vManager context.

For example, you can create a clean script, as shown below (sample):

```
#!/bin/sh
#Session dir cleanup
# Put any clean up commands to apply for the session dir
#Run dir cleanup
for a run in chain*/run_*; do
    # Put any cleanup commands to apply to ALL runs here
    if [ -r ${arun}/vm_run_status ]; then
        status=`cat ${arun}/vm_run_status`;
    if [ $status = "passed" ]; then
        # Put any clean up commands to apply only to PASSED runs here
    elif [ $status = "failed" ]; then
        # Put any clean up commands to apply only to FAILED runs here
    else
        echo "Unknown status";
    fi
    fi
done
```

This script assumes that it is called from the session directory. You can create such a script, and specify its execution in the *Configuration* dialog box. For more details, see [Incisive vManager Installation and Configuration Guide](#).

2.2.2 Importing Sessions

The *Import* button allows you to load a VSOF / VSOFX file. This is mostly for the users who already have session data generated and now want to analyze it.

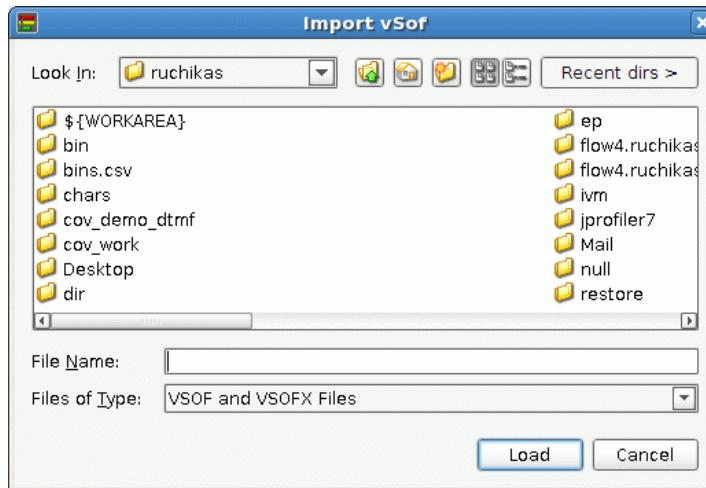
Note: The vManager server should have read and access permissions to this vsofx file you are trying to import.

To import a session:

1. Click the *Import* button on the *Global Operations* toolbar.

The *Import vSof* dialog box is displayed, as shown in [Figure 2-16](#) on page 90.

Figure 2-16 Import Session

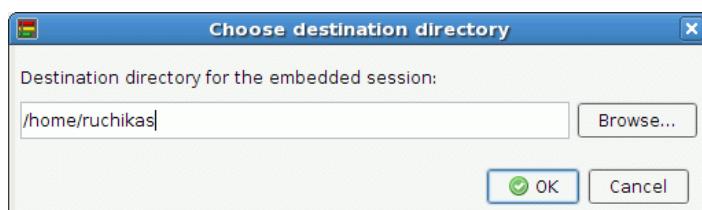


2. Navigate through the hierarchy and select the relevant VSOF or VSOFX file. For example, to open the vsop file named `cdn_uart.vsof` select `cdn_uart.vsof` and click *Load*.

 **Important**

In case you are trying to load a VSOFX file that was exported with the session directory, an additional dialog box appears at the time of importing, as shown in [Figure 2-17](#) on page 90.

Figure 2-17 Location for Uncompressing Session Directory



In this dialog box, specify the location where the session directory must be uncompressed and click *OK*.

The selected file is then loaded. After the vsop file is loaded, you can view the session details such as, number of runs, number of failed runs, passed runs, and so on.

Note: The import action fails if the total number of runs exceeds the limit of the number of runs supported by the license.

2.3 Exporting Sessions

Using vManager you can export sessions from the database to a VSOFX file. Exporting a session helps you at the time of archiving and also when you want to move sessions from one server to another.

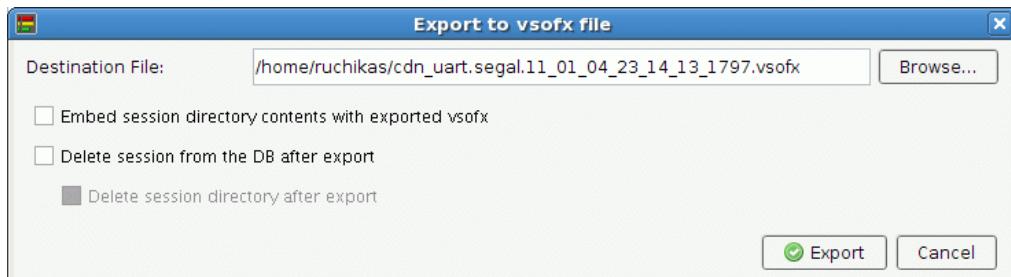
To export a session:

1. In the *Sessions* table, select the session(s) you want to export.
2. Click the *Export* button on the *Sessions* toolbar.

Note: Alternatively, you can select the *Export* option from the *Regression* menu.

The *Export to vsofx file* dialog box is displayed, as shown in [Figure 2-18](#) on page 91.

Figure 2-18 Export to vsofx file



3. In the *Destination File* field, specify the location where the session must be exported. You can also click the *Browse* button and navigate to the location where the session must be exported. By default, the exported session file is named as <name_of_session>.vssofx. You can change the file name, as required.

Note: In case multiple sessions are selected at the time of export, then for each session a vssofx file (<name_of_session>.vssofx) will be created in the location specified in the *Destination File* field.

4. Select the *Embed session directory contents with exported vssofx* check box to include the session directory in the exported VSOFX file. By default, the session directory is not included in the exported VSOFX file. (If you select this option while exporting, then at the time of import, you will be prompted for the location where the session directory must be uncompressed.)

5. In case you want to delete the session after the export action, select the *Delete session from the DB after export* check box.
 - a. When you select the *Delete session from the DB after export* check box, the *Delete session dir after export* check box becomes active. Select this check box if you want to delete the session as well as the session directory after the export action.
6. Click *Export*.

This will export the session to the specified location.

Note: The exported file (vSOFX) can be imported only in vManager. In addition, you can also export a session in the CLI mode using the export command.

2.4 Exporting Merged Coverage of Sessions

By default, vManager uses an internal automatic merge mechanism for any operation that requires merge (for example —collect runs, analyze session coverage, compaction, and so on). The merge results in such cases are not available to the user for analysis outside of vManager.

Using the *Export Merge* option of vManager, you can save the automatically merged coverage to a location so that it can be used later for analysis outside of vManager (for example, IMC).

To export merged coverage of a session:

1. In the *Sessions* table, select the session(s) for which you want to export the merged coverage data.
2. Click the *Export Merge* button on the *Sessions* toolbar.

Note: Alternatively, you can select the *Export Merge* option from the *Regression* menu or you can right-click the session and select the *Export Merge* option.

The *Merge Session* dialog box is displayed, as shown in [Figure 2-19](#) on page 92.

Figure 2-19 Merge Session



3. In the *Merge Directory* field, specify the location where the merged coverage of the session must be exported. You can also click the *Browse* button and navigate to the required location. A folder with the name as the session name is created under the location you specify in the *Merge Directory* field.

Note: If multiple sessions are selected, then for each session a folder (<name_of_session>) will be created in the location specified in the *Destination File* field.

4. Click *OK*.

This will export the merged coverage of the selected session to the specified location.

Note: The exported merged coverage can then be used for analysis even outside of vManager (for example, IMC). You can also export merged coverage of a session in the CLI mode using the export_merge command.

2.5 Collecting Runs

vManager allows you to create a new session by collecting runs result from the disk. This is a typical flow for users who run their regressions using external regression runner.

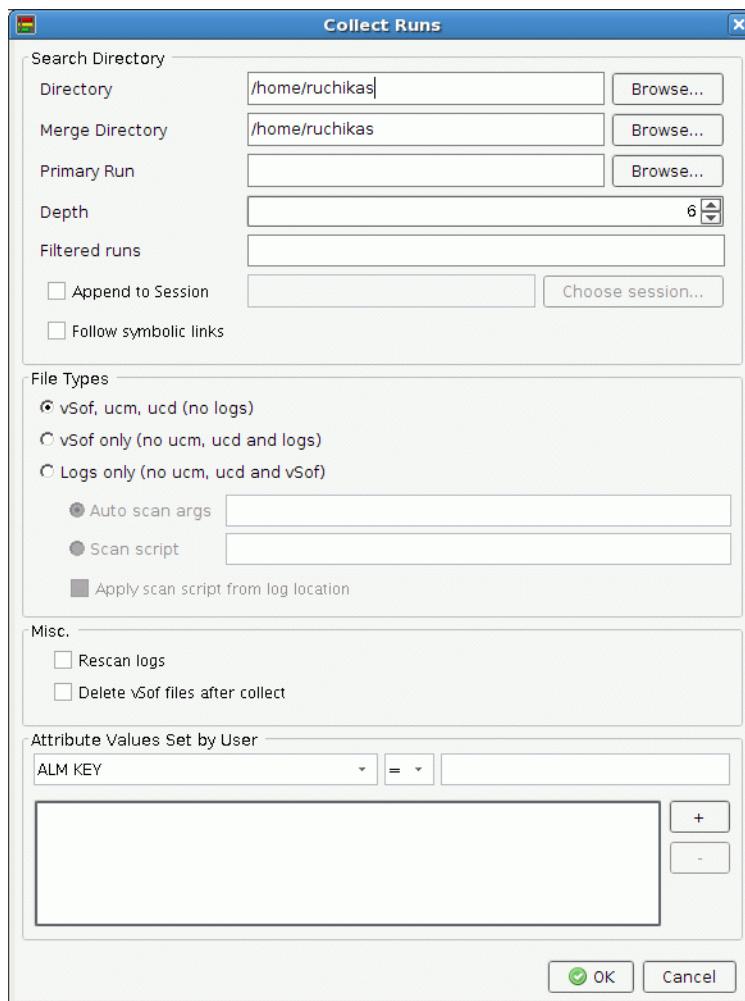
To collect runs and create a new session:

1. Click the *Collect Runs* button on the *Global Operations* toolbar.

Note: Alternatively, you can select the *Collect Runs* option from the *Regression* menu.

The *Collect Runs* dialog box is displayed, as shown in [Figure 2-20](#) on page 94.

Figure 2-20 Collect Runs



2. In the *Directory* field, specify the directory where runs can be found for collection process.
3. In the *Merge Directory* field, specify the location where the merged ucd directory will be created.
4. In the *Primary Run* field, specify the primary run for the merge operation. Only ICC runs can be specified as primary runs. An error is reported if a JG/UXE run is specified as the primary run. If not specified, then the model with the most recent timestamp is taken as the primary run.
Note: You can also set this attribute (Primary Run) later, using the [Edit each](#) dialog box. For more details on Primary Run attribute, see [Primary Run](#) on page 377.
5. In the *Depth* field, specify the limit below which the run directories will not be collected.

6. In the *Filtered runs* field, specify the regular expression pattern that defines the directories or the file names that should be collected from the top directory. For example, if you want to collect runs from /home/ben/top/ and there are three directories x/, xx/, and y/ in it and you want to collect only from x/ and xx/, then specify /home/ben/top/ as the top directory and x* in the *Filtered runs* field.
7. You can choose to append the collected session to other collected sessions. If there are more collected sessions in the database, the *Append to Session* check box will be enabled and you can choose the already collected session to which the new session must be appended.
8. You can select the *Follow symbolic links* check box to cause the collect runs operation to follow the symbolic links.
9. In the *File Types* section of the dialog box, specify the files to be generated during the collection process. You can select any of the following options:
 - vSof, ucm, ucd (no logs) — To enable collection of vsosf files and coverage run directories. With this option, collection of log files is ignored.
 - vSof only (no ucm, ucd, and logs) — To enable collection of only a single run vsosf file. With this option, run directories that include no vsosf (even directories with coverage databases) is ignored.
 - Logs only (no ucm, ucd, and vSof) — To enable collection of only the log files. With this option, collection of vsosf files and coverage run directories is ignored. If you select this option, you must specify how to scan the logs. The logs can be scanned in any of the following ways:
 - Auto scan args
 - Scan script
10. Select *Rescan logs* check box to rescan logs instead of considering scan results from single run vsosf files.
11. Select the *Delete vSof files after collect* check box to delete the single run vsosf files after the collect operation.
12. In the *Attributes Values Set by User* section of the dialog box, specify the attribute and value that will override the value of the specified attribute in the collected runs at the time of creating the new session. Select the attribute name from the drop-down list.
13. Select the operator from the drop-down list. You can select any of the following operators:

- : means to resolve the specified value with the collected value.
- = means force setting a value. In this case, the value specified by the user is set.

14. Click the + button to add the attribute-value pair to override the values. To remove an attribute-value pair, select the attribute=value pair and click the – button.

15. Click *OK*.

Note: The collect runs action fails if the total number of runs exceeds the limit of the number of runs supported by the license. You can proceed with the collect runs action by selecting the *Append to Session* check box in the *Collect Runs* dialog box.

Important

vManager supports *Collect Runs* action on coverage coming from Palladium as well as from IFV or JasperGold. After you collect runs, a new session will be created and you can analyze metrics and also perform other analysis tasks on the collected session. For details on analyzing coverage coming from Palladium, see [Analyzing Coverage Coming from Palladium](#) on page 235. For details on analyzing coverage coming from JasperGold, see [Analyzing Coverage Coming from JasperGold](#) on page 237.

2.6 Refreshing Sessions

When you launch a VSIF file, it runs the session. The time taken to complete the session run may vary depending on the machine configuration and the number of runs within the session.

After the session is complete, the session completion status is reflected in the *Sessions* pane.

However, you also have an option of viewing the session status intermediately while the session is being run in the background. You can do this by clicking the *Refresh* button in the *Sessions* toolbar.

The *Refresh* button refreshes the status of all the running sessions and shows the updated status of the sessions till that time.

2.6.1 Configuring Session Refresh Options

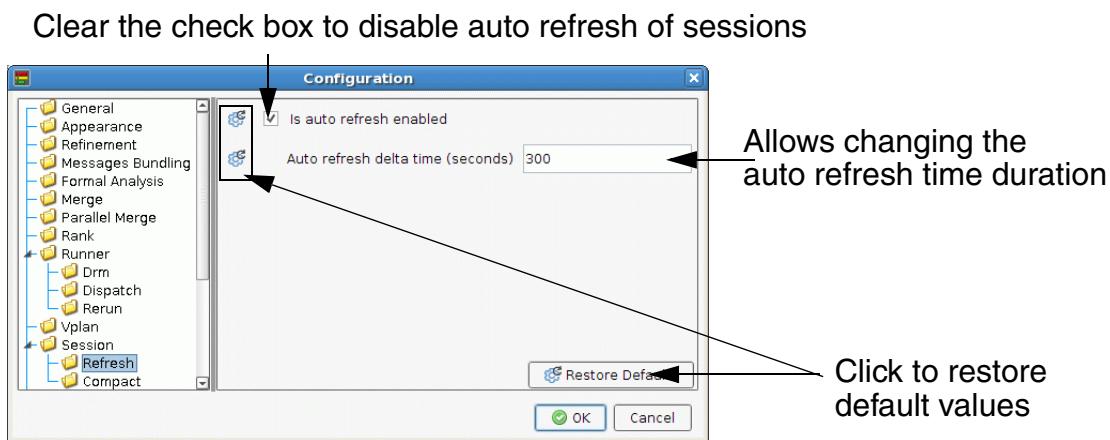
By default, the session status is refreshed every 300 seconds. You can modify this time duration using the *Configuration* option in the *View* menu.

To configure session refresh options:

1. Select *Configuration* from the *View* menu.
2. Click *Refresh* under the *Session* folder.

The session refresh options are displayed, as shown in [Figure 2-21](#) on page 97.

Figure 2-21 Configuration—Refresh Options



3. In this dialog box, you can:

- ❑ Enable or disable the auto refresh option. By default, auto refresh is enabled and the sessions refresh every 300 seconds.
- ❑ Modify the time duration after which session should refresh automatically.

For example, to change the time duration for auto refresh to 3 seconds, specify 3 in the text box and click *OK*.

The value specified in the *Configuration* dialog box applies to all of the loaded sessions.

2.7 Opening Session Directory in Terminal

Using vManager, you can open a terminal window in which the selected session will be shown as the current working directory. You can then navigate through the session directory and also browse its contents.

To open a session directory in a terminal window, do any of the following:

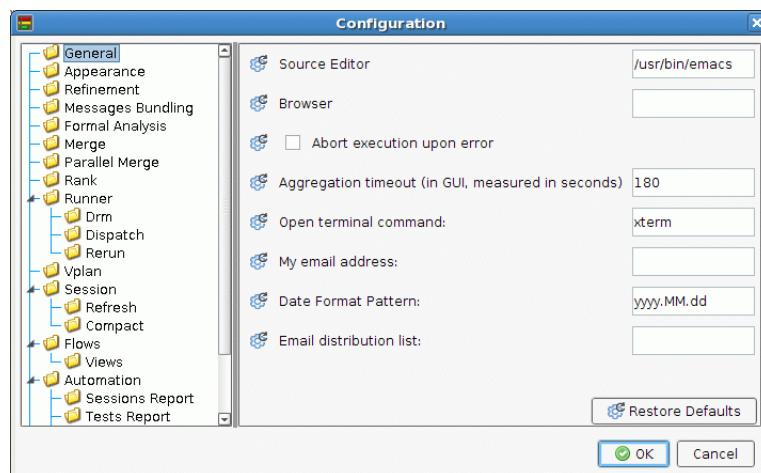
- Select the session and select *Open session directory* from the *Regression* menu.
- Select the session and select *Open dir* button on the *Sessions* toolbar.

- Select the session, right-click and select *Open session directory*.

By default, `xterm` is opened as the terminal. However, you can change using the *Configuration* option in the *View* menu. To change terminal in which session directory is opened, select *Configuration* from the *View* menu.

The session *General* options are displayed, as shown in [Figure 2-22](#) on page 98.

Figure 2-22 Configuration—General Options



By default, *Open terminal command* is specified as `xterm`. In case you want to change the terminal, specify it in the text box and click *OK*.

For details on other options available in this page, see [Configuring General Options](#) on page 55.

Note: The value you specify in the *Configuration* dialog box applies to all of the loaded sessions.

2.8 Viewing Session Information

If a session fails, then you might want to investigate the reason for session failure and fix it accordingly. To investigate the reason, you need to view the session failures and associated log files. A session fails if any failures of severity *error* or *critical* are issued during the performance of a session task and those failures are not attributed to a particular run or group of runs.

Session failures typically occur when:

- The pre-session script or post-session script fails.

- The master DPL process fails and is not restarted.

Note: If the session is not marked as *failed*, but none of the runs has completed normally, it is possible that a DRM-related failure occurred that was not reported to vManager. In such a case, you must contact Cadence Support for assistance.

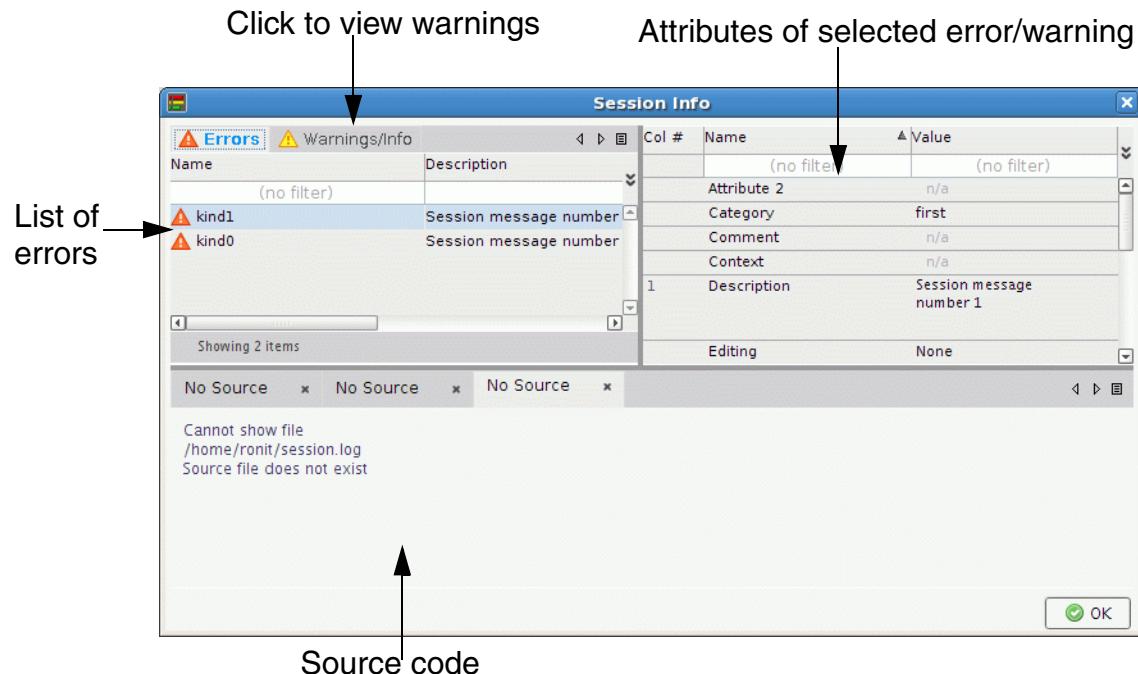
To view session failures and associated log files:

1. Select the session in the *Sessions* pane.
2. Click *Session Info* button in the *Sessions* toolbar.

Note: Alternatively, you can select *Session Info* from the *Regression* menu.

The *Session Info* dialog box is displayed, as shown in [Figure 2-23](#) on page 99.

Figure 2-23 Session Info



Note: The *Session Info* dialog box does not show anything in the *Errors & Warnings* and *Attributes* pane, if you select a completed session.

The *Session Info* dialog box shows session information such as errors, warnings, and associated logs.

2.9 Stopping Sessions

To stop a session, you can do any of the following:

- Select the session that you want to stop and click the *Stop* button in the *Sessions* toolbar.
- Right-click the session that you want to stop and select *Stop*.

A confirmation dialog appears and once you confirm the stopping, the session is stopped.

You can stop multiple sessions at a time by selecting them (to select multiple sessions, use Shift+Select or Ctrl+Select).

2.10 Stopping Session Based on Conditions

vManager allows you to automatically stop the session progress based on certain conditions (as desired). For example, some of the conditions could be:

- The number of failed runs reaches a specified number.
- The number of failed runs with specified message reaches a specified number.
- The number of failed runs in specified runs reaches a specified number.

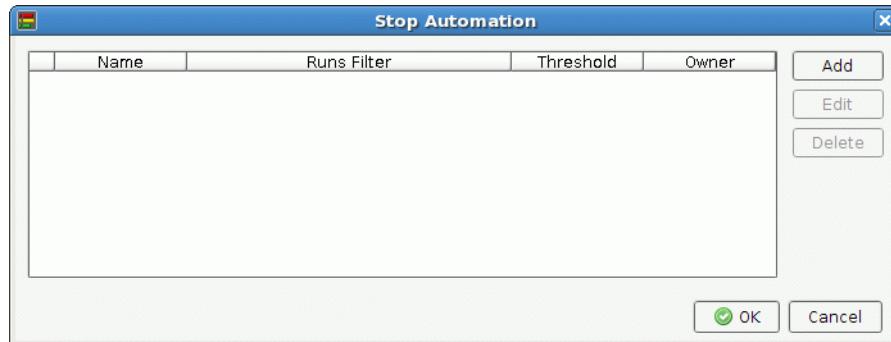
This feature helps in saving the resources, such as CPU and time because it automatically stops the session based on the specified conditions.

To enable stopping of sessions based on conditions, you need to define rules in the *Stop Automation* dialog box. For this, perform the following steps:

1. Click the *Stop Automation* button in the Sessions toolbar, or select *Stop Automation* from Regression menu.

The *Stop Automation* dialog box appears, as shown in [Figure 2-24](#) on page 101.

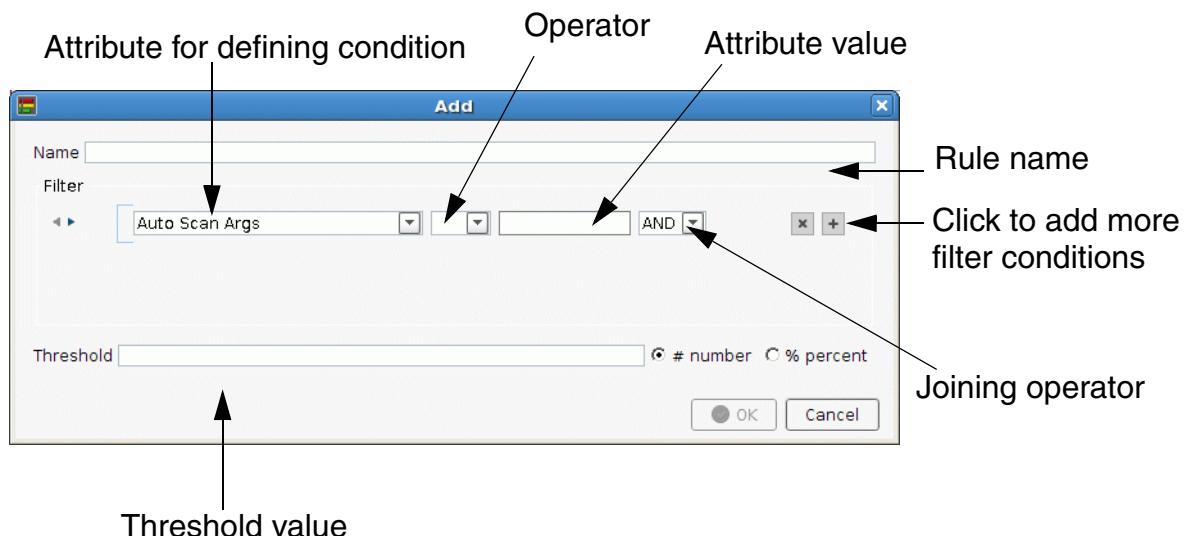
Figure 2-24 Stop Automation



2. Click *Add* to add a new rule.

The *Add* dialog box appears, as shown in [Figure 2-25](#) on page 101.

Figure 2-25 Add



Note: The semantics of the rule is -- *Stop the session if the runs matching the [Filter_Criteria] are greater than the [Threshold] [Type]*.

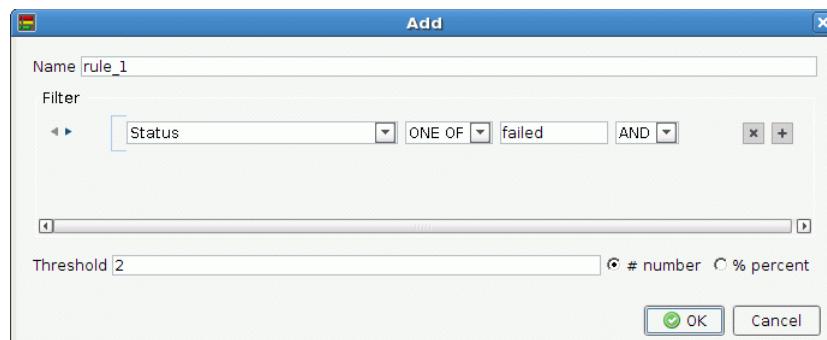
3. Specify the name of the rule in *Name* field. For example, specify the name as `rule_1`.
4. Define the filter criteria (condition) for the rule in the *Filter* group box. For example, consider creating a rule such that status of the run is failed.
 - a. First, specify the attribute for the filter criteria. In this case, select *Status* from the drop-down list.

- b. Select the relevant operator in the next drop-down. In this case, select *ONE OF*.
- c. In the next drop-down, specify the attribute value. In this case, select *failed* and click *Filter*.

Note: In case you want to add more conditions to the filter criteria, specify the appropriate joining operator and click the + sign.
5. After specifying the filter criteria, specify the threshold for the rule. The threshold is the value that affects the way in which the filter criteria is evaluated. For example, if you want to apply the rule such that the filter criterion is met two times, then specify 2 in the *Threshold* field.
6. By default, # *number* radio button (number of runs in count) is already selected. In case you want the threshold to be in percentage, then select the % *percent* (number of runs in percentage) radio button.

[Figure 2-26](#) on page 102 shows the *Add* dialog box with specified values.

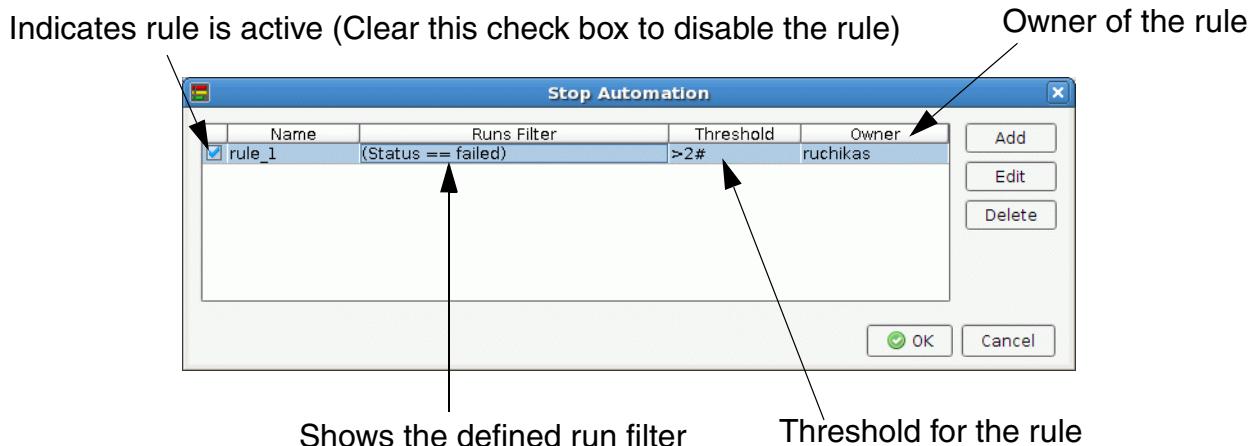
Figure 2-26 Add Rule



7. Click *OK*.

You then return to the *Stop Automation* dialog box with the rule created, as shown in [Figure 2-27](#) on page 103.

Figure 2-27 Add Rule



The check box in front of the rule indicates if the rule is enabled or disabled. By default, when you define a rule, it gets enabled. You can disable the rule by clearing the check box. In this dialog box, if required, you can add more rules, edit existing rules or delete rules using the relevant buttons.

8. Click *OK*.

Similarly, you can create more rules, as required.

Things to Remember When Defining Rules

- The rules are shared across users; however, the condition activation (configuration settings) is not shared between users.
- The *Stop Session Automation* is also applicable in the case of reruns (only if the run mode is set as *Start new session*).
- If multiple rules are enabled, then session is stopped even if one of the activated rules is satisfied.
- If a session is stopped due to a stop session condition, then an error message is added to the session level and following attributes are populated with appropriate values:
 - Name: *Stop Session Automation*
 - Description: <criteria_that_failed>
 - Severity: *Error*

You can check these values in the *Session Info* dialog box (Click the *Session Info* button in the *Sessions* toolbar to open this dialog box).

Note: If the user has requested a session done report via the email (in the *Configuration* dialog box), then the above details are also included in the email.

- You can also define stop session automation in CLI mode, using the [automation](#) command.

2.11 Suspend / Resume Sessions

vManager supports suspend and resume actions only for the sessions launched with the new runner.

To suspend a session, you can do any of the following:

- Select an already running session and click the *Suspend* button in the *Sessions* toolbar.
- Right-click the already running session and select *Suspend*.

Note: The *Suspend* option shows enabled only if you select an incomplete session launched with the new runner.

When you suspend a session, the *Status* of the session changes to suspended.

After a session is suspended, no more new jobs are submitted to the DRM until the session is resumed. All active jobs (submitted to the DRM) are suspended using the appropriate DRM commands (lsf: bstop, nc: nc suspend, sge/uge:qmod –sj). For local DRM, all running jobs continue to run until they are completed. To suspend a session in CLI mode, use the [suspend](#) command.

To resume a suspended session, you can do any of the following:

- Select a suspended session and click the *Resume* button in the *Sessions* toolbar.
- Right-click a suspended session and select *Resume*.

Note: The *Resume* option shows enabled only if you select a suspended session.

When a *Resume* action is applied on a suspended session, all the suspended jobs are resumed using the appropriate DRM command (lsf bresume, nc: nc resume, sge/uge: qmod –usj) and the jobs manager continues to submit new jobs as needed. The session status is updated to the status the session had before suspension.

To resume a session in CLI mode, use the [resume](#) command.

Note: Only the session owner can suspend or resume the session.

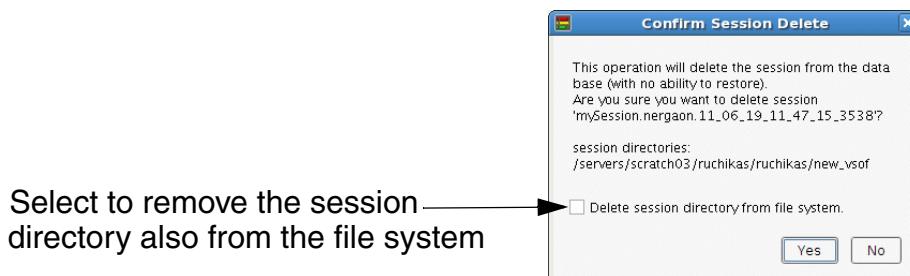
2.12 Deleting Sessions

To delete a session, you can do any of the following:

- Select the session that you want to delete and click the *Delete* button in the *Sessions* toolbar.
- Right-click the session that you want to delete and select *Delete*.

A confirmation dialog appears, as shown in [Figure 2-28](#) on page 105.

Figure 2-28 Deleting Session



Once you confirm the deletion, the session is deleted.

Note: You can delete multiple sessions at a time by selecting them (to select multiple sessions, use Shift+Select or Ctrl+Select). In addition, you can delete only the sessions that belong to you (that is, where the session owner is <your_name>).

 *Important*

You should be the owner of the session to delete it. You can change the owner of a session using the *Edit each* or *Edit all at once* options. For more details, see [Editing Session Details](#) on page 108.

2.13 Relocating Session

Relocating a session enables you to move a session from one location to some other location possibly on a different file system without affecting the analysis capabilities.

The relocation action moves the session directory, and also updates the references to the directory in vManager database.

Note: You can also move the session directory manually and then perform the relocation action to just update the references in the vManager database.

Another user-flow in which relocation can be useful is:

- Export the session from the vManager database to a `vsوفx` file.
- Send both (the session directory and the exported `vsوفx` file) to the person who needs it. The receiver might have a different file system.
- The receiver can then import the `vsوفx` file and perform the relocation action to update the references.

To relocate a session:

1. Select the session in the *Sessions* pane.
2. Click the *Relocate* button on the *Sessions* toolbar.

The *Session Relocation* dialog box is displayed, as shown in [Figure 2-29](#) on page 106.

Figure 2-29 Session Relocate



3. In the *New session top directory* field specify the new session's top directory. By default, it is created in the current working directory.
4. Select the *Delete old session directory (if exists)* check box to delete the old relocated session directory, if it exists.
5. Click OK.

The session is relocated to the specified location.

You can also relocate a session from command-line mode of vManager. For more details, see [relocate](#) on page 439.

2.14 Handling Inaccessible Sessions

vManager does not allow loading, importing, collecting, or launching of sessions if the runs in the database has reached the limit of runs specified for the license.

By default, an error is reported in these cases. However, in the case of launching a session, vManager allows you to enable the launch of session even after the run count limit has reached. For this, you need to set the *Enable Launch Over Runs Limit* configuration option in the *Configuration* dialog box.

After the *Enable Launch Over Runs Limit* configuration option is set, vManager allows launching of sessions even after the maximum runs limit has reached; however, the session is marked as inaccessible.

An inaccessible session does not contain any runs data.

Note: Launching separate pages for runs and metrics analysis is not permitted for sessions that are marked inaccessible. Only stop and delete actions are allowed on an inaccessible session. For any other action (both in GUI as well as batch mode), an error is reported and the action or command is ignored for the inaccessible session.

In case you try to make multiple selections in the *Sessions* pane (with inaccessible sessions and other sessions) and try to launch separate runs or metrics analysis pages, then the inaccessible session is ignored and separate analysis pages are launched for other selected sessions.

An inaccessible session gets enabled (becomes a regular session) if you delete a few sessions and the run count becomes less than the limit of runs specified for the license. After an inaccessible session gets enabled, vManager starts to read the session run data.

In case there are multiple inaccessible sessions, then the session which gets enabled first is based on the value of attribute `DB Insertion Priority`. The possible values of this attribute are:

- `LOW` (default) — Indicates lowest precedence
- `MID` — Indicates medium precedence
- `HIGH` — Indicates highest precedence

By default, the value of this attribute is `LOW`, which indicates that the session must be enabled based on FIFO policy (that is first-in first-out) considering the start time of the session. A session with the `DB Insertion Priority` attribute set as `HIGH` gets precedence over the sessions with value as `MID` or `LOW`.

You can set the value of this attribute in the `vsif` file or by using the *Edit Attributes* feature of GUI. For more details on editing the attributes values, see [Editing Session Details](#) on page 108.

2.15 Editing Session Details

Using vManager, you can add or modify session details, such as comments, description, editable user-defined attributes, max runs in parallel, database insertion priority, or owner of the session.

This helps you in capturing your observations and remarks so that they are available to you and others at a later date.

The changes made to any of the attributes immediately take into effect.

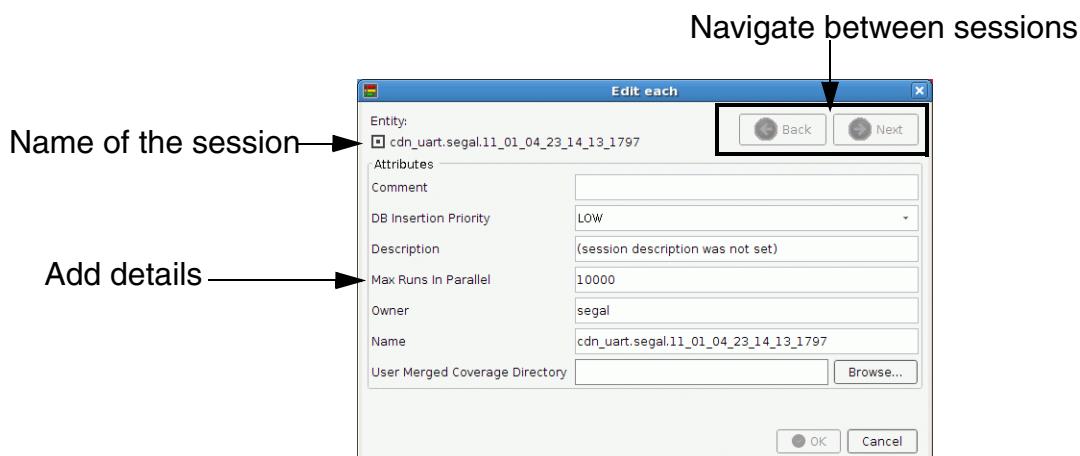
The *Attributes* toolbar allows you to add comments and description to the sessions. It also allows you to view the attribute change history. The *Attributes* toolbar has the following options:

- Edit each—To add comments or description to the selected session one by one
- Edit all at once —To add comments or description to all the selected sessions together
- History—To view attribute change history

2.15.1 Edit each

The *Edit each* option allows you to add comments to the selected sessions one by one. Figure 2-30 on page 108 shows the *Edit each* dialog box.

Figure 2-30 Edit Each



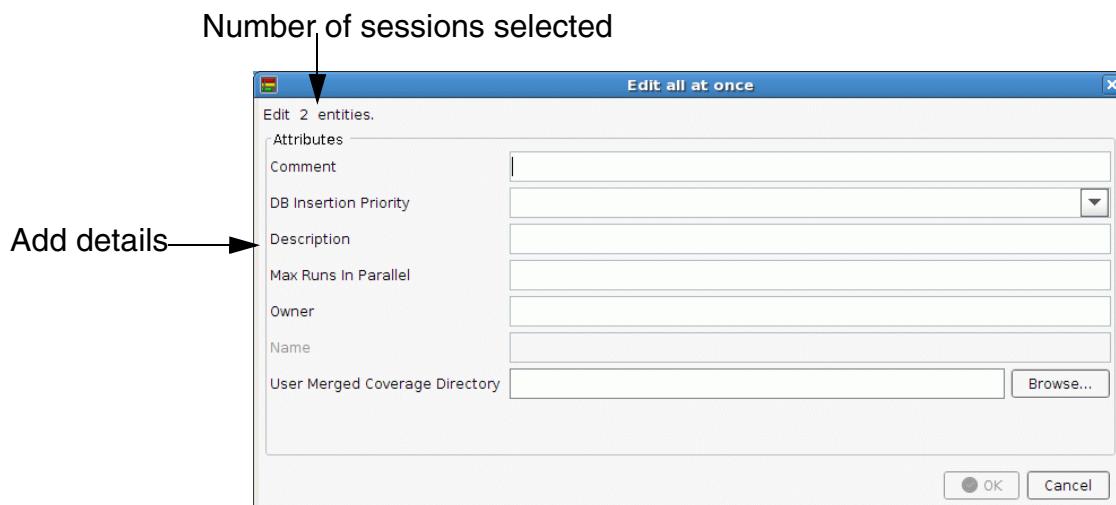
Using this dialog box, you can add comments, owner, or description for each session individually. You can add details and then navigate to the next or previous session using the *Back* or *Next* buttons.

The information you add in this dialog box is visible in the *Attributes* pane.

2.15.2 Edit all at once

The *Edit all at once* option allows you to add comments to all of the selected sessions at once. [Figure 2-31](#) on page 109 shows the *Edit all at once* dialog box.

Figure 2-31 Edit All At Once

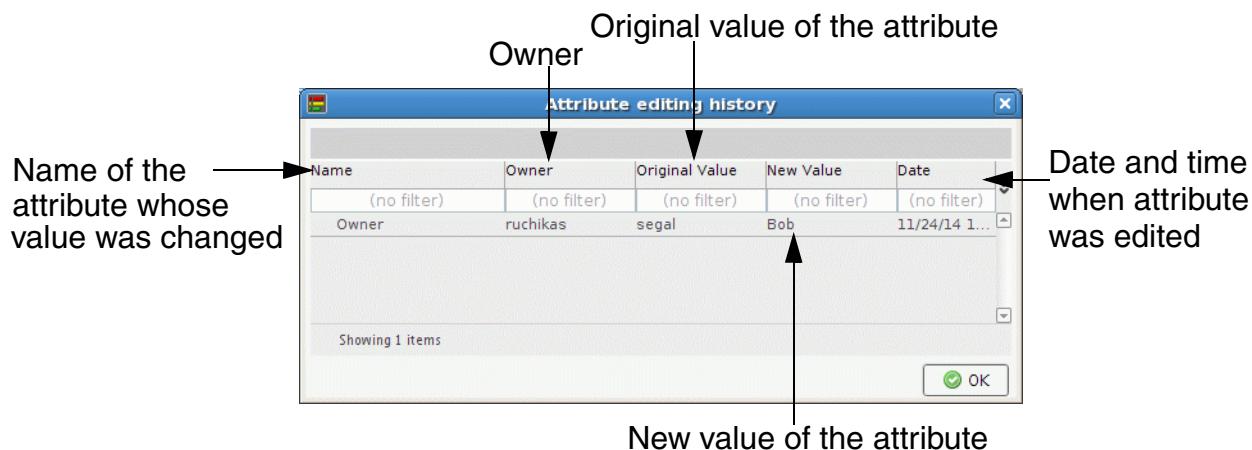


The information added using this dialog box, applies to all of the selected sessions.

2.15.3 History

The *History* option allows you to view attribute change history of the selected session. [Figure 2-32](#) on page 110 shows the *Attribute editing history* dialog box.

Figure 2-32 Attribute Editing History



This dialog box maintains the history of all the changes made to different attributes of the selected session.

2.16 Associating Merged Coverage with Sessions

By default, vManager uses an internal automatic merge mechanism for any operation that requires merge (for example — collect runs, analyze session coverage, compaction, and so on).

Note: vManager allows you to save the automatically merged coverage to a location so that it can be used later for analysis outside of vManager (for example, IMC). For more details, see [Exporting Merged Coverage of Sessions](#) on page 92.

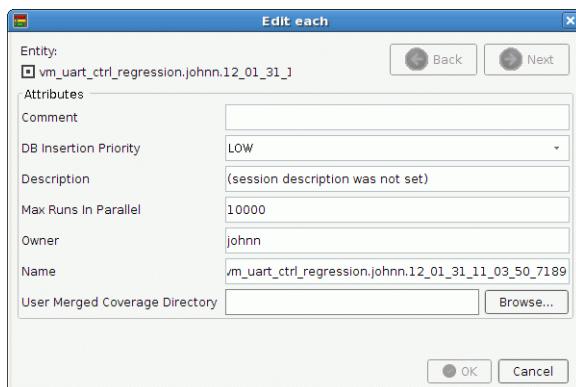
If required, instead of automatic merge mechanism, you can attach previously merged results (from same or a different tool) to a vManager session and use it for analysis.

To attach previously merged data and to use it, perform the following steps:

1. Select the session with which you want to attach the previously merged data.
2. Right-click the session and select *Edit each*.

The *Edit each* dialog box appears, as shown in [Figure 2-33](#) on page 111.

Figure 2-33 Edit each



3. In the *Edit each* dialog box, specify the path of already merged coverage data in the *User Merged Coverage Dir* field.

Note: You can also set the value of this attribute using the vManager `edit` command, as shown below:

```
vmanager> edit <session_name> -attribute  
user_merged_coverage_dir=<location_of_merged_results>
```

After the value of this attribute/field is set, for any operation that requires merge, vManager uses this already merged data instead of merging automatically. The only exception to this is in the case of rank and correlation. The rank and correlation operations always perform the automatic merge, regardless of the value specified in the *User Merged Coverage Dir* field.

Important

The value specified in the *User Merged Coverage Dir* field/attribute is not considered until the *Consider User Merged Coverage* check box is selected in the *Configuration* dialog box. The *Consider User Merged Coverage* check box is available under the *Merge* folder in the *Configuration* dialog box. For more details, see [Configuring Merge Options](#) on page 67.

2.17 Creating Compact Sessions

Large sessions with many coverage files take a lot of disk space and also impacts performance. vManager provides you with an ability to compact the sessions based on your requirements, and if required save the database space by deleting the original session after compaction.

When you compact a session:

- vManager creates a new session in the database and each line in the runs table is represented as a new single run. This new run is called a compact run.
- vManager also creates a new run directory for each compact run.
- A single .ucd file is created for each compact run by merging all the .ucd files of the original runs within the group.
- vManager also creates a standard cov_work directory for each run and saves the .ucd file in test directory. vManager also creates a model_dir directory in the new session directory which includes all the unique .ucm files, and each run contains a symbolic link to the appropriate ucm file. The cov_work directory can be loaded to IMC for analysis.
- All the attributes that belong to original run gets added to each new run, with their aggregated value.
- The computed attributes of each run are recalculated based on the new data.

Note: The compaction action fails if the number of runs exceeds the limit of the number of runs supported by the license.

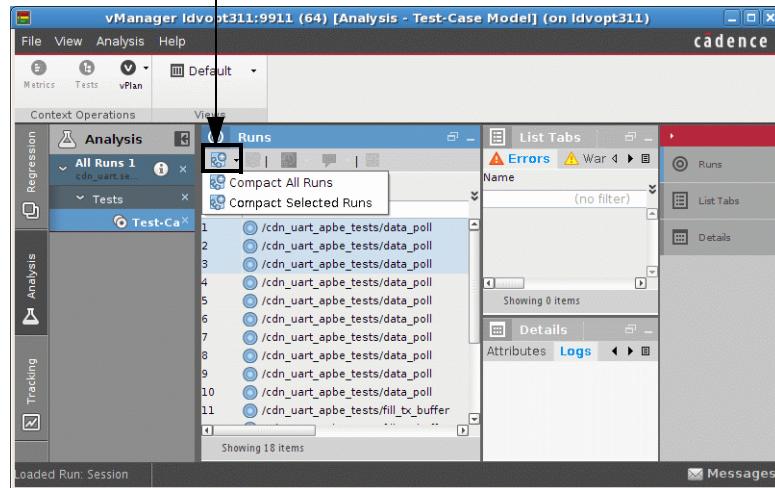
To create a compact session, perform the following steps:

1. Import vsof(s) or launch the vsif(s) of the sessions you want to compact.
2. Select the sessions and launch the *Runs Analysis* page. For example, right-click the sessions and select *Analyze All Runs* from the popup menu.
3. If required, you can apply filters and group the runs and then perform the compaction action. The grouping or filtering actions are not mandatory for compaction.

[Figure 2-34](#) on page 113 shows the Runs Analysis page.

Figure 2-34 Runs Analysis

Click to create new compacted context



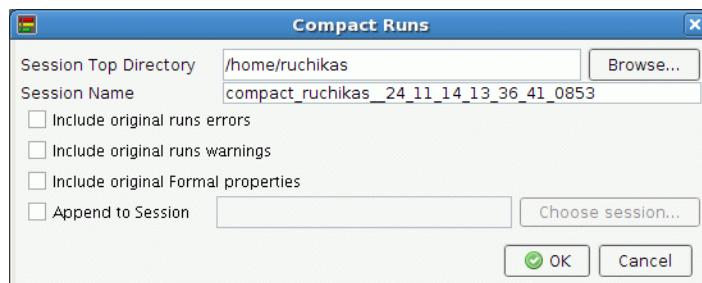
4. Click the drop-down button highlighted in above figure to create a new compacted context. It has following options:

- Compact All Runs* — Used to compact all the runs listed in *Runs* table.
- Compact Selected Runs* — Used to compact the selected runs in the *Runs* table.

For example, select a few runs in the runs table and then select *Compact Selected Runs* from the drop-down list.

The *Compact Runs* dialog box is displayed, as shown in [Figure 2-35](#) on page 113.

Figure 2-35 Compact Runs



5. In the *Session Top Directory* field, specify the location where the compacted session must be created. By default, it shows the location from which vManager was invoked.

6. Specify the name of the compacted session in the *Session Name* field. By default, it shows the name as `compact_username_dd-MM-yyyy_hh~mm~ss`. For example, specify the name as `compact_mike`.
7. By default, errors from the original runs are not copied to the compacted session. To copy errors from the original session to the compacted session, select the *Include original runs errors* check box.

Note: Even if the errors are not copied to the new compacted runs, the errors will be considered in status calculation.
8. By default, warnings from the original runs are not copied to the compacted session. To copy warnings from the original session to the compacted session, select the *Include original runs warnings* check box.
9. By default, formal properties from the original runs are not copied to the compacted session. To copy formal properties from the original session to the compacted session, select the *Include original Formal properties* check box.
10. In case you want to append the selected runs to an already compacted session, select the *Append to Session* check box. When you select this check box, the *Session Top Directory* field and the *Session Name* field become disabled. You can then click the *Choose session* button and select the compacted session to which the selected runs must be appended. When you click the *Choose session* button, the *Select session* dialog box appears, as shown in [Figure 2-36](#) on page 114.

Figure 2-36 Select session



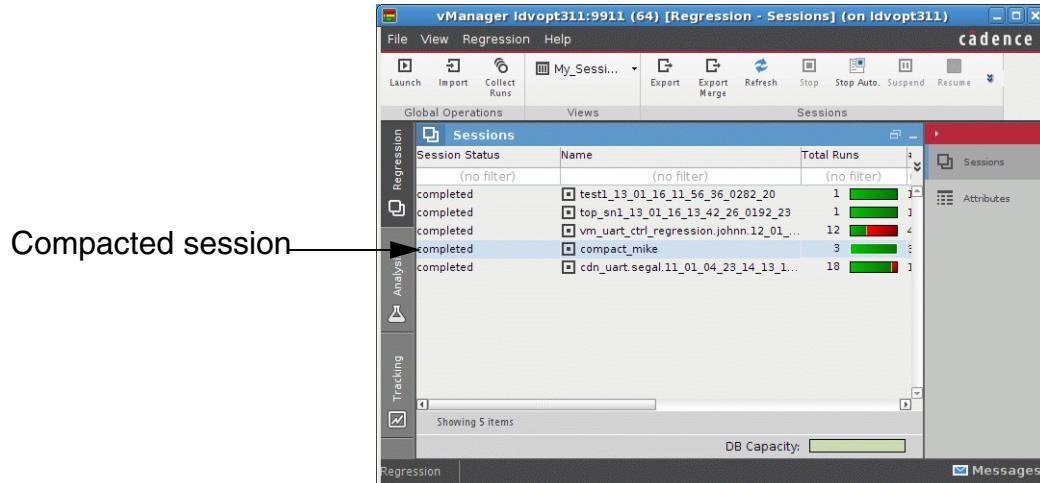
The *Select session* dialog box lists the already compacted sessions. You can select the compacted session to which the selected runs must be appended and click *OK*.

Note: The Append to session feature enables appending the selected runs only to an already compacted session.

11. Click *OK* in the Compact Runs dialog box to compact the session.

The compacted session then appears in the list of sessions in the Regression center, as shown in [Figure 2-37](#) on page 115.

Figure 2-37 Compacted Session

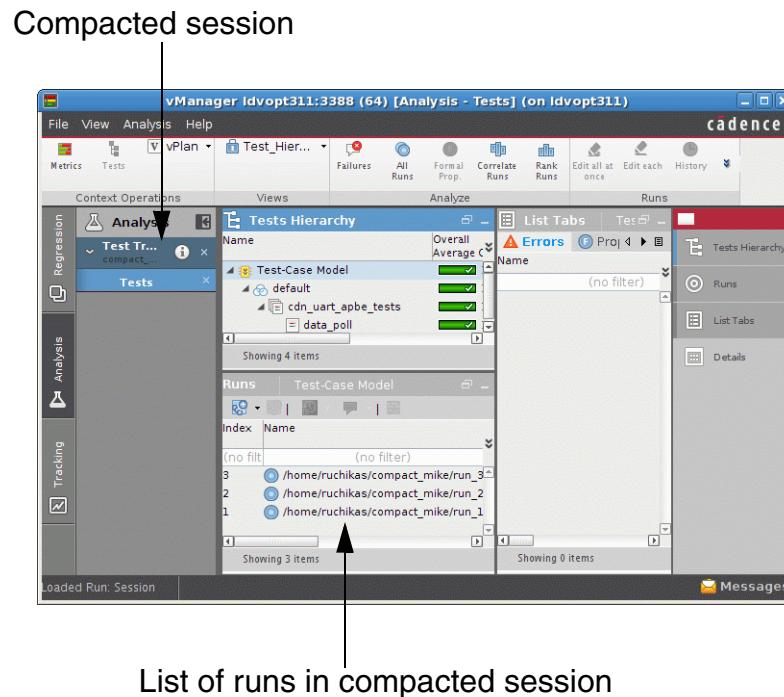


You can then open the compacted session in the Analysis center by selecting appropriate Analyze options.

For example, select the compacted session, right-click, and then select *Analyze Tests*.

[Figure 2-38](#) on page 116 shows the compacted session opened in *Analysis Center*.

Figure 2-38 Compacted Session Opened in Analysis Center



Note: By default, the original test tree is preserved. In case you do not want to preserve the original test tree upon compaction, clear the *Preserve original test tree* check box in the *Configuration* dialog box. It is available in the *Compact* folder under *Session* category. For more details, see [Configuring Session Compact Options](#) on page 64.

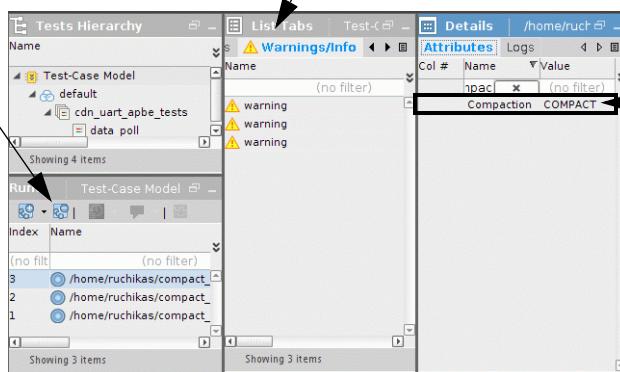
The Runs pane show the runs that were selected for compaction, as shown in [Figure 2-39](#) on page 117.

Figure 2-39 Compacted Runs

Click to create uncompacted context from compacted run

Error and warning are listed if the corresponding check boxes are selected at the time of compaction

Indicates if the run is a result of compaction



The error and warning are listed only if the corresponding check boxes (*Include original runs errors* and *Include original runs warnings*) are selected at the time of compaction.

The attributes values are recalculated and can be viewed in the *Attributes* pane. The *Compaction* attribute indicates if the run is a result of compaction or not. Valid values for this attribute are:

- COMPACT indicates that the run is the result of compaction.
- FLAT indicates that the run was created during simulation.

The other attributes such as #Failed, #Passed, #Runs and so on are also recalculated based on the grouping.

Note: You can modify the attributes of the compacted runs. Any modifications to the attributes of the compacted runs will not have any effect on the original runs or failures.

For details on compacting sessions in command line interface, see [compact](#) on page 456.

2.18 Rollup Attributes to Session Level

vManager allows you to show information from runs, metrics, or vPlan at the session level. For this, perform the following tasks:

1. [Write a Calculation Script File](#)
2. [Define User-defined Attributes in the CSV File](#)
3. [Import the CSV File](#)

4. Recalculate UDA

2.18.1 Write a Calculation Script File

In the calculation script file, define the attribute whose value has to be rolled up to session level. To show values rolled up from runs, metrics, or vPlan, the calculation script function `compute_attribute_value` is extended to support following functions:

- calculate
- calculateMetricGrade
- calculateVplanGrade

2.18.1.1 calculate

The syntax structure for using the `calculate` function is:

```
setAttributeValue [calculate entityType attributeName aggregationMethod]
```

where:

- `entityType` is the type of entity whose attribute will be used for calculation. It is specified in the string format. Currently, only `Run` entity is supported.
- `attributeName` is the name of the attribute that is used for the calculation. It is specified in the string format.
- `aggregationMethod` is the method used for calculation. Possible methods that can be used are:
 - `UniqueValueCount` — Calculates the number of unique values.
 - `Sum` — Calculates the sum of all the values.
 - `Average` — Calculates the average of all the values.

Examples:

The following calculates on *Runs*: how many unique values are per comment.

```
proc compute_attribute_value {attributes type} {  
    upvar attributes attrs;  
    setAttributeValue [calculate run comment UniqueValueCount];  
}
```

The following script calculates on *Runs*: how many unique values are per host.

```
Proc compute_attribute_value(attributes) {
    setAttributeValue [calculate run host UniqueValueCount];
}
```

The following script calculates on *Runs*: the sum of CPU time.

```
Proc compute_attribute_value(attributes) {
    setAttributeValue[calculate run cpuTime Sum];
}
```

2.18.1.2 calculateMetricGrade

Returns the grade value from the metrics. The value returned has to be set by the `setAttributevalue` function. For this calculation, vManager will merge the coverage data.

The syntax structure for using the `calculateMetricGrade` function is:

```
setAttributeValue [calculateMetricGrade verificationScope hierarchy cctype metricAttribute]
```

where:

- `verificationScope` is the verification scope of the entity to be used for calculation. It is specified in string format. If not specified, `default` will be used.
- `hierarchy` is the hierarchy tree of the entity to be used for calculation. It is specified in string format and can include wildcard characters.
- `cctype` is the type of entity. It can either be `Instance` or `Type`. It is specified in string format.
- `metricAttribute` is the name of the attribute of the entity that is used for the calculation. It is specified in string format.

Example:

The following script shows `BLOCK_AVERAGE_GRADE` of instance `uart/top/u1` in verification scope `VS`.

```
proc compute_attribute_value(attributes) {
    setAttributeValue [calculateMetricGrade VS uart/top/u1 Instance
        BLOCK_AVERAGE_GRADE];
}
```

The following script shows overall average grade from the top entity:

```
proc compute_attribute_value {attributes type} {
    upvar attributes attrs;
    setAttributeValue [calculateMetricGrade default "" OVERALL_AVERAGE_GRADE];
}
```

2.18.1.3 calculateVplanGrade

Returns the grade value from the vPlan specified in vPlanFilePath. The value returned has to be set by the setAttributevalue function. For this calculation, vManager will merge the coverage data and load the vplan, if needed. The syntax structure for using the calculateVplanGrade function is:

```
setAttributeValue [calculateVplanGrade vPlanFilePath hierarchy metricAttribute]
```

where:

- vPlanFilePath is the absolute path to the vplan file. It is specified in string format. If not specified, the hierarchy is taken from the metrics.
- hierarchy is the hierarchy tree of the entity to be used for calculation. It is specified in string format and can include wildcard characters.
- metricAttribute is the name of the attribute of the entity that is used for the calculation. It is specified in string format.

Example:

The following script shows BLOCK_AVERAGE_GRADE of instance uart/top/u1 in vPlan file myplan.vplanx vs.

```
proc compute_attribute_value(attributes) {
    setAttributeValue [calculateVplanGrade /hm/belanger/myplan.vplanx uart/top/u1
        BLOCK_AVERAGE_GRADE];
}
```

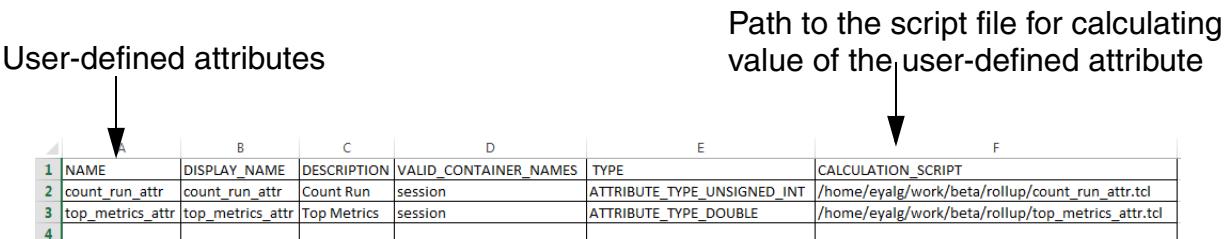
Note: The calculation scripts must be accessible to the server.

2.18.2 Define User-defined Attributes in the CSV File

After writing the calculation script file, define the user-defined attribute in the CSV file that will use this calculation script file to calculate the value of the user-defined attribute.

[Figure 2-40](#) on page 121 shows a sample CSV file with new computed user-defined attributes.

Figure 2-40 CSV File With Computed User-defined Attributes



	B	C	D	E	F
1	NAME	DISPLAY_NAME	DESCRIPTION	VALID_CONTAINER_NAMES	TYPE
2	count_run_attr	count_run_attr	Count Run	session	ATTRIBUTE_TYPE_UNSIGNED_INT
3	top_metrics_attr	top_metrics_attr	Top Metrics	session	ATTRIBUTE_TYPE_DOUBLE
4					/home/eyalg/work/beta/rollup/count_run_attr.tcl
					/home/eyalg/work/beta/rollup/top_metrics_attr.tcl

For example, in this case, two new attributes `count_run_attr` and `top_metrics_attr` are defined. The `CALCULATION_SCRIPT` column specifies the path of the script file that evaluates the value of these user-defined attributes.

Note: vManager installation includes a csv file (`predefined_rollup_attributes.csv`) and matching calculation scripts for the following attributes and their aggregation methods:

Attribute	Aggregation Method
First Failure Description	UniqueValueCount
First Failure Module	UniqueValueCount
First Failure Tool	UniqueValueCount
Host	UniqueValueCount
Owner	UniqueValueCount
SVE Name	UniqueValueCount
Target	UniqueValueCount
Overall Average Grade	Average (metric based)

The CSV file named `predefined_rollup_attributes.csv` is located at the following location:

```
<install_dir>/tools.lnx86/vmgr/files/rollup_scripts/
```

By default, these attributes are not available in vManager. If required, you can import this CSV file in vManager. In addition, you can edit this file and select a subset of attributes, as required.

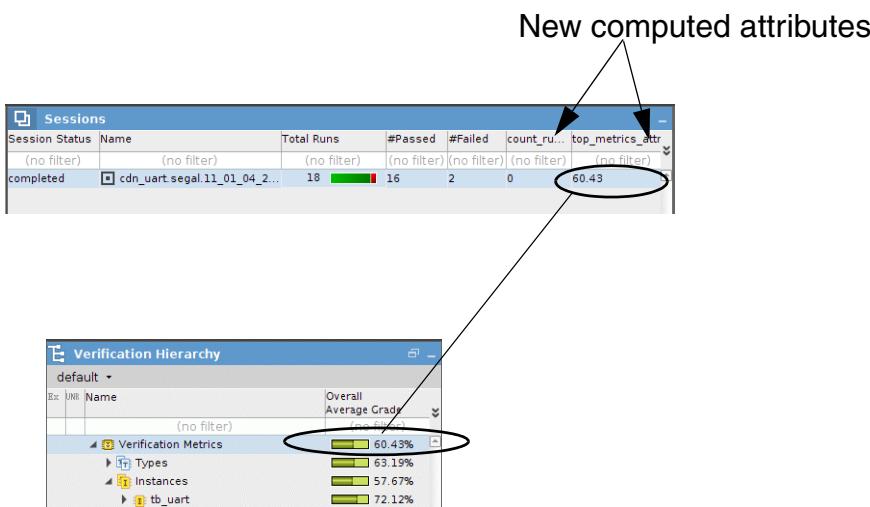
2.18.3 Import the CSV File

After defining the attributes in the CSV file, you need to import the CSV file so that the user-defined attributes are available in vManager.

The attributes defined in the CSV file are available along with other attributes.

You can add those attributes to the Sessions table, and run the Recalculate UDA action to update the values of these attributes. [Figure 2-41](#) on page 122 shows the updated rolled up values.

Figure 2-41 CSV File With Computed User-defined Attributes



The values shown for the user-defined computed attributes are rolled up to the session level.

2.18.4 Recalculate UDA

You can do any of the following to update the values of computed UDA:

- Select the session and then click the *Recalc UDA* button in the *Sessions* toolbar.
- Select the session and then select *Recalculate UDA Script* option in the Regression menu.
- Select the session, right-click, and then select *Recalculate UDA Script* option from the popup menu.

Note: In the CLI mode, you can run the recalc_uda command to update the values.

Note: Other than the explicit requests to recalculate UDA (as stated above), the rolled up attributes are updated automatically after the launched session has ended, after the rerun chain has ended, or a report has been requested.



If there is a change to the session and a *recalc UDA* action is required, then a * indication is shown in the *Name* field, as shown in [Figure 2-42](#) on page 123.

Figure 2-42 Report Sessions

A * indicates that recalc UDA action is required.

Session Status	Name	Total Runs
(no filter)	(no filter)	(no filter)
completed	test1_13_01_16_11_56_36_0282_20*	1
completed	top_sn1_13_01_16_13_42_26_0192_23	1
completed	compact_mike	3

When you notice a * indication in the session name attribute, run the *recalc UDA* action to reflect the new updated values.

2.19 Generating Charts

The *Create Chart* drop-down button on the *Sessions* toolbar of the Regression center has the following options for generating charts:

- Chart all Session-runs (Default) —To generate a chart for all the sessions.
- Chart Selected Session-runs—To generate a chart for the selected sessions.

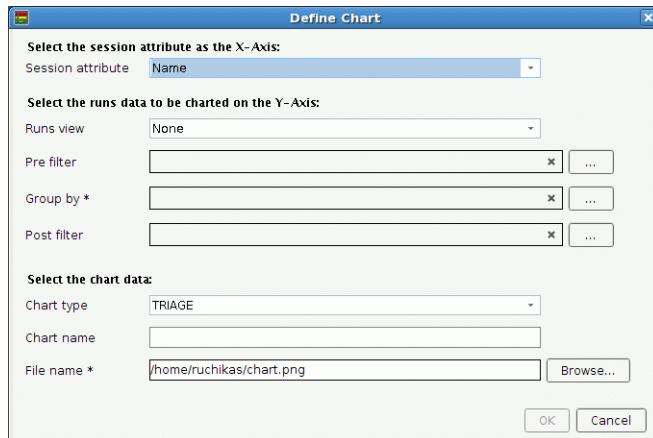
Note: You can also select these options from the *Regression* menu.

To generate charts, perform the following steps:

1. Select the session(s) for which you want to generate a chart.
2. Click *Create Chart* drop-down button on the *Sessions* toolbar, and select any of the following:
 - a. Chart all Session-runs (Default) —To generate a chart for all the sessions.
 - b. Chart Selected Session-runs—To generate a chart for the selected sessions.

The *Define Chart* dialog box is displayed, as shown in [Figure 2-43](#) on page 124.

Figure 2-43 Define Chart



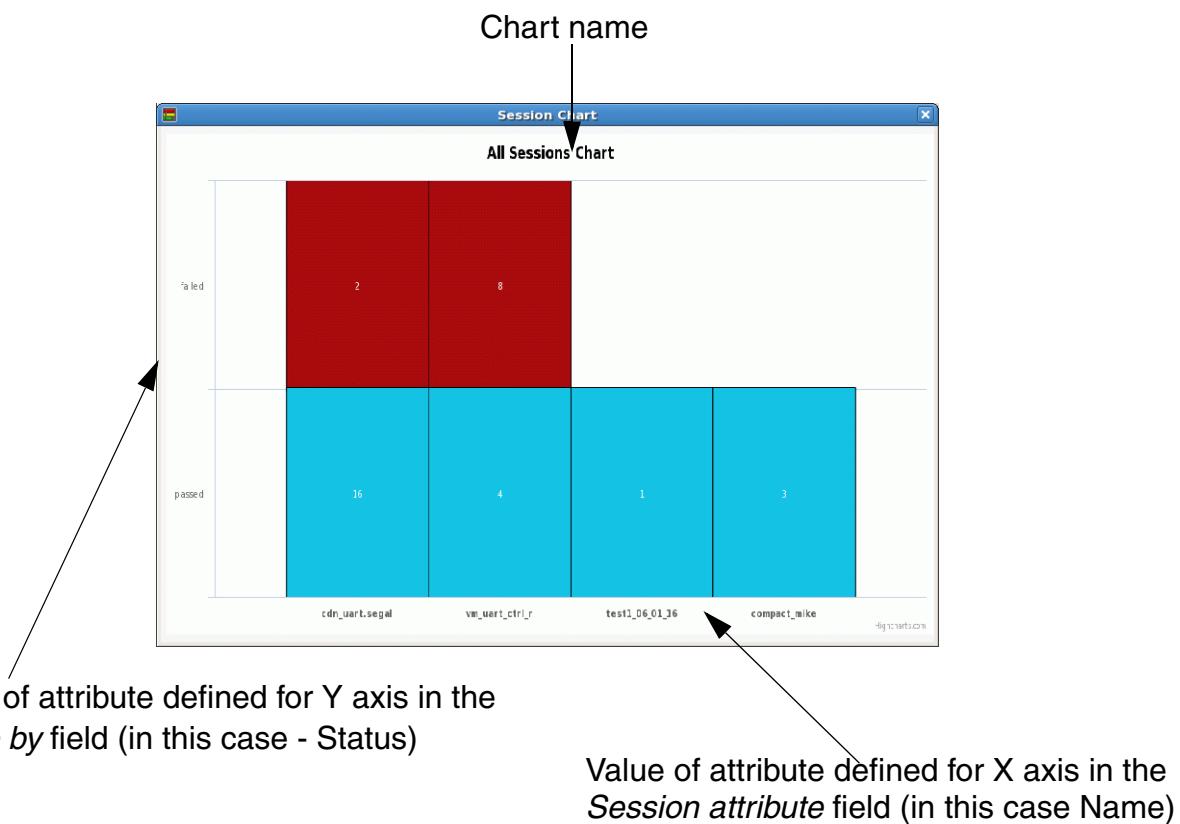
3. In the *Session attribute* field, specify the attribute to be shown on the X axis of the chart. By default, *Name* (the name of the session) is taken as the value to be shown on the X-axis. You can click the drop-down and select the attribute you want to show on the X-axis.
4. Now specify the data for Y-axis.
 - a. You can select the runs view to be used for generating the chart from the *Runs view* drop-down field.
 - b. In the *Pre filter* field, specify the filter to be applied on runs prior to grouping. Clicking on the ... button next to the *Pre filter* field will open the *Edit filter* dialog box, where you can define the required filter.
 - c. In the *Group by* field, specify the attribute to be shown on the Y axis of the chart. The *Group by* field is a mandatory field. Clicking on the ... button next to the *Group by* field will open the *Attribute Selector* dialog box. You can select the attributes, as required.
 - d. In the *Post filter* field, specify the filter to be applied on runs after the grouping. Clicking on the ... button next to the *Post filter* field will open the *Edit filter* dialog box, where you can define the required filter.
5. In the *Chart type* field, specify the type of chart (triage, stacked, column, line) to be created. By default, a *Triage* type chart is created.
6. Specify the name of the chart in the *Chart name* field. It will appear at the top of the chart as the chart heading.
7. In the *File name* field, specify the name and location where the chart (in png format) will be stored. By default, it takes the location as the location from where you invoked

vManager and the file name is taken as `chart.png`. You can change the location and name, as required.

8. Click *OK*.

Figure 2-44 on page 125 shows the generated chart.

Figure 2-44 Generated Chart



The generated chart is shown in a separate window named *Sessions Chart*. This chart is also saved in a PNG format at the location specified in the *File name* field.

You can also generate these charts using the CLI command [triage chart](#).

2.20 Taking Snapshots

To take snapshots from the *Regression* center, perform the following steps:

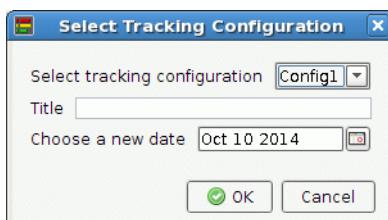
1. Select the session in the *Sessions* table of *Regression* activity center.

2. Right-click and select *Take snapshot* from the pop-up menu.

Note: Alternatively, you can select *Take snapshot* from the Regression menu or click *Take snapshot* button on the Tracking toolbar.

This will open the *Select Tracking Configuration* dialog box, as shown in [Figure 2-45](#) on page 126.

Figure 2-45 Select Tracking Configuration

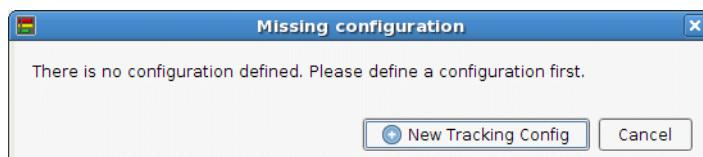


3. Select the tracking configuration to which the snapshot should be associated from the *Select tracking configuration* drop-down.
4. Specify the title of the snapshot in the *Title* field.
5. Specify the snapshot date in the *Choose a new date* field.
6. Click *OK*.

 **Important**

If no tracking configuration is defined at the time of taking snapshots from the *Regression* center, then a message appears, as shown in [Figure 2-46](#) on page 126.

Figure 2-46 Missing Tracking Configuration



You can do either of the following:

- Cancel the operation by clicking the *Cancel* button,
- Define the tracking configuration by clicking the *New Tracking Config* button. This will take you to the Tracking center and also open the *New Tracking Config* dialog

box. You can then define the new tracking configuration, as required. For more details, see [Creating a Configuration](#) on page 325.

2.21 Generating Sessions Report

Using vManager you can generate a session report in an HTML format and then publish it on the internet. Generating an HTML report is useful for users who want to make the session results available for viewing to people who might not have vManager installed with them.

The *Report* drop-down button on the *Report* toolbar has the following options for generating reports:

- All (Default)—To generate a sessions report for all the launched or imported sessions.
- On Selected Entity—To generate a sessions report for the selected session.
- Runs—To generate a runs report (list of all runs) from all of the loaded or imported sessions.
- Runs On Selected Entity —To generate a runs report (list of all runs) from the selected session.
- Summary Report—To generate a summary report. For more details, see [Generating Summary Report](#) on page 51.

Note: You can also select these options from the Regression menu.

Example: Generating Sessions Report for all the Loaded Sessions

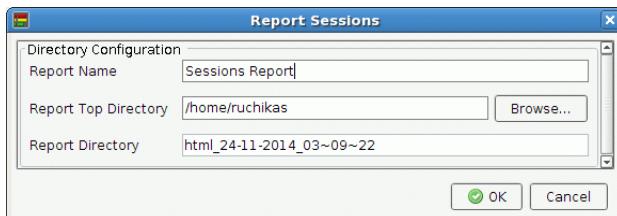
For example, to generate a sessions report for all the launched or imported sessions:

1. Click the *Report* button in the *Report* toolbar.

Note: Alternatively, you can click the down-arrow next to the *Report* button in the *Report* toolbar and select *All (Default)*.

The *Report Sessions* dialog box is displayed, as shown in [Figure 2-47](#) on page 128.

Figure 2-47 Report Sessions



2. Specify the name of the report in the *Report Name* text box. By default, the name is specified as *Sessions Report*. This name appears on the HTML report. For example, specify the name as *All Loaded Sessions*.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`. For example, change the name as `all_sessions`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 2-48](#) on page 128.

Figure 2-48 Report Done



This dialog box shows the location where the report is generated.

6. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 2-49](#) on page 129 shows the generated HTML report.

Figure 2-49 Sessions Report

Report name Report location

File Edit View History Bookmarks Tools Help

All Loaded Sessions

Generated by ruchikas on 2014/11/24 15:10:39 (host ldvopt311)

All Loaded Sessions

Sorting: [Start Time (DESCENDING)]

No.	Failures	Warnings	Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner
1	0	0	completed	compact_mike	5	5	0	0	0	0	Wed Jan 05 02:44:28 IST 2011	ruchikas
2	0	0	completed	cdn_uart.segal.11_01_04_23_14_13_1797	18	16	2	0	0	0	Wed Jan 05 02:44:13 IST 2011	segal

List of sessions

The report shows the information such as, who generated the report and when the report was generated. You can sort table data by clicking on the table headers. You can also click on the hyperlinks available in the reports to view more detailed information about it.

Example: Generating Runs Report for all the Loaded Sessions

For example, to generate a runs report (list of all runs) from all of the loaded or imported sessions:

1. Click the down-arrow next to the *Report* button in the *Report* toolbar and select *Runs*.

The *Report Sessions* dialog box is displayed, as shown in [Figure 2-50](#) on page 129.

Figure 2-50 Report Sessions



2. Specify the name of the report in the *Report Name* text box. By default, the name is specified as *Sessions Report*. This name appears on the HTML report. For example, specify the name as *All Runs*.

3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report file in the *Report File* text box. By default, it shows the name as `html_<timestamp>.html`. For example, change the name as `all_runs.html`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 2-51](#) on page 130.

Figure 2-51 Report Done



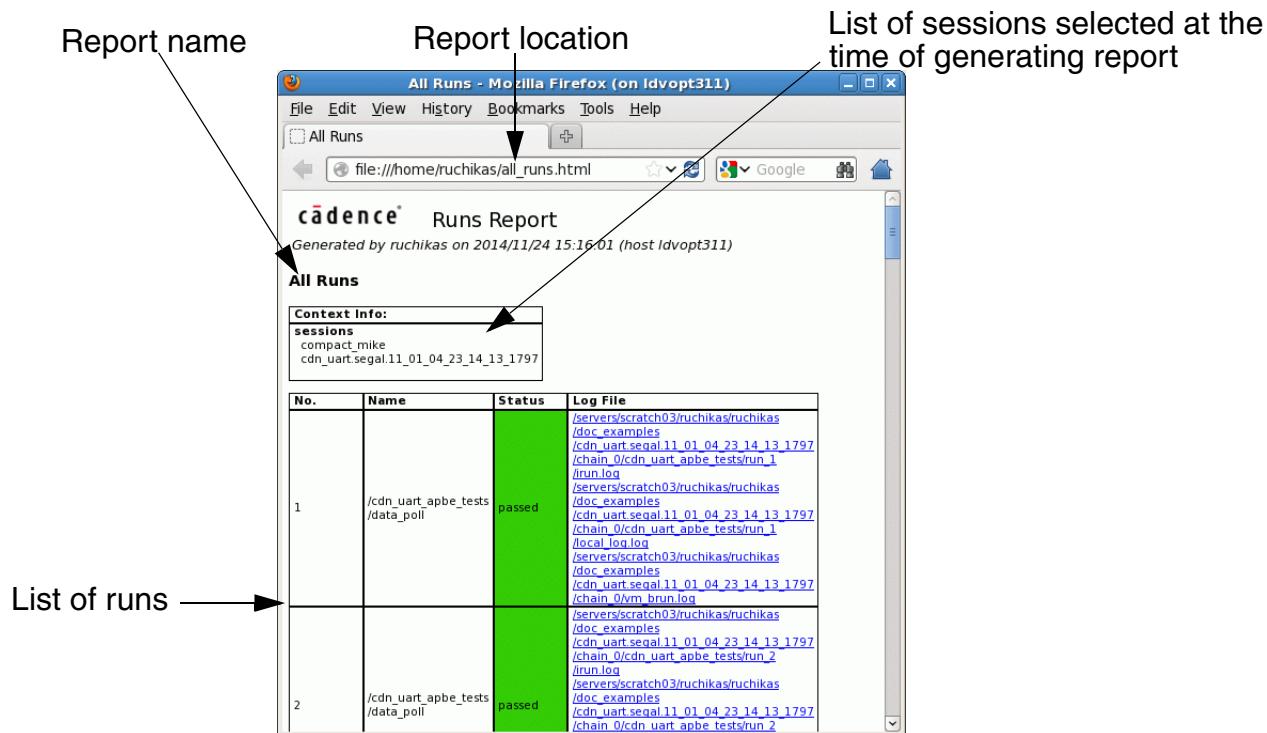
This dialog box shows the location where the report is generated.

6. Click *OK*.

You can navigate through the location shown above, and then open the HTML file in a Web browser.

[Figure 2-52](#) on page 131 shows the generated HTML report.

Figure 2-52 Runs Report



The runs report shows information such as, who generated the report and when the report was generated. The report lists the runs in the loaded sessions. For each run, information such as, run number, name, status, and log file(s) is shown. You can click the links in the *Log File* column and view the log file.

Note: Operations such as filtering, grouping, or sorting cannot be applied on these reports.

You can also generate this report from the CLI mode using the [report_runs](#) command.

2.22 Invoking Analysis Center

From the *Regression* menu, you can invoke Analysis center using any of the following options:

- Analyze Tests
- Analyze Metrics
- Analyze vPlan
- Analyze Failures

- Analyze All Runs
- Analyze Pass Runs
- Analyze Failed Runs
- Analyze Incomplete Runs

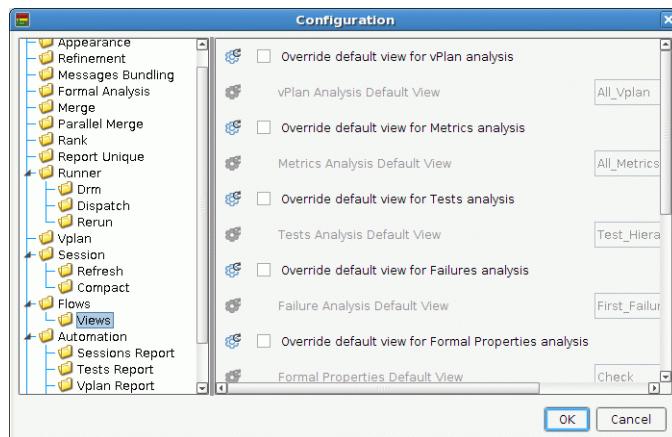
Note: These options are also available on the *Analyze* toolbar.

Note: When you select any of these options a separate page is opened with a default view set for that page. You can change that default view in the *Configuration* dialog box by:

1. Select *View* —> *Configuration*.
2. Select *Views* under the *Flows* folder in the left pane.

The Views options are displayed, as shown in [Figure 2-53](#) on page 132.

Figure 2-53 Configuration—Views Options



3. In this dialog box, you can change the default views for vPlan, metrics, tests, failures, all runs, pass runs, failed runs, and incomplete runs, as required.
4. After selecting the desired views from the given drop-down menus click *OK*.

This will set your specified view as the default view.

Analyzing Runs and Failures

After loading the session, you view the session results for passed and failed status. You then analyze the session further to identify failed runs and the ones that must be run again. vManager allows you to do further analysis in the *Analysis* center, where you can analyze runs, failures, metrics, or verification plan. This chapter discusses how to analyze runs and failures. Analyzing metrics and verification plans is discussed in later chapters.

This chapter covers the following topics:

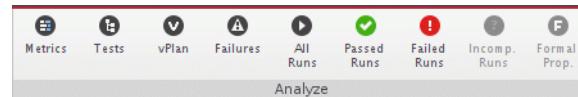
- [Launching Analysis Center](#) on page 133
- [Analysis Center Interface](#) on page 134
- [Analyzing Runs](#) on page 154
- [Analyzing Failures](#) on page 165
- [Analyzing Formal Properties](#) on page 172
- [Grouping Runs](#) on page 180
- [Filtering Runs](#) on page 184
- [Ranking Runs](#) on page 184
- [Correlating Runs](#) on page 195
- [Generating Tests Report](#) on page 204
- [Integrating vManager with Indago Debug Analyzer](#) on page 209

3.1 Launching Analysis Center

The *Analysis* center of vManager allows you to analyze runs, failures, metrics, and verification plans. The *Analysis* center of vManager can be launched from the *Regression* center in any of the following ways:

- Select a session in the *Sessions* table and click any of the buttons in the *Analyze* toolbar, as shown in [Figure 3-1](#) on page 134.

Figure 3-1 Regression Center—Launching Analysis Center

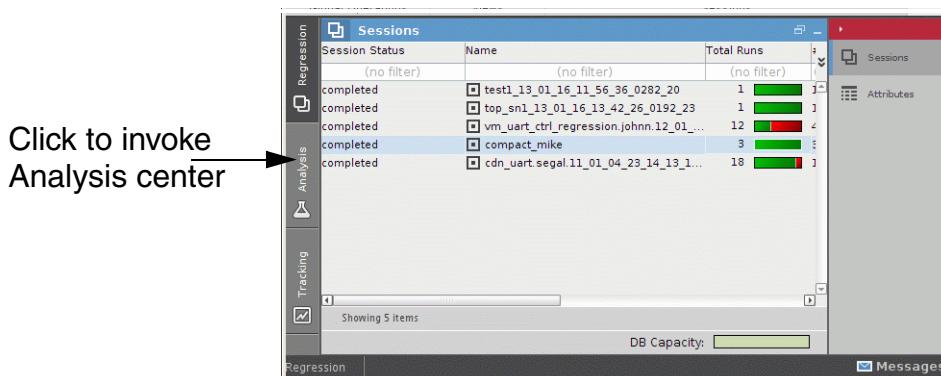


Click any of the buttons to invoke Analysis center

Note: These options are also available in the *Regression* menu and also when you right-click a session.

- Click the *Analysis* tab in left side of the *Regression* center, as shown in [Figure 3-2](#) on page 134.

Figure 3-2 Regression Center—Launching Analysis Center



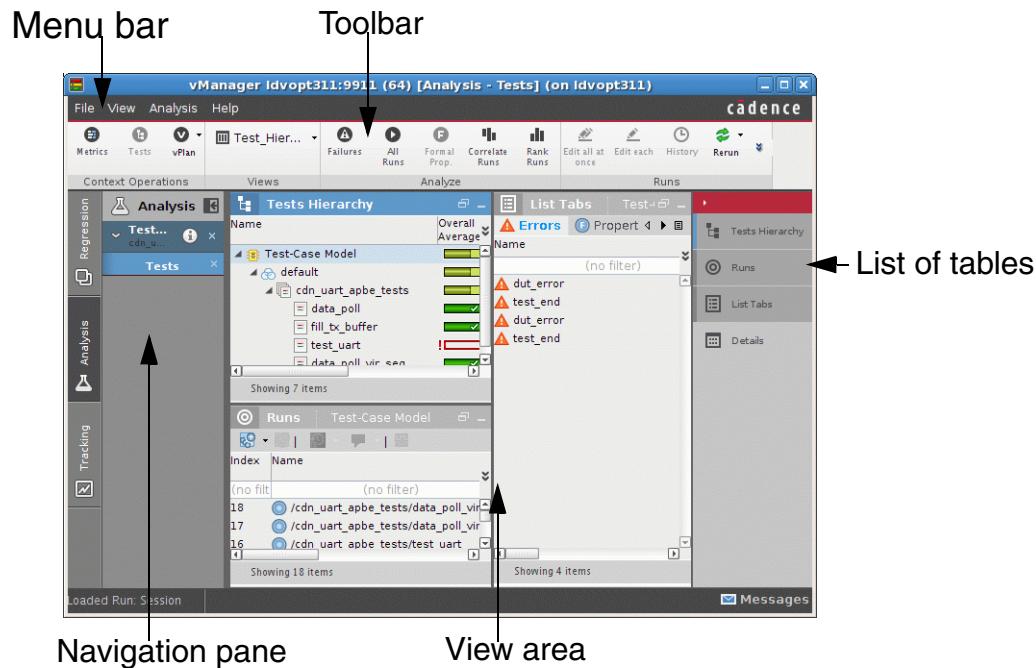
3.2 Analysis Center Interface

The information shown in the *Analysis* center is based on the option you select from the *Analyze* toolbar to launch the *Analysis* center.

For example, to launch *Analysis center* and start with the analysis, you select the *Tests* from the *Analyze* toolbar.

[Figure 3-3](#) on page 135 shows the *Analysis center* with the *Tests* page.

Figure 3-3 Analysis Center



The main components of the *Analysis* center are:

- Menu Bar
- Toolbars
- Analysis Navigation Panel
- View Area

3.2.1 Menu Bar

Figure 3-4 on page 135 shows the menu bar available in the *Analysis* center.

Figure 3-4 Menu Bar



Menu options *File*, *View*, and *Help* are common to all activity centers and are discussed in the Chapter 1, “Getting Started with vManager”.

The *Analysis* menu is related to the *Analysis* center and is discussed here.

From the *Analysis* menu, you can select any of the following options:

- Analyze Metrics—To launch a separate Metrics page for analyzing metrics. For more details, see [Analyzing Metrics](#) on page 217.
- Analyze Tests —To launch a separate Tests page for analyzing tests.
- Analyze vPlan—To load a verification plan for analysis. For more details, see [Analyzing Verification Plans](#) on page 251.
- Scripts Manager—To add user-defined actions on toolbar. For more details, see [User-Defined Actions](#) on page 71.
- New vPlan—To launch vPlanner to create a new vPlan. For more details, see [Analyzing Verification Plans](#) on page 251.
- Edit vPlan —To launch vPlanner with the current verification plan loaded for editing. For more details, see [Analyzing Verification Plans](#) on page 251.
- Reload vPlan—To reload the loaded verification plan.
- Context information—To view the name and location of the loaded sessions, coverage model files, loaded verification plans, and the loaded refinement files. For more details, see [Context](#).
- Source Map —To map the source to new paths. For more details, see [Source Map](#).
- Refresh —To refresh the test tree
- Analyze Failures—To launch a separate page for analyzing failures. For more details, see [Analyzing Failures](#) on page 165.
- Analyze All Runs—To launch a separate page for analyzing all of the runs related to selected test or test group. For more details, see [Analyzing Runs](#) on page 154.
- Analyze Formal Properties—To launch a separate page for analyzing formal properties. For more details, see [Analyzing Formal Properties](#) on page 172.
- Correlate Runs—To correlate runs. For more details, see [Correlating Runs](#) on page 195.
- Rank Runs—To rank runs. For more details, see [Ranking Runs](#) on page 184.
- Edit all at once—To add comments or description to all the selected runs together.
- Edit each—To add comments or description to the selected runs one by one.
- Attribute change history—To view changes made to attributes of the selected run.
- Rerun Runs—To rerun the selected run or group of runs.
- Create Context—To create a new context from the selected runs or messages.

- Stop Run—To stop selected runs.
- Open run directory—To open a terminal window and show the selected run as the current working directory.
- Reports—To generate run reports.

3.2.2 Toolbars

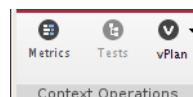
The *Analysis* center has following toolbars:

- Context Operations
- Scripts
- Planner
- Settings
- Views
- Analyze
- Runs
- Report

3.2.2.1 Context Operations

Figure 3-5 on page 137 shows the *Context Operations* toolbar.

Figure 3-5 Context Operations Toolbar



The *Context Operations* toolbar has following options:

- Metrics—To launch a separate page for analyzing metrics. For more details, see [Analyzing Metrics](#) on page 217.
- Tests—To launch the test hierarchy tree. For more details, see [Analyzing Runs](#) on page 154.
- vPlan—To load a verification plan for analysis. For more details, see [Analyzing Verification Plans](#) on page 251.

3.2.2.2 Scripts

Figure 3-6 on page 138 shows the *Scripts* toolbar.

Figure 3-6 Scripts Toolbar



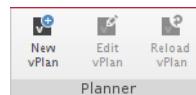
The *Scripts* toolbar has the following options:

- *Scripts Manager*—To add user-defined actions on toolbar. For more details, see [User-Defined Actions](#) on page 71.

3.2.2.3 Planner

Figure 3-7 on page 138 shows the *Planner* toolbar.

Figure 3-7 Planner Toolbar



The *Planner* toolbar has following options:

- **New vPlan**—To launch vPlanner to create a new vPlan. For more details, see [Analyzing Verification Plans](#) on page 251.
- **Edit vPlan**—To launch vPlanner with the current verification plan loaded for editing. For more details, see [Analyzing Verification Plans](#) on page 251.
- **Reload vPlan**—To reload the loaded verification plan.

3.2.2.4 Settings

Figure 3-8 on page 139 shows the *Settings* toolbar.

Figure 3-8 Settings Toolbar

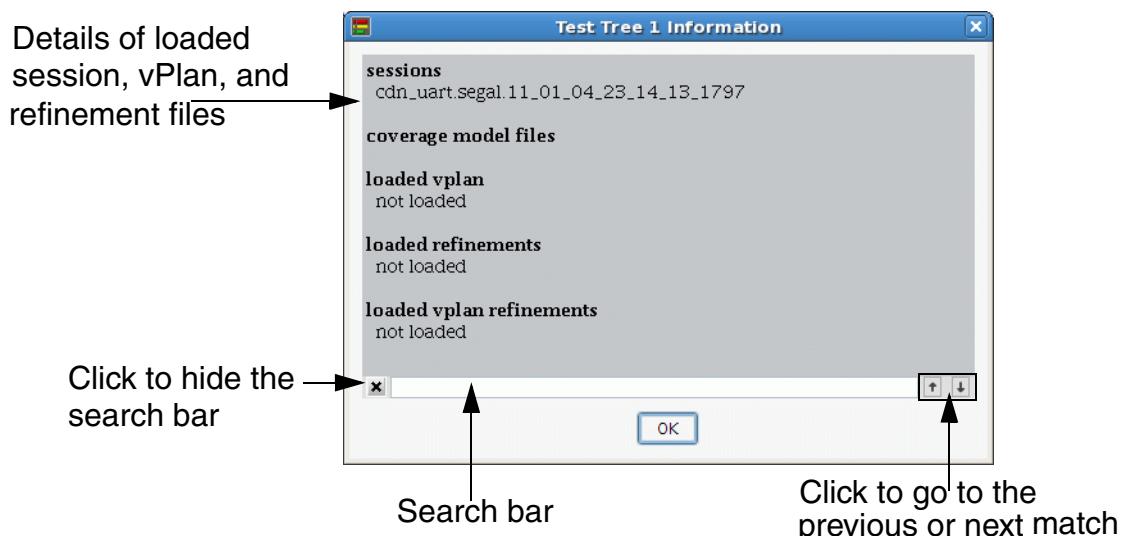


The *Settings* toolbar has following options:

- Context info—To display the context information

It allows you to view the name and location of the loaded sessions, coverage model files, loaded verification plans, and the loaded refinement files. When you click the *Context info* button, the *Context Information* dialog box is displayed, as shown in [Figure 3-9](#) on page 139.

Figure 3-9 Context Information Dialog Box



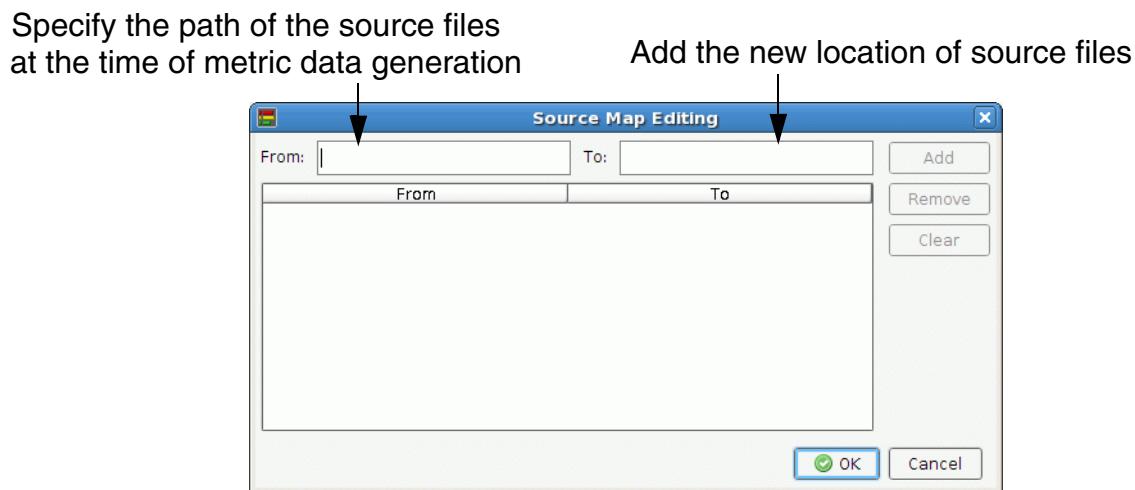
In addition to viewing the context information, you can perform a search on the information displayed in the *Context Information* dialog box.

To quickly search for a specific text, type the search text in the text box. As you type the text in the text box, the text gets highlighted in the context information. You can use the *Up* and *Down* arrow keys to go to previous or next match. You can hide the search bar by clicking the *X* icon shown on the search bar. To reopen the search bar, press *Ctrl + F* keys.

- A Source Map button — It allows you to map the source to new paths or remove existing mapped sources. This feature is useful if source files are moved, or are referenced differently on different systems. For example, if the source files are moved to a different

location, then during data analysis the *Source* tab page does not show the source code. To resolve the issue and to show the source code, you can map the source path to a new location where the source files are located. For this, click the *Source Map* button in the *Settings* toolbar. When you click the *Source Map* button, the *Source Map Editing* dialog box is displayed, as shown in [Figure 3-10](#) on page 140.

Figure 3-10 Source Map Editing Dialog Box



- ❑ In the *From* text box, specify the path of the source files at the time of metrics data generation.
- ❑ In the *To* text box, specify the new location of the source files.

For example, if the original source location was /home/usr1/temp/src and the new location of the source files is /home/usr1/src_files/vlog, then specify /home/usr1/temp/src in the *From* text box and specify /home/usr1/src_files/vlog in the *To* text box.

- ❑ Click *Add* to map the old path (*From*) to the new path (*To*) and click *OK*.

Note: If the suffix of the source path is same, then instead of specifying the complete path in the *From* and *To* fields, you can consider specifying just the prefix of the path. For example, if the original source location is /home/usr1/temp/src and the new location of the source files is /home/usr2/temp/src, then you can specify /home/usr1 in the *From* text box and /home/usr2 in the *To* text box.

After adding a source map path, you must reload the run to ensure that the *Source* tab page shows the source code based on the path you mapped in the *Source Map Editing* dialog box.

Note: The *Remove* button allows you to remove an already mapped path. You can

select an already mapped path and click *Remove* to delete the path. The *Clear* button allows you to clear all the mapped paths together. For example, to clear all the mapped paths, click the *Clear* button. It will remove all the mapped paths together.

Mapping of source paths is also available in command-line mode of vManager. For details on mapping source paths in command-line mode, see [sourcemap](#) on page 407.

3.2.2.5 Views

[Figure 3-11](#) on page 141 shows the *Views* toolbar.

Figure 3-11 Views Toolbar

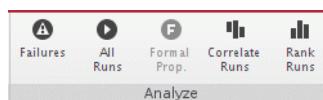


The *Views* toolbar allows you to define views, organize views, set views, and quickly launch a view. For more details, see [Defining and Organizing Views](#) on page 43.

3.2.2.6 Analyze

The buttons in the Analyze toolbar are based on the current context. [Figure 3-12](#) on page 141 shows the *Analyze* toolbar when *Test Tree* context is active.

Figure 3-12 Analyze Toolbar



The *Analyze* toolbar has following options:

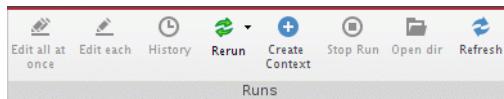
- Failures—To launch a separate page for analyzing failed messages. For more details, see [Analyzing Failures](#) on page 165.
- All Runs—To launch a separate page for analyzing all of the runs. For more details, see [Analyzing Runs](#) on page 154.
- Formal Properties—To launch a separate page for analyzing formal properties. For more details, see [Analyzing Formal Properties](#) on page 172.
- Correlate Runs—To correlate runs. For more details, see [Correlating Runs](#) on page 195.

- Rank Runs—To rank runs. For more details, see [Ranking Runs](#) on page 184.

3.2.2.7 Runs

[Figure 3-13](#) on page 142 shows the *Runs* toolbar.

Figure 3-13 Runs Toolbar



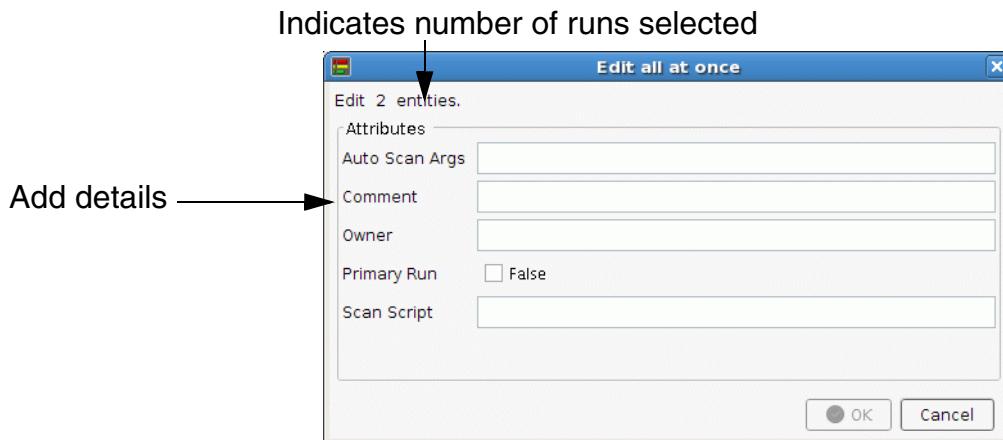
The *Runs* toolbar has the following options:

- [Edit all at once](#)—To add comments, edit owner, or user-defined attributes of all the selected runs together
- [Edit each](#)—To add comments, edit owner, or user-defined attributes of selected runs one by one
- [History](#)—To view changes made to attributes of the selected session
- [Rerun Runs](#)—To rerun the selected run or group of runs
- [Create Context](#)—To create a new context from the selected runs
- Stop Run—To stop selected runs
- Open dir—To open a terminal window and show the selected run as the current working directory.
- Refresh—To refresh the table data.

Edit all at once

The *Edit all at once* option allows you to add or edit attributes of all of the selected runs at once. [Figure 3-14](#) on page 143 shows the *Edit all at once* dialog box.

Figure 3-14 Edit All At Once



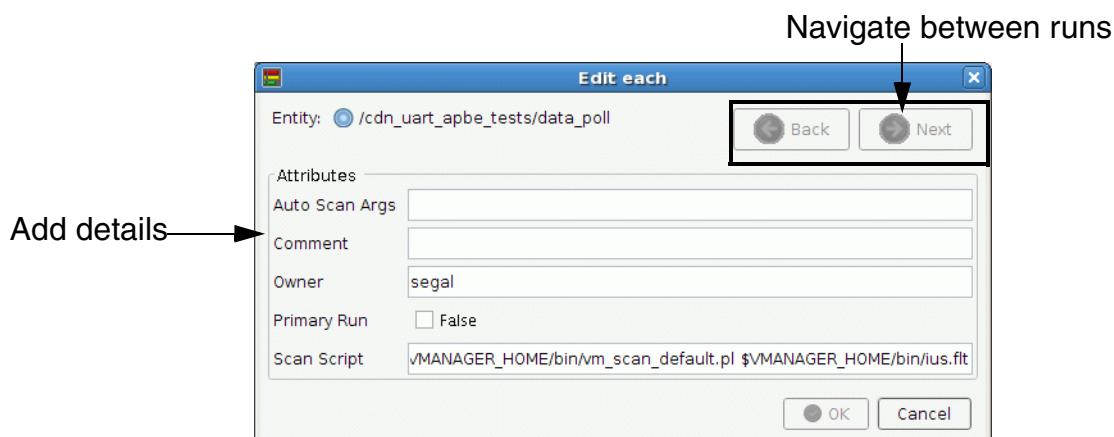
The information added using this dialog box, applies to all of the selected runs.

After you click *OK*, the details added are set and are available in the *Runs* pane and also in the *Attributes* pane.

Edit each

The *Edit each* option allows you to add or edit attributes of the selected runs one by one. [Figure 3-15 on page 143](#) shows the *Edit each* dialog box.

Figure 3-15 Edit Each



Using this dialog box, you can add comments and owner for each run individually. You can also change the scanning settings by editing the attributes *Scan Script* and *Auto Scan*.

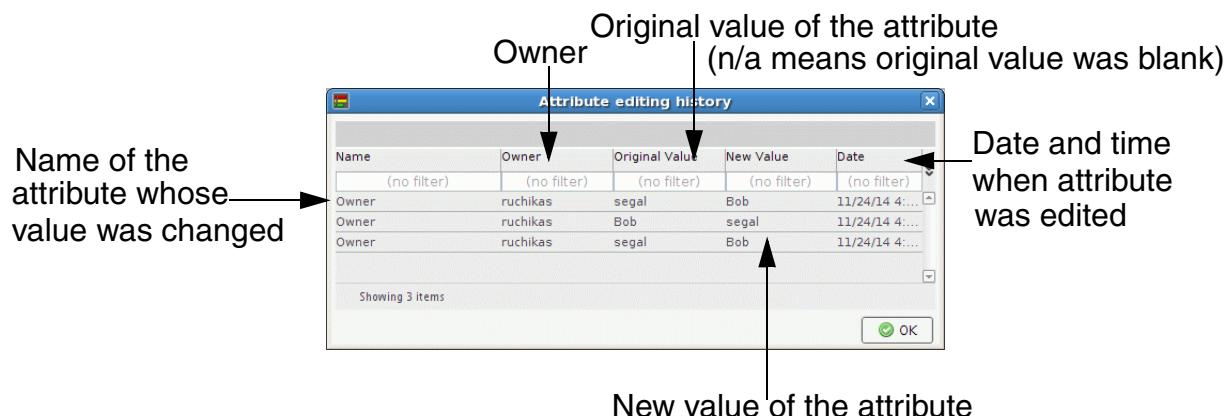
Args. You can add details and then navigate to the next or previous session using the *Back* or *Next* buttons.

The information you add in this dialog box is visible in the *Runs* pane as well as the *Attributes* pane.

History

The *History* option allows you to view attribute change history of the selected run. [Figure 3-16 on page 144](#) shows the *Attribute editing history* dialog box.

Figure 3-16 Attribute Editing History



This dialog box maintains the history of all the changes made to different attributes of the selected run.

Rerun Runs

The *Rerun* drop-down has following options:

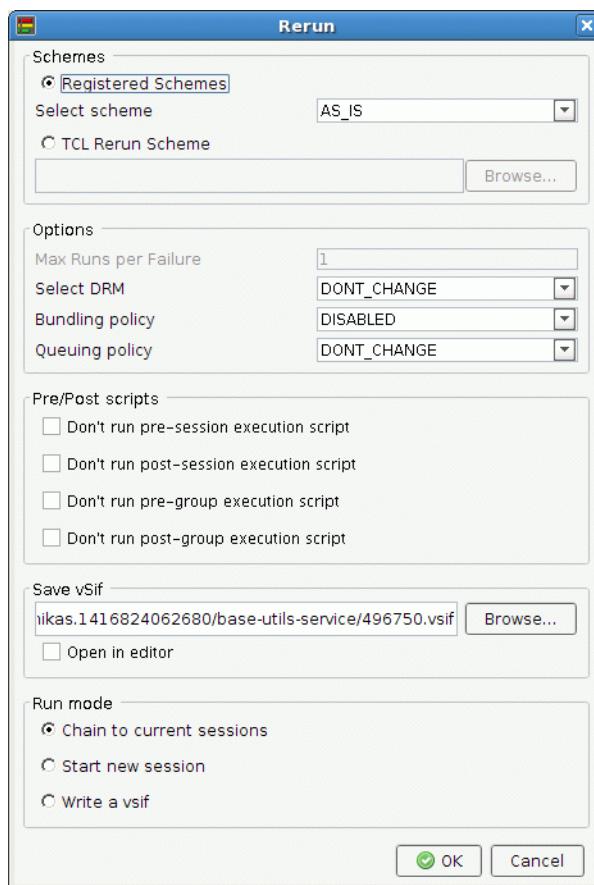
- Rerun — To rerun the selected runs.
- Manager — To register a user-defined rerun scheme. For more details on registering a user-defined rerun scheme, see [Incisive vManager Installation and Configuration Guide](#).
- Rerun all runs — To rerun all the runs that are currently in the Runs table.
- Rerun all failures — To rerun all the failures that are currently in the Failures table.

To rerun selected runs, all of the runs, or all failures:

1. Select the run or runs you want to re-execute. This step is optional if you want to rerun all the runs in the Runs table.
2. In the *Runs* toolbar, click the *Rerun* drop-down and do any of the following:
 - Select *Rerun* (if you want to rerun selected runs)
 - Select *Rerun all runs* (if you want to rerun all the runs in the Runs table)
 - Select *Rerun all failures* (if you want to rerun all the failures in the Failures table)

The *Rerun* dialog box appears, as shown in [Figure 3-17](#) on page 145.

Figure 3-17 Rerun



3. Select the rerun scheme. The rerun scheme defines how the rerun is executed. You can specify any of the following:
 - Registered Schemes — The predefined schemes and user-defined registered schemes are available in the *Select scheme* drop-down list. If you want to execute

the run as specified in the original vsif, select AS_IS. Otherwise, select any of the following schemes from the drop-down list:

- BATCH—To execute the run in batch mode with timeout value as specified in the original vsif.
- BATCH_DEBUG—To execute the run in batch-debug mode with timeout value as specified in the original vsif.
- INTERACTIVE—To execute the run in interactive mode with timeout value as 0, which means timeout is disabled.
- INTERACTIVE_DEBUG—To execute the run in interactive-debug mode with timeout value as 0, which means timeout is disabled.
- INDAGO_BATCH_DEBUG—To enable recording of IDA data at the time of rerunning the runs.
- RESCAN_LOGS—To rescan logs without rerunning the runs. This option is supported only with the *Chain to current sessions* run mode option.

Note: If there are any user-defined registered schemes, they will also show in this drop-down list. For more details on user-defined rerun schemes, see [Incisive vManager Installation and Configuration Guide](#).

- TCL Rerun Scheme — This option allows you to use a user-defined rerun scheme from a TCL file without registering it. You can browse to the location where the TCL file is saved and select the file you want to use.
4. In the *Max Runs per Failure* field, specify the maximum number of runs to be rerun from each failure kind. The default value specified is 1. You can increase the value, as required.

Note: The maximum number of runs per failure is limited to 1000 runs. In addition, the total rerun runs (from all failures) is limited to 10,000 runs.

5. Select a different DRM from the drop-down list, or leave it as DONT_CHANGE. You can select from any of the following options:
- LSF — Specifies the DPL-based integration with IBM Platform Load Sharing Facility (LSF).
 - SGE — Specifies the DPL-based integration with the open source batch-queuing system, Sun Grid Engine (SGE), supported by Sun Microsystems.
 - LOAD_LEVELER — Specifies the DPL-based integration with IBM's LoadLeveler.
 - PARALLEL_LOCAL — Specifies the DPL-based execution of runs in parallel on the local machine.

- SERIAL_LOCAL — Specifies the DPL-based execution of runs serially on the local machine.

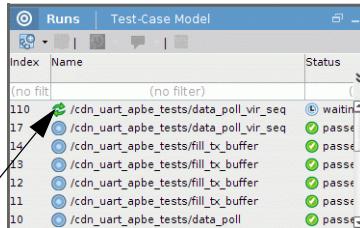
6. Specify the Bundling policy as either *DISABLED* or *GROUP_BASED*.
7. Select a different Queuing policy from the drop-down list, or leave it as DONT_CHANGE. You can select from any of the following options:
 - LONG2SHORT — Makes the longest runs first in the queue. This reduces the “long run tail” at the end of a session.
 - SHORT2LONG — Makes the shortest runs first in the queue. This allows you to get the first test results from the regression as quickly as possible.
 - ROUND_ROBIN — Runs the tests in an arbitrary order, but creates a round-robin ordering between the groups and tests with a count greater than 1 or with multiple seeds. This mode is useful for vsifs that are arranged in logical groups, where you want to quickly see each type of test run once.
 - VSIF_ORDER—Runs the tests in the order that they are defined in the vsif.
8. Specify whether to execute the pre-session, post-session, pre-group, and post-group scripts. By default, these scripts are run. To not execute a particular script, select the corresponding check box.
9. Select a Run mode. You can select any of the following options:
 - Chain to current sessions—To chain each run to the original session. In other words, the original vsof file is extended, and the run results are written to a chain_(N+1) directory, where chain_N is the name of the original run directory.
 - Start new session—To execute the rerun as a separate session.
 - Write a vsif—To save the vsif file without executing it.
10. If Run mode is specified as *Start new session* or *Write a vsif*, then you can specify a name and location for the new vsif file in the *Save vSif* text box.

Note: Select the *Open in editor* check box if you want to edit the vsif file. This will open the editor specified in the *Configuration* dialog box (rerun options) and allow editing of vsif. For more details, see [Configuring Rerun Options](#) on page 148.

11. Click *OK*.

The selected runs are re-executed based on the options specified in the *Rerun* dialog box. A different icon is shown for the runs that were re-executed, as shown in [Figure 3-18](#) on page 148.

Figure 3-18 Re-executed Run



Index	Name	Status
(no filter) (no filter)		
110	/cdn_uart_apbe_tests/data_poll_vir_seq	waitin
17	/cdn_uart_apbe_tests/data_poll_vir_seq	passed
14	/cdn_uart_apbe_tests/fill_tx_buffer	passed
13	/cdn_uart_apbe_tests/fill_tx_buffer	passed
12	/cdn_uart_apbe_tests/fill_tx_buffer	passed
11	/cdn_uart_apbe_tests/fill_tx_buffer	passed
10	/cdn_uart_apbe_tests/data_poll	passed

Indicates that the run was re-executed

The Runs pane shows a different icon for the runs that are a rerun of another run. To open the original run, you can right-click the re-executed run and select *Open original run* from the pop-up menu. This will open a Runs analysis page to show the original run.

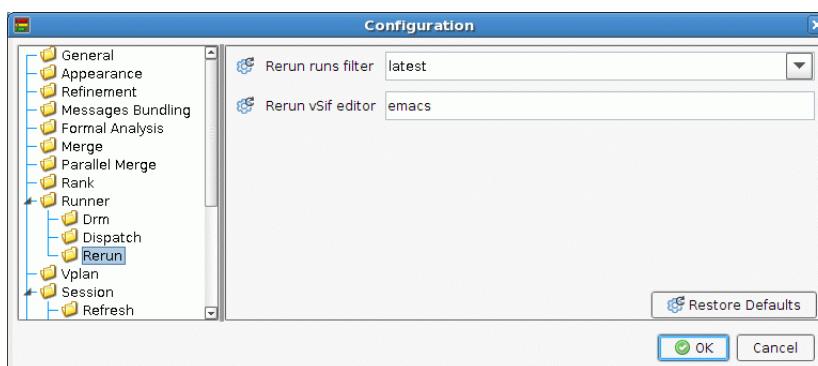
Configuring Rerun Options

To configure rerun options:

1. Select *View* —> *Configuration*.
2. Select the *Rerun* folder in the left pane.

The *Configuration* dialog box displays the Rerun options, as shown in Figure 3-19 on page 148.

Figure 3-19 Configuration (Rerun Options)



3. In this dialog box, you can specify the following:

- ❑ Rerun runs filter—Lets you specify the filter that must be used to show the runs. You can specify any of the following filters from the drop-down list:

- Original: Shows only the original runs, that is, the runs available before the rerun action.
 - Latest: This is the default filter. It shows only the latest runs, that is, only the runs that were last rerun.
 - All: Shows all the runs, that is the original runs as well as the ones that were rerun.
- Rerun vSif editor—Lets you specify the editor in which the vsif file will be opened and edited at the time of rerun.

Specify the values for runs filter and vsif editor, as required and click *OK*.

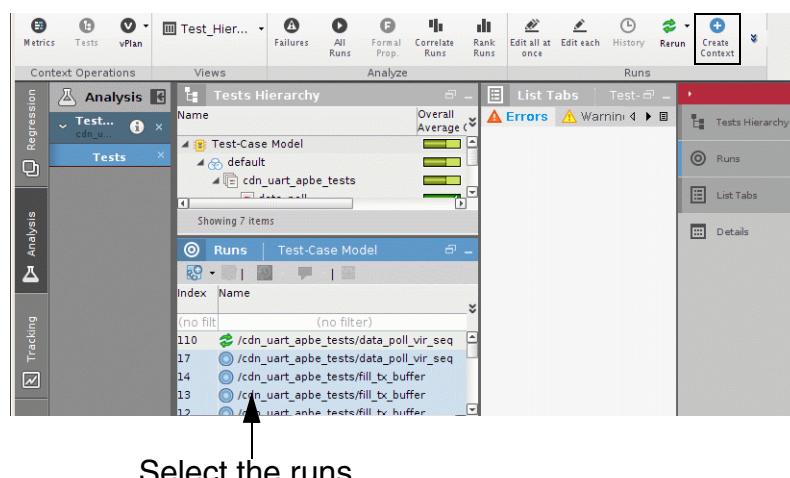
Create Context

The *Create Context* button allows you to create a new context from the selected test group, test, run, or runs. For example, while navigating through the runs, you identify a run of your interest and you want to create a separate context for analyzing this run.

To create a context:

1. Select the runs of your interest, as shown in [Figure 3-20](#) on page 149.

Figure 3-20 Select the Run

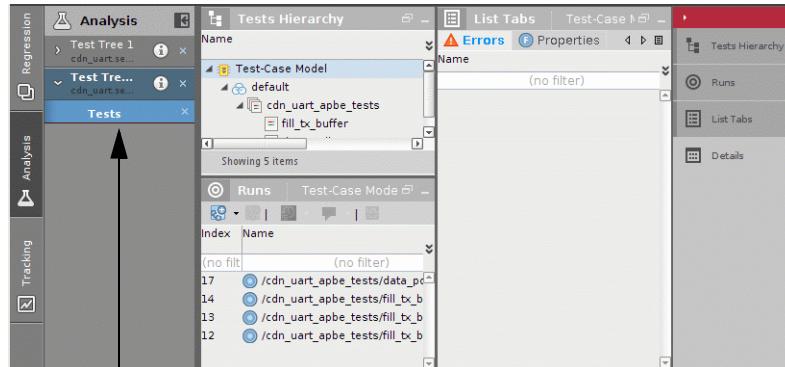


2. Click the *Create Context* button in the Runs toolbar.

Note: You can also double-click on a run to open a separate page or context for analyzing it.

A new context is created, as shown in [Figure 3-21](#) on page 150.

Figure 3-21 New Context Created



Separate context created with the selected run

You can now analyze the selected run in this context. You can also rename the context, as required.

Open Run Directory in Terminal

Using vManager, you can open a terminal window in which the selected run will be shown as the current working directory. You can then navigate through the run directory and also browse its contents.

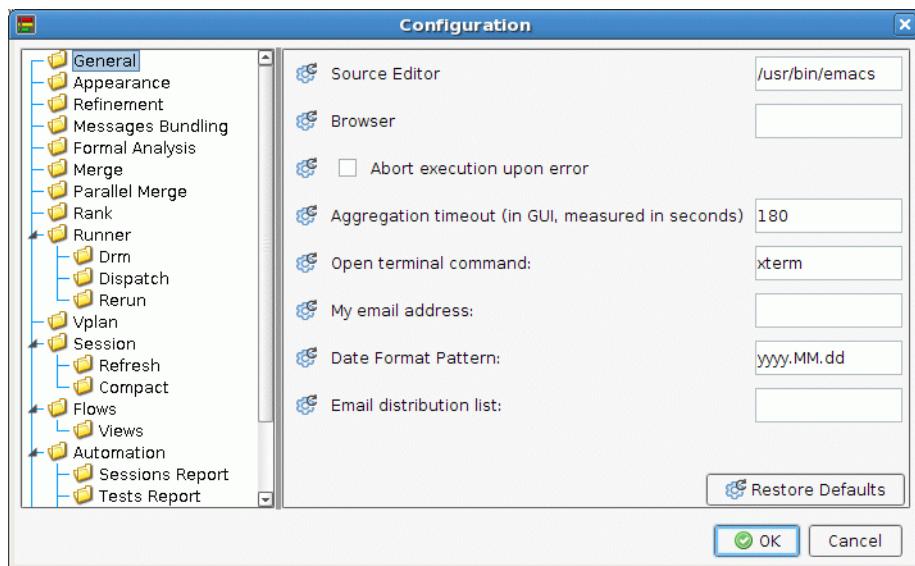
To open a run directory in a terminal window, do any of the following:

- Select the run and select *Open run directory* from the *Analysis* menu.
- Select the run and select *Open dir* button on the *Runs* toolbar.
- Select the run, right-click and select *Open run directory*.

By default, `xterm` is opened as the terminal. However, you can change using the *Configuration* option in the *View* menu. To change terminal in which run directory is opened, select *Configuration* from the *View* menu.

The session *General* options are displayed, as shown in [Figure 3-22](#) on page 151.

Figure 3-22 Configuration—General Options



By default, *Open terminal command* is specified as `xterm`. In case you want to change the terminal, specify it in the text box and click *OK*.

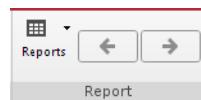
For details on other options available in this page, see [Configuring General Options](#) on page 55.

Note: The value you specify in the *Configuration* dialog box applies to all of the runs.

3.2.2.8 Report

[Figure 3-23](#) on page 151 shows the *Report* toolbar.

Figure 3-23 Report Toolbar



The *Report* toolbar has following options:

- [Report](#)—To generate tests reports
- Selection History buttons—To quickly navigate through the previous selections

vManager maintains a history of selections that you make in the various panes and pages of the vManager window. vManager maintains this history until you reload the

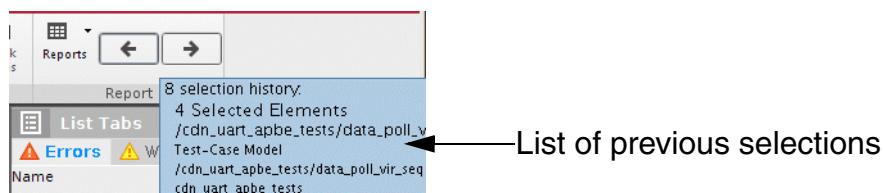
session. You can quickly navigate through those selections using the *Selection History* navigation buttons, as shown in [Figure 3-24](#) on page 152.

Figure 3-24 Selection History Navigation Buttons



When you place the cursor over the navigation buttons, a list of selections is displayed. [Figure 3-25](#) on page 152 displays a list of previous selections.

Figure 3-25 List of Selections

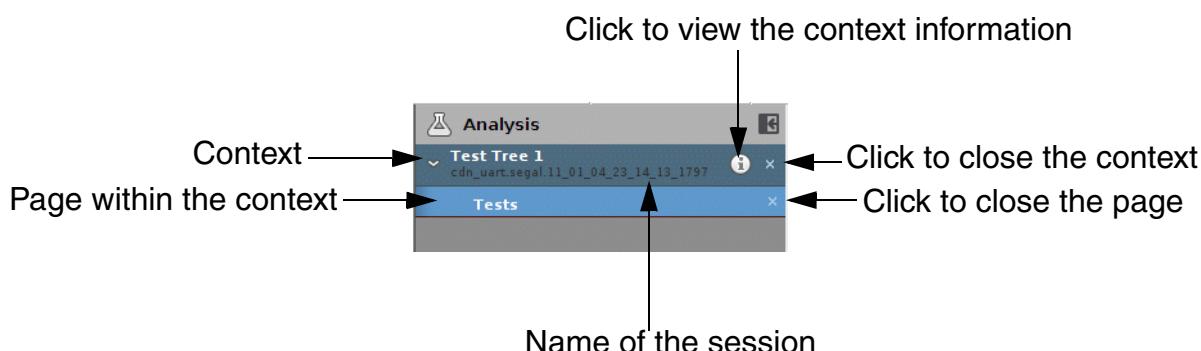


Placing the cursor over the next arrow key will display the selections made after the current selection. You can click the next arrow key to quickly reach the next selection.

3.2.3 Analysis Navigation Panel

[Figure 3-26](#) on page 152 shows the Analysis navigation panel.

Figure 3-26 Analysis Navigation Panel



By default, vManager automatically names the context at the time of creating it. For easy identification, you can rename the contexts, as required. You can also view the context information by clicking the icon named *i* next to the context name. This will open a dialog box

that will show name and location of the sessions, verification plan, and the refinement file related to the context.

You can also right-click on a context to perform any of the following operations:

- Activate—To launch the context
- Close —To close the context
- Rename—To [Rename a Context](#)
- Export —To [Export Active Context](#)

Rename a Context

To rename a context:

1. Right-click the context in the *Navigation* pane, and select *Rename*.

Note: Alternatively, double-click the context name in the *Navigation* pane.

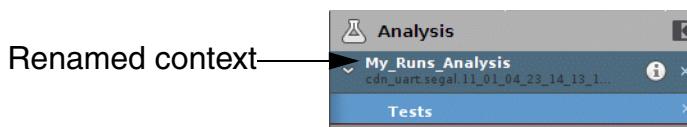
2. The cursor appears in the *Navigation* pane, as shown in [Figure 3-27](#) on page 153 and allows inline editing of the name of the context. Specify the new name for the context and press *Enter*.

Figure 3-27 Rename a Context



For example, if you specify the name as `My_Runs_Analysis`, then the renamed context is shown in the *Navigation* pane, as shown in [Figure 3-28](#) on page 153.

Figure 3-28 Navigation Pane with Renamed Context



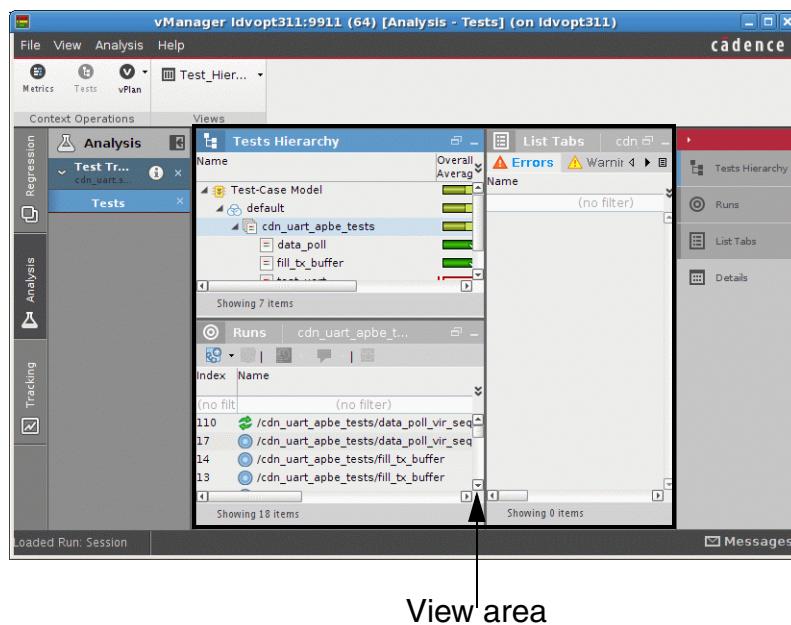
Notice that the `TEST_TREE 1` is renamed as `My_Runs_Analysis`.

Similarly, you can also rename the pages within the context.

3.2.4 View Area

The view area of the *Analysis* center shows different panes/tables depending on the selection you made at the time of launching it. For example, if you selected *Tests*, then the *Analysis* center will show the panes, as shown in [Figure 3-29](#) on page 154.

Figure 3-29 View Area



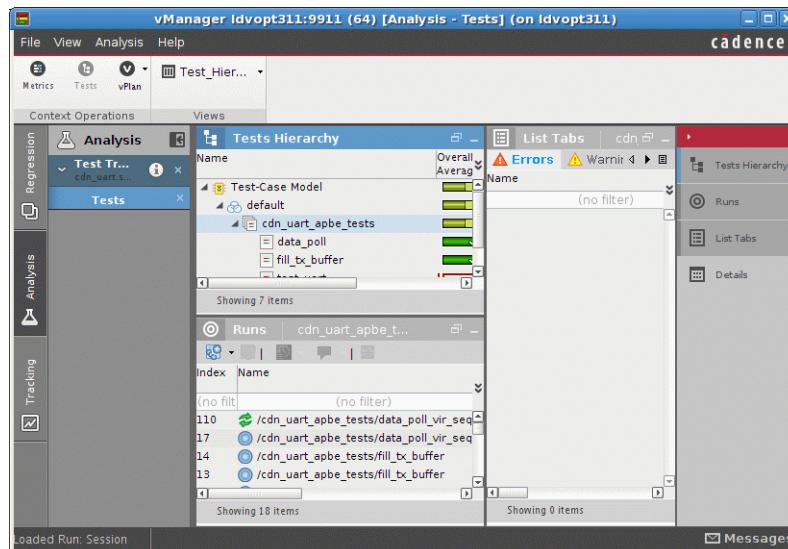
The above screen shows the panes that are shown when you launch the *Tests* page in *Analysis* center.

3.3 Analyzing Runs

Assume that you are currently in the *Regression* center of vManager. After loading the required session, the simplest way to start with analyzing runs is to right-click the session and select *Tests* from the pop-up menu.

This invokes the *Activity* center and launches the *Tests* page, as shown in [Figure 3-30](#) on page 155.

Figure 3-30 Analysis Center with All Tests Page



By default, when you launch *Tests* page, the *Test_Hierarchy* view is launched. However, you can change the default view in the *Configuration* dialog box by:

1. Select *View* —> *Configuration*.
2. Select *Views* under the *Flows* folder in the left pane.
3. Select the *Override default view for Tests analysis* check box.
4. Select the desired view from the *Tests Analysis Default View* drop-down.
5. Click *OK*.

This will set the desired view as the default view when you launch the *Tests* page.

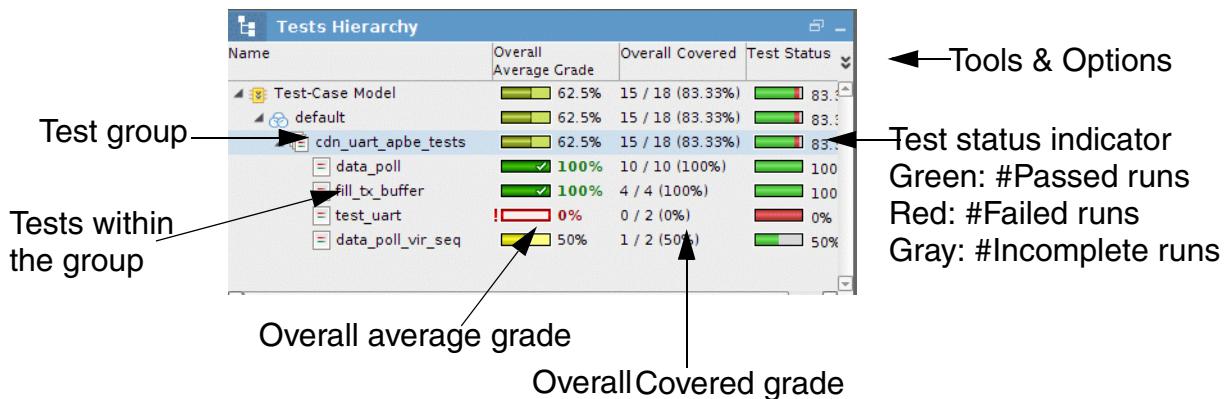
You can now:

- [Navigate Through Various Test Groups](#)
- [Launch Separate Runs Page for Analysis](#)
- [View Runs within the Selected Test or Test Group](#)
- [Launch SimVision Waveform Window for Analysis](#)
- [View Errors, Warnings, and Properties Associated with Selected Item](#)
- [View Attributes and Logs Associated with the Selected Item](#)
- [Generate Report on Runs Page](#)

3.3.1 Navigate Through Various Test Groups

The *Tests Hierarchy* pane shown in [Figure 3-31](#) on page 156 displays the list of test groups in the selected session.

Figure 3-31 Analysis Center—Tests Hierarchy Pane



Using the *Tests Hierarchy* pane, you can navigate through the tests hierarchy. By default, the *Tests Hierarchy* pane displays the tests within different test groups, Overall Average grade, overall covered grade, and Test status of each test. However, you can add more columns or remove columns from the table. You can also detach the pane using the Toggle floating option. You can hide the pane using the Hide option.

You can sort table data, and also search for items that meet specific search criteria.

In the *Test Status* indicator field:

- Green portion represents the number of passed runs (#Passed)
- Red portion represents the number of failed runs (#Failed), and
- Gray portion represents incomplete runs (#Runs - (#Passed + #Failed))
- Grade is the ratio of passed runs out of the total runs (#Passed / #Runs) * 100%

The tooltip for this attribute shows as:

```
Tests Status: <(#Passed / #Runs) * 100 %>
(Passed: <#Passed>, Failed: <#Failed>, Other: <value_of_incomplete_runs>)
```

3.3.2 Launch Separate Runs Page for Analysis

You can launch a separate page for analyzing runs:

- In the *Tests Hierarchy* pane, select the test group or the test whose runs you want to analyze, right-click, and select *Analyze All Runs*.

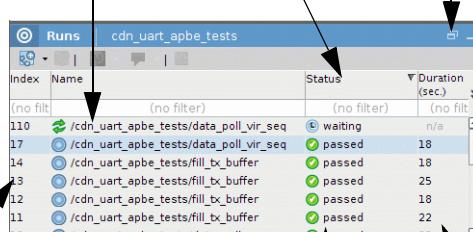
Note: Alternatively, you can select *Analyze All Runs* from the *Analysis* menu.

A separate analysis page is launched for runs analysis.

3.3.3 View Runs within the Selected Test or Test Group

The *Runs* pane displays the runs within the selected test or test group selected in the *Tests Hierarchy* pane. The *Runs* pane is shown in [Figure 3-32](#) on page 157.

Figure 3-32 Analysis Center—Runs Pane



The screenshot shows a table titled "Runs" with the header "cdn_uart_apbe_tests". The columns are: Index, Name, Status, Duration (sec.), and (no filter). The table contains the following data:

Index	Name	Status	Duration (sec.)	(no filter)
110	/cdn_uart_apbe_tests/data_poll_vir_seq	waiting	n/a	(no filter)
17	/cdn_uart_apbe_tests/data_poll_vir_seq	passed	18	
14	/cdn_uart_apbe_tests/fill_tx_buffer	passed	18	
13	/cdn_uart_apbe_tests/fill_tx_buffer	passed	25	
12	/cdn_uart_apbe_tests/fill_tx_buffer	passed	18	
11	/cdn_uart_apbe_tests/fill_tx_buffer	passed	22	
10	/cdn_uart.apbe.tests/data.poll	passed	22	

Annotations with arrows point to various parts of the interface:

- "Runs within selected test or test group" points to the table header.
- "Run identifier (automatically generated by runner)" points to the "Name" column.
- "Run status" points to the "Status" column.
- "Run duration" points to the "Duration (sec.)" column.
- "Tools & Options" points to the dropdown menu icon in the top right corner of the table header.
- "Hide" points to the "Toggle floating" button in the top right corner of the pane.

The *Runs* pane shows all the runs within the selected test or test group. By default, it displays the run identifier, name of run, its status (passed or failed), run duration, top-level file of the test, and the start time of each run. However, you can add more columns or remove columns from the table using the Tools & Options drop-down. You can detach the pane using the Toggle floating option. You can hide the pane using the Hide option.

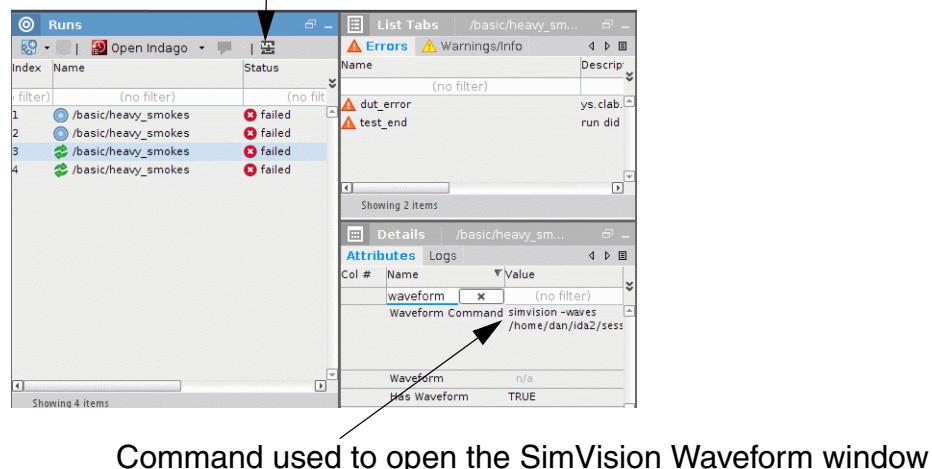
You can also filter data in the table to display only the required items, sort table data, search for items that meet specific search criteria, and also group the runs.

3.3.4 Launch SimVision Waveform Window for Analysis

[Figure 3-33](#) on page 158 displays the *Show Waveform* button. This button shows active for the runs for which waveform is recorded.

Figure 3-33 Show Waveform

Click to launch the waveform for analysis in SimVision



You can open the waveform for analysis by selecting the run and then clicking the *Show Waveform* button in the toolbar.

Note: Alternatively, you can right-click the run and select *Show Waveform* from the popup menu.

When you click the *Show Waveform* option, the SimVision Waveform window is opened with the recorded waveform launched.

3.3.5 View Errors, Warnings, and Properties Associated with Selected Item

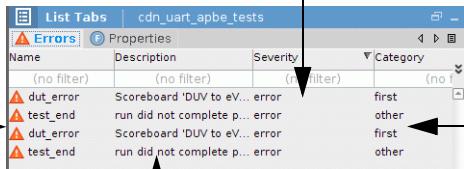
The *List Tabs* pane displays the errors, warnings, and properties of the selected item. The *List Tabs* pane has the following tabs:

- Errors
- Warnings (This shows only when a selection is made in the *Runs* pane)
- Properties

Errors Tab

[Figure 3-34 on page 159](#) displays the *Errors* tab page.

Figure 3-34 Errors Tab Page



The screenshot shows a table titled "cdn_uart_apbe_tests" with the "Errors" tab selected. The table has four columns: Name, Description, Severity, and Category. There are four rows of data:

Name	Description	Severity	Category
dut_error	Scoreboard 'DUV to eV... error	first	other
test_end	run did not complete p... error	other	first
dut_error	Scoreboard 'DUV to eV... error	first	other
test_end	run did not complete p... error	other	other

Annotations with arrows point to specific parts of the table:

- An arrow labeled "Errors generated during session run" points to the first row.
- An arrow labeled "Error severity" points to the "Severity" column header.
- An arrow labeled "Error category" points to the "Category" column header.
- An arrow labeled "Error description" points to the "Description" column header.

The *Errors* tab page shows all the errors associated with the selected test, test group, or run.

By default, the *Errors* tab page displays the error name, description, severity, and category of each error. However, you can add more columns or remove columns from the table, as required.

You can also filter data in the table to display only the required items, sort table data, search for items that meet specific search criteria, and also group the errors.

Warnings Tab

This tab is shown only if a selection is made in the *Runs* pane. [Figure 3-35 on page 159](#) shows the *Warnings* tab page.

Figure 3-35 Warnings Tab Page



The screenshot shows a table titled "cdn_uart_apbe_tests" with the "Warnings" tab selected. The table has three columns: Name, Description, and Tool. There are three rows of data:

Name	Description	Tool
warning	Ignoring \$SPECMAN_HOME. Ac...	specman
warning	WARN_DEFAULT_GENERATOR: specman	specman
warning	Some warnings were discover...	specman

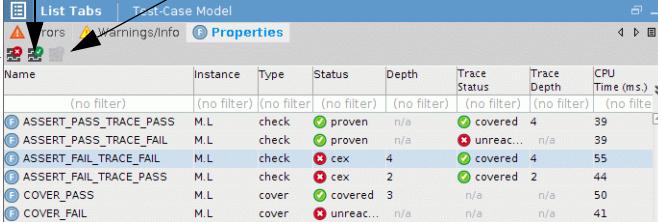
An annotation with an arrow labeled "Warnings" points to the first row.

The *Warnings* tab page shows the warnings associated with the selected run. By default, the *Warnings* tab page displays the name, description, tool that generated the warning. However, you can add more columns or remove columns from the table, as required.

Properties Tab

[Figure 3-36 on page 160](#) shows the *Properties* tab page.

Figure 3-36 Properties Tab Page



Name	Instance	Type	Status	Depth	Trace Status	Trace Depth	CPU Time (ms.)	⋮
(no filter)								
ASSERT_PASS_TRACE_PASS	M.L	check	proven	n/a	covered	4	39	
ASSERT_PASS_TRACE_FAIL	M.L	check	proven	n/a	unreac...	n/a	39	
ASSERT_FAIL_TRACE_FAIL	M.L	check	cex	4	covered	4	55	
ASSERT_FAIL_TRACE_PASS	M.L	check	cex	2	covered	2	44	
COVER_PASS	M.L	cover	covered	3	n/a	n/a	50	
COVER_FAIL	M.L	cover	unreac...	n/a	n/a	n/a	41	

The *Properties* tab page shows the list of formal properties along with details, such as instance, type, status, depth and so on. You can add or remove columns from the table, as required.

If a particular property has trigger waveform, trace waveform, or counter example waveform, then the following corresponding buttons in the table show as active:

- Show Counter example — Launches JasperGold (JG) and opens the waveform of the formal counter example.
- Show trace waveform —Launches JasperGold (JG) and opens the waveform of a formal trace.
- Show trigger waveform —Launches JasperGold (JG) and opens the waveform of a formal trigger.

3.3.6 View Attributes and Logs Associated with the Selected Item

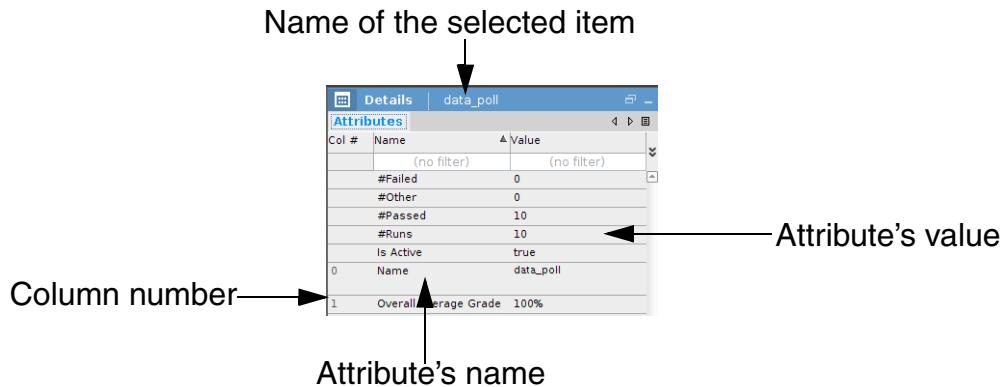
The *Details* pane displays the attributes and logs, and source files of the selected item. The *Details* pane has the following tabs:

- Attributes
- Logs (This shows only when a selection is made in the *Runs* pane)
- Source (This shows only when a selection is made in the *Properties* page of List Tabs pane)

3.3.6.1 Attributes Tab

When you select a test group, test, or a run, the corresponding attributes are shown in the *Attributes* tab page. For example, [Figure 3-37](#) on page 161 shows the attributes of test `data_poll`.

Figure 3-37 Attributes Tab Page



The *Col #* column of the *Attributes* pane shows the column number of the corresponding attribute. No value in the *Col #* column indicates that the corresponding attribute is not displayed in the pane.

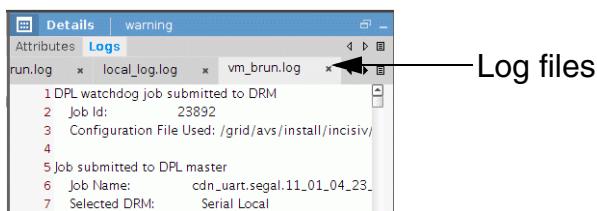
You can right-click on an attribute and select any of the following:

- Add Columns—To add the selected column in the pane of the selected item
 - Remove Columns—To remove the selected column from the pane of the selected item
- Note:** Alternatively, you can double-click an attribute to add it, and then again double-click it to remove it from the pane.
- Unfilter Table—To release an already applied filter
 - Unsort Table—To unsort the table data
 - Copy Cell—To copy the data in the selected cell and paste it in any editor outside of vManager
 - Copy Row—To copy the data in the selected row and paste it in any editor outside of vManager

3.3.6.2 Logs Tab

This tab is shown only if a selection is made in the *Runs* pane. When you click on a run, the corresponding log file is shown in the *Logs* tab page, as shown in [Figure 3-38](#) on page 162.

Figure 3-38 Logs Tab Page

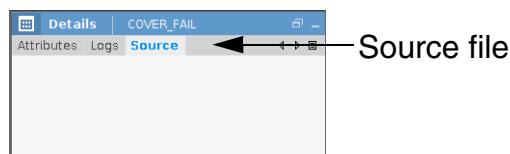


The *Logs* tab page shows multiple log files associated with the selected run. If the log file is not found at the specified location, an error is shown on the *Logs* tab page.

3.3.6.3 Source Tab

This tab is shown only if a selection is made in the *Properties* page of *List Tabs* pane. When you click on a property, the corresponding source file is shown in the *Source* tab page, as shown in [Figure 3-39](#) on page 162.

Figure 3-39 Source Tab Page



If the source file is not found at the specified location, nothing is shown in the *Source* tab page.

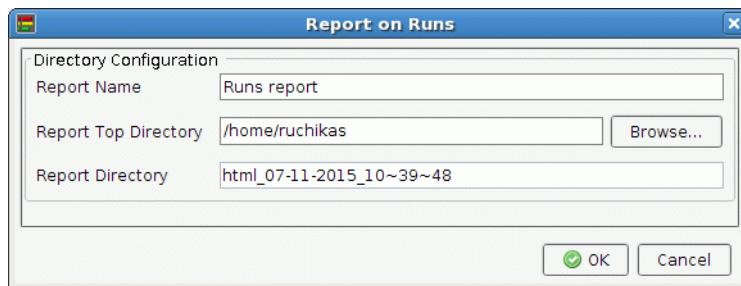
3.3.7 Generate Report on Runs Page

To generate report on *Runs* page:

1. In the *Runs* analysis page, click the *Report* button in the Report toolbar.

The *Report on Runs* dialog box is displayed, as shown in [Figure 3-40](#) on page 163.

Figure 3-40 Report on Runs



2. Specify the name of the report in the *Report Name* text box. This name will appear on the HTML report.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 3-41](#) on page 163.

Figure 3-41 Report Done



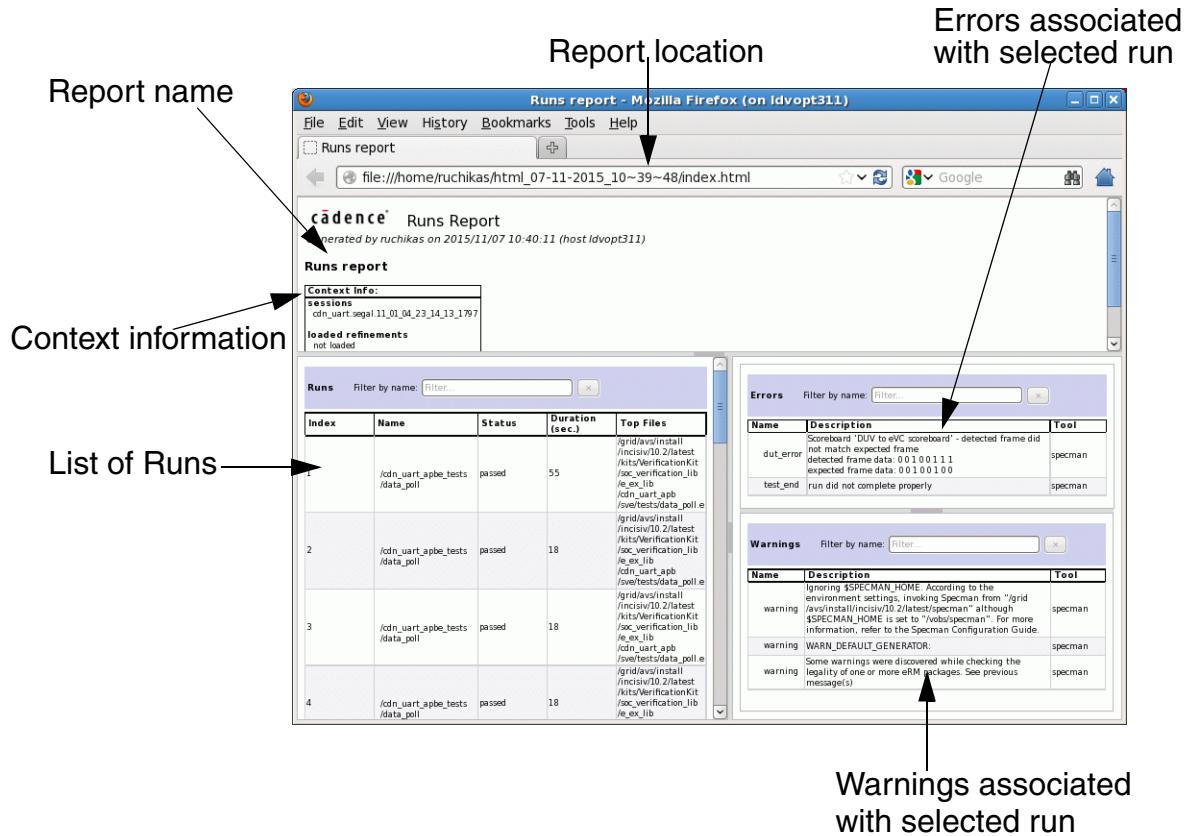
This dialog box shows the location where the report is generated.

6. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 3-42](#) on page 164 shows the generated report.

Figure 3-42 Cover group Report



The report shows the information such as, who generated the report and when the report was generated. It also includes the context information, such as session and loaded refinement file.

The report lists the runs information of the selected session or test. It shows following tables:

- **Runs**—This table lists the runs within the selected session or tree.
- **Errors**—This table lists the errors associated with the selected run.
- **Warnings**—This table lists the warnings associated with the selected run.

The columns shown in the report, the sort order, and data shown in the report (filters) is based on the current state of the table or the view selected at the time of generating the report. If required, you can adjust the columns that you want to include or remove from the table, apply sort order, and also apply filters before generating the report.

Note: After you have generated the report, you can filter data based on the *Name* field using the *Filter by name* text box provided at the top of each table in the report.

3.4 Analyzing Failures

The *Analysis* center of the vManager also allows you to analyze failures (failed messages).

You can launch a separate page for analyzing failures using any of the following methods:

- In the *Tests Hierarchy* pane, select the test group or the test whose failures you want to analyze, right-click, and select *Analyze Failures*.

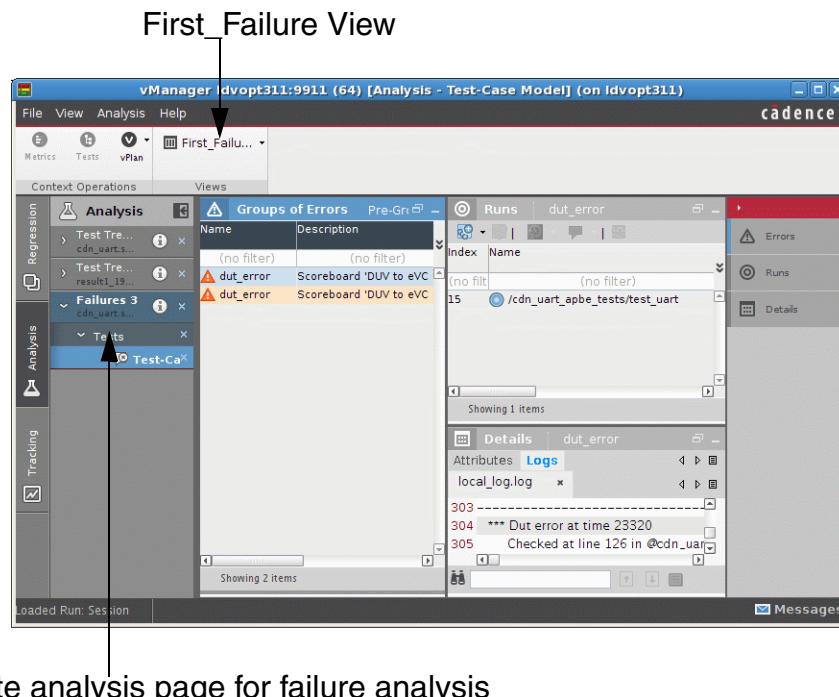
Note: Alternatively, you can select *Analyze Failures* from the *Analysis* menu.

- In the *Regression* center, select the session whose failures you want to analyze, right-click, and select *Analyze Failures*.

Note: You can also launch a separate page for analyzing failures from the *Regression* center by clicking the *Failures* button in the *Analyze* toolbar.

A separate analysis page is launched for failure analysis, as shown in [Figure 3-43](#) on page 165.

Figure 3-43 Separate Failure Analysis Page Launched



By default, when you open failure analysis, the *First_Failure* view is launched. However, you can change the default view in the *Configuration* dialog box by:

1. Select *View* —> *Configuration*.
2. Select *Views* under the *Flows* folder in the left pane.
3. Select the *Override default view for Failure analysis* check box.
4. Select the desired view from the *Failure Analysis Default View* drop-down.
5. Click *OK*.

3.4.1 Available Views for Failure Analysis

The following pre-defined views are available for failure analysis:

- [First Failure View](#)
- [All Failures View](#)
- [Indago Debug Analyzer View](#)

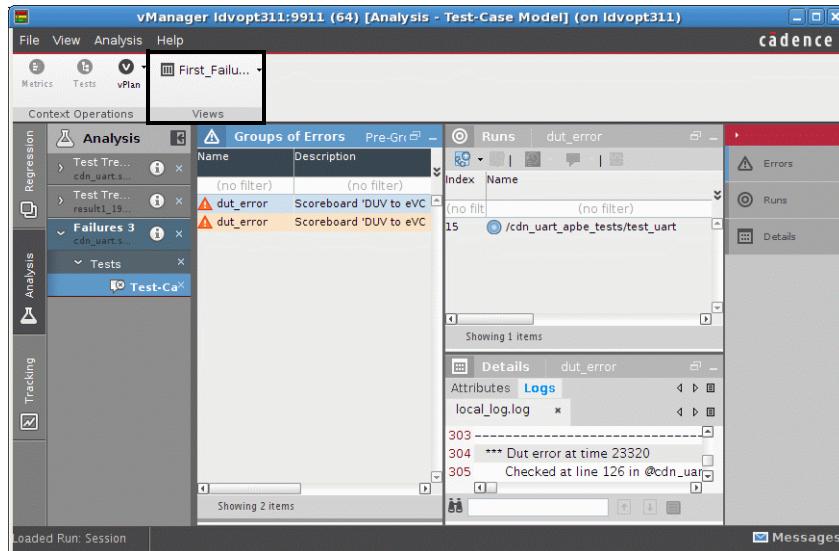
3.4.1.1 First Failure View

The *First_Failure* view groups the errors by their description and name, and also filters the errors where category = first.

The *First_Failure* view helps you easily determine the runs that have failed prematurely because of a missing file or a license timeout.

[Figure 3-44](#) on page 167 shows the *First_Failure* view.

Figure 3-44 Analysis Center (First_Failure View)



The *First_Failure* view shows the following panes:

- Groups of Errors
- Runs
- Details

Groups of Errors

Figure 3-45 on page 167 shows the *Groups of Errors* pane.

Figure 3-45 First_Failure View (Groups of Errors)

The Groups of Errors pane displays a table with columns: Name, Description, Context, Severity, Category, and Number Of Entities. Two rows are shown, both for "dut_error" with severity "error" and category "first". The "Number Of Entities" column shows values 1 and 1 respectively.

Name	Description	Context	Severity	Category	Number Of Entities
dut_error	Scoreboard 'DUV to ... cdn_uart..	error	error	first	1
dut_error	Scoreboard 'DUV to ... cdn_uart..	error	error	first	1

The *Groups of Errors* pane of the *First_Failure* view shows the list of groups created after specifying the following filter:

- Severity of the error should be `error` or `critical`, and
- Category should be `first`

For each listed group, the group name, short description, context, severity, category, and number of errors in the group are shown. The errors in the selected group are shown right below the *Groups of Errors* pane. For each error, a small description, context and severity is shown.

Runs

[Figure 3-46](#) on page 168 shows the *Runs* pane.

Figure 3-46 First_Failure View (Runs)

The screenshot shows a table titled "Runs" with one row. The columns are "Index", "Name", and "Status". The "Index" column has value "16". The "Name" column has value "/cdn_uart_apbe_tests/test_uart...". The "Status" column has value "failed" with a red error icon. The table has a header row with column names and a footer row "(no filter)" for each column.

Index	Name	Status
(no filter)	(no filter)	(no filter)
16	/cdn_uart_apbe_tests/test_uart...	failed

The *Runs* pane shows the runs associated with the selected group or the selected message. For more details, see [View Runs within the Selected Test or Test Group](#) on page 157.

Details

[Figure 3-47](#) on page 168 shows the *Details* pane.

Figure 3-47 First_Failure View (Details)

The screenshot shows a table titled "Details" with five rows. The columns are "Col #", "Name", and "Value". The "Col #" column has values "#Failed", "#Failures", "#Passed", and "#Runs". The "Name" column has values "1", "5", "0", and "1". The "Value" column has values "(no filter)", "(no filter)", "(no filter)", and "(no filter)". The table has a header row with column names and a footer row "(no filter)" for each column.

Col #	Name	Value
(no filter)	(no filter)	(no filter)
#Failed	1	(no filter)
#Failures	5	(no filter)
#Passed	0	(no filter)
#Runs	1	(no filter)

The *Details* pane shows the attributes and logs associated with the selected group.

For more details, see [View Attributes and Logs Associated with the Selected Item](#) on page 160.

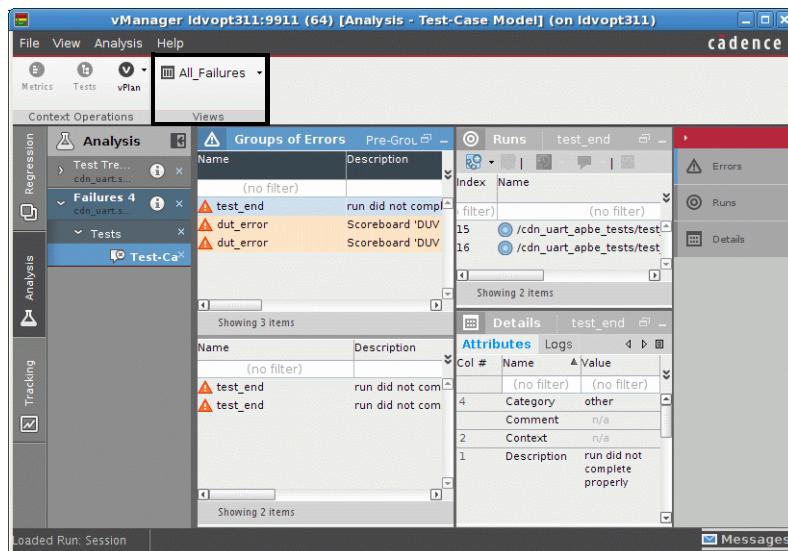
3.4.1.2 All_Failures View

The *All_Failures* view displays all the failures that occurred in each run. This view groups failures by the kind of failure and the failure description. It also filters out informational messages from the list of failures for each run.

Note: The *All_Failures* view facilitates the analysis of checks, where multiple assertion failures might occur during a single run.

[Figure 3-48 on page 169](#) shows the *All_Failures* view.

Figure 3-48 Analysis Center (All_Failures View)



The *Groups of Errors* pane of the *All_Failure* view shows the list of groups created after specifying the following filter:

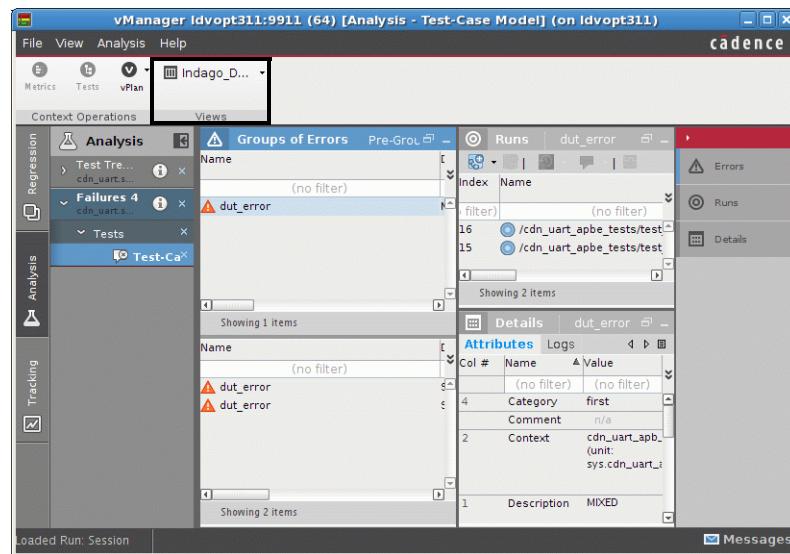
- Severity of the error should be `error` or `critical`.

For the details on different panes in this view, see [First Failure View on page 166](#).

3.4.1.3 Indago_Debug_Analyzer View

[Figure 3-49 on page 170](#) shows the *Indago_Debug_Analyzer* view.

Figure 3-49 Analysis Center (Indago_Debug_Analyzer View)



Indago Debug Analyzer is a post-process, graphical debugger. You can use vManager to automatically rerun failing tests and gather debugging data as part of regression testing.

For details on Debug Analyzer's integration with vManager, see [Integrating vManager with Indago Debug Analyzer](#) on page 209.

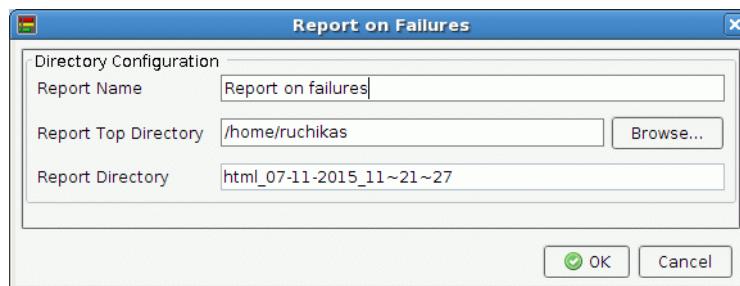
3.4.2 Generate Report on Failure Page

To generate report on *Failure* page:

1. In the *Failures* analysis page, click the *Report* button in the Report toolbar.

The *Report on Failures* dialog box is displayed, as shown in [Figure 3-50](#) on page 170.

Figure 3-50 Report on Failures



2. Specify the name of the report in the *Report Name* text box. This name will appear on the HTML report.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 3-51](#) on page 171.

Figure 3-51 Report Done



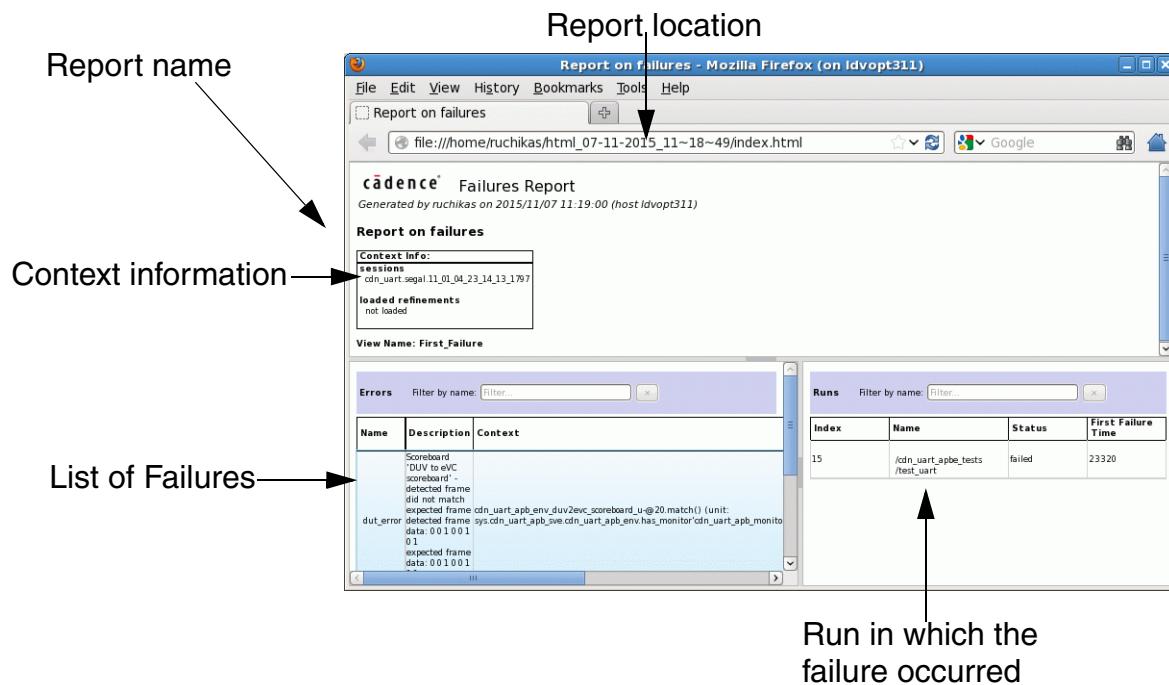
This dialog box shows the location where the report is generated.

6. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 3-52](#) on page 172 shows the generated report.

Figure 3-52 Failures Report



The report shows the information such as, who generated the report and when the report was generated. It also includes the context information, such as session and loaded refinement file.

The report lists the failure information of the selected session or test. It shows following tables:

- Errors—This table lists the failures within the selected session or tree.
- Runs—This table lists the runs in which the selected failure occurred.

The columns shown in the report, the sort order, and data shown in the report (filters) is based on the current state of the table or the view selected at the time of generating the report. If required, you can adjust the columns that you want to include or remove from the table, apply sort order, and also apply filters before generating the report.

After you have generated the report, you can filter data based on the *Name* field using the *Filter by name* text box provided at the top of each table in the report.

3.5 Analyzing Formal Properties

The *Analysis* center of the vManager allows you to analyze formal properties.

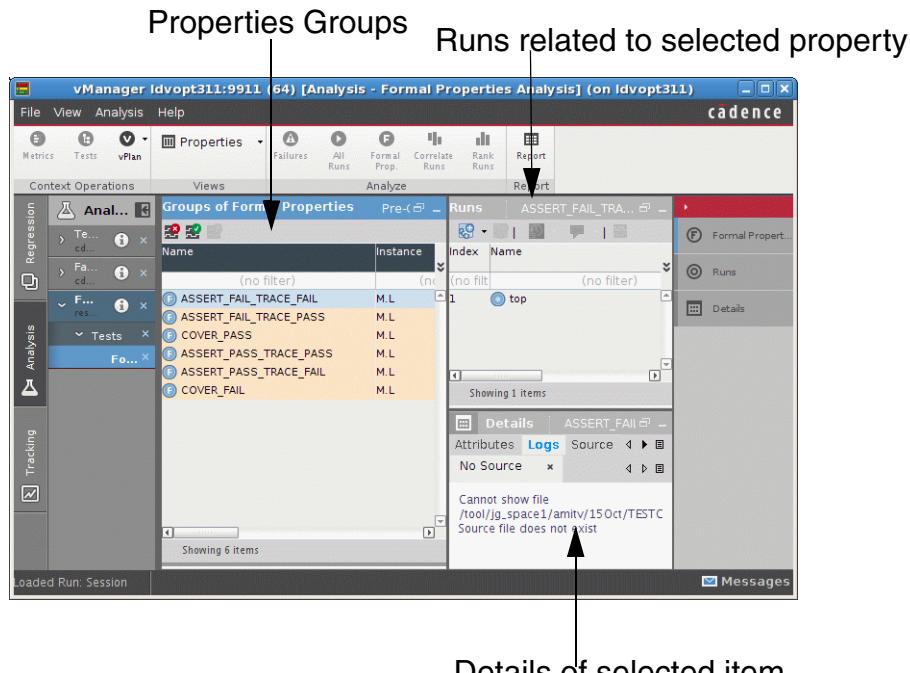
3.5.1 Launching the Formal Properties Analysis Page

To launch a *Formal Properties Analysis* page, do any of the following:

- If you are in *Regression* center, select the session that has the formal properties, right-click, and then select *Analyze Formal Properties*.
- If you are already in the *Analysis* center on the tests page, select a test, right-click, and then select *Analyze Formal Properties*.

The *Formal Properties Analysis* page is launched, as shown in [Figure 3-53](#) on page 173.

Figure 3-53 Formal Properties Analysis Page Launched



The *Formal Properties Analysis* page is based on all properties from the selected session/runs. This page is divided into different panes, and shows following details:

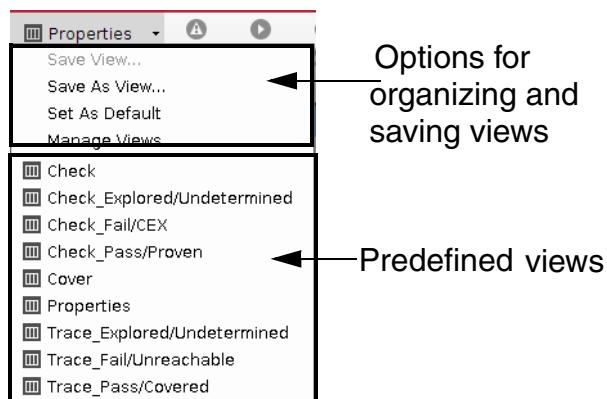
- The *Group of Properties* pane shows the various groups under which the various properties are classified. In this pane by default, the grouping is done on attributes: Name, Instance, Type, and Status; and sorting is done on attributes: Name, Instance, and Status.
- The *Runs* pane shows the source or the parent run related to the selected formal property or property group.

- The *Details* pane shows the details (attributes, logs, source reference) of the selected item.

3.5.2 Properties Views

The *Formal Properties Analysis* page provides with the following views, as shown in [Figure 3-54](#) on page 174.

Figure 3-54 Views Drop-Down



You can select any of the following predefined views:

- Properties—To show *Name*, *Instance*, *Type*, *Status*, *Depth*, *Trace Status*, *Trace Depth*, *CPU Time*, and *Number of Entities* attributes in the *Groups of Formal Properties* pane, as shown in [Figure 3-55](#) on page 174.

Figure 3-55 Properties View

The screenshot shows a table titled 'Groups of Formal Properties' with a 'Pre-Grouping Filter: No filter' header. The table has columns: Name, Instance, Type, Status, Depth, Trace Status, Trace Depth, CPU Time (ms.), and Number Of Entities. The rows list various formal properties: ASSERT_FAIL_TRACE_FAIL, ASSERT_FAIL_TRACE_PASS, COVER_PASS, ASSERT_PASS_TRACE_PASS, ASSERT_PASS_TRACE_FAIL, and COVER_FAIL. An annotation points to the table with the text 'Shows formal properties attributes'.

Name	Instance	Type	Status	Depth	Trace Status	Trace Depth	CPU Time (ms.)	Number Of Entities
(no filter)	(no filter)	(no filter)	(no filter)	(no fil	(no fil	(no fil	(no fil	(no fil
ASSERT_FAIL_TRACE_FAIL	M.L	check	● cex	4	● covered	4	55	1
ASSERT_FAIL_TRACE_PASS	M.L	check	● cex	2	● covered	2	44	1
COVER_PASS	M.L	cover	● covered	3	n/a	n/a	50	1
ASSERT_PASS_TRACE_PASS	M.L	check	● proven	n/a	● covered	4	39	1
ASSERT_PASS_TRACE_FAIL	M.L	check	● proven	n/a	● unreacha...	n/a	39	1
COVER_FAIL	M.L	cover	● unreachable	n/a	n/a	n/a	41	1

- Check—This view shows all the attributes as shown in the *Properties* view with additional predefined filter on attribute *Type*. This view filters the table data to show only the items for which *Type* is *check*, as shown in [Figure 3-56](#) on page 175.

Figure 3-56 Check View

Name	Instance	Type	Status	Depth	Trace Status	Trace Depth	CPU Time (ms)	No.
(no filter)	(no filter)	[check]	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	
ASSERT_FAIL_TRACE_FAIL	M.L	check	cex	4	covered	4	55	1
ASSERT_FAIL_TRACE_PASS	M.L	check	cex	2	covered	2	44	1
ASSERT_PASS_TRACE_PASS	M.L	check	proven	n/a	covered	4	39	1
ASSERT_PASS_TRACE_FAIL	M.L	check	proven	n/a	unresolved	n/a	39	1

Shows only the items for which *Type* is check

- **Check_Explored/Undetermined**—This view is same as the *Check* view with an additional predefined filter on attribute *Status*. This view filters the table data to show only the items for which *Type* is check and *Status* is undetermined or explored, as shown in [Figure 3-57](#) on page 175.

Figure 3-57 Check_Explored/Undetermined View

Name	Instance	Type	Status	Depth	Trace Status
(no filter)	(no filter)	[check]	(explored, undetermined)	(no filter)	(no filter)
ASSERT_FAIL_TRACE_PASS	M.L	check	[explored, undetermined]	2	(no filter)

Shows only the items for which *Type* is check

Status is undetermined or explored

- **Check_Fail/CEX**—This view is same as the *Check* view with an additional predefined filter on attribute *Status*. This view filters the table data to show only the items for which *Type* is check and *Status* is fail or cex, as shown in [Figure 3-58](#) on page 175.

Figure 3-58 Check_Fail/CEX View

Name	Instance	Type	Status	Depth	Trace Status
(no filter)	(no filter)	[check]	(fail, cex)	(no filter)	(no filter)
ASSERT_FAIL_TRACE_PASS	M.L	check	cex	2	covered

Shows only the items for which *Type* is check

Status is fail or CEX

- **Check_Pass/Proven**—This view is same as the *Check* view with an additional predefined filter on attribute *Status*. This view filters the table data to show only the items for which *Type* is check and *Status* is pass or proven, as shown in [Figure 3-59](#) on page 176.

Figure 3-59 Check_Pass/Proven View

Name	Instance	Type	Status	Depth
(no filter)	(no filter)	[check]	✗ [pass, proven]	✗ (no filter)
ASSERT_PASS_TRACE_PASS	M.L	check	✓ proven	n/a
ASSERT_PASS_TRACE_FAIL	M.L	check	✓ proven	n/a

Shows only the items for which *Type* is check

Status is pass or proven

- Cover—This view shows all the attributes as shown in the *Properties* view with additional predefined filter on attribute *Type*. This view filters the table data to show only the items for which *Type* is *cover*, as shown in [Figure 3-60](#) on page 176.

Figure 3-60 Cover View

Name	Instance	Type	Status	Depth	Trace Status
(no filter)	(no filter)	[cover]	✗ (no filter)	(no filter)	(no filter)
COVER_PASS	M.L	cover	✓ covered 3	n/a	n/a
COVER_FAIL	M.L	cover	✗ unreach...	n/a	n/a

Shows only the items for which *Type* is cover

- Trace_Explored/Undetermined—This view shows all the attributes as shown in the *Properties* view with additional predefined filter on attribute *Trace Status*. This view filters the table data to show only the items for which *Trace Status* is explored or undetermined, as shown in [Figure 3-61](#) on page 176.

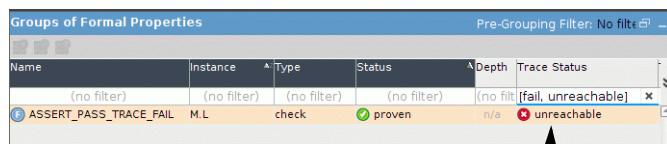
Figure 3-61 Trace_Explored/Undetermined View

Name	Instance	Type	Status	Depth	Trace Status	Tr...
(no filter)	[Explored, undetermined]	✗				

Shows only the items for which *Trace Status* is explored or undetermined

- Trace_Fail/Unreachable—This view shows all the attributes as shown in the *Properties* view with additional predefined filter on attribute *Trace Status*. This view filters the table data to show only the items for which *Trace Status* is fail or unreachable, as shown in [Figure 3-62](#) on page 177.

Figure 3-62 Trace_Fail/Unreachable View



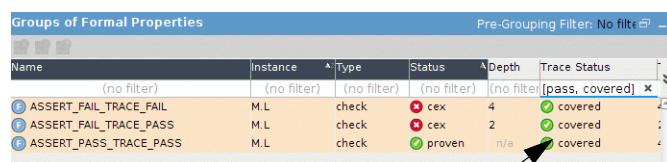
The screenshot shows a table titled 'Groups of Formal Properties'. The columns are: Name, Instance, Type, Status, Depth, and Trace Status. A 'Pre-Grouping Filter' dropdown at the top right is set to 'No filter'. An arrow points to the 'Trace Status' column header, which has a dropdown menu open showing '(no filter)' and '[fail, unreachable]'. The row for 'ASSERT_PASS_TRACE_FAIL' is highlighted in orange, showing 'unreachable' in the Trace Status column.

Name	Instance	Type	Status	Depth	Trace Status
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
ASSERT_PASS_TRACE_FAIL	M.L	check	proven	n/a	unreachable

Shows only the items for which *Trace Status* is fail or unreachable

- **Trace_Pass/Covered**—This view shows all the attributes as shown in the *Properties* view with additional predefined filter on attribute *Trace Status*. This view filters the table data to show only the items for which *Trace Status* is pass or covered, as shown in [Figure 3-63 on page 177](#).

Figure 3-63 Trace_Pass/Covered View



The screenshot shows a table titled 'Groups of Formal Properties'. The columns are: Name, Instance, Type, Status, Depth, and Trace Status. A 'Pre-Grouping Filter' dropdown at the top right is set to 'No filter'. An arrow points to the 'Trace Status' column header, which has a dropdown menu open showing '(no filter)' and '(pass, covered)'. The rows for 'ASSERT_FAIL_TRACE_FAIL' and 'ASSERT_FAIL_TRACE_PASS' are highlighted in orange, showing 'covered' in the Trace Status column.

Name	Instance	Type	Status	Depth	Trace Status
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
ASSERT_FAIL_TRACE_FAIL	M.L	check	x cex	4	covered
ASSERT_FAIL_TRACE_PASS	M.L	check	x cex	2	covered
ASSERT_PASS_TRACE_PASS	M.L	check	proven	n/a	covered

Shows only the items for which *Trace Status* is pass or covered

You can set any of the views, as required, by selecting it from the *Views* drop-down in the toolbar.

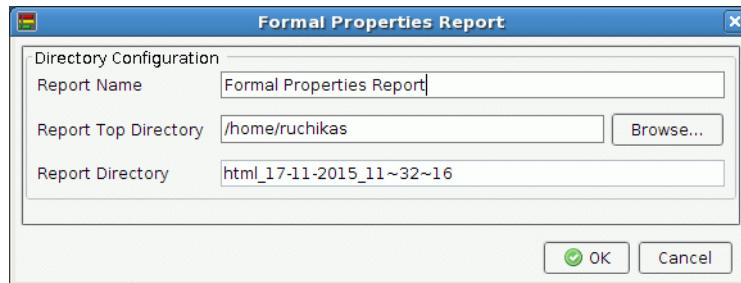
3.5.3 Report on Formal Properties Page

To generate report on *Formal Properties* page:

1. In the *Formal Properties* analysis page, click the *Report* button in the Report toolbar.

The *Formal Properties Report* dialog box is displayed, as shown in [Figure 3-64 on page 178](#).

Figure 3-64 Report on Formal Properties



2. Specify the name of the report in the *Report Name* text box. This name will appear on the HTML report.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 3-65](#) on page 178.

Figure 3-65 Report Done



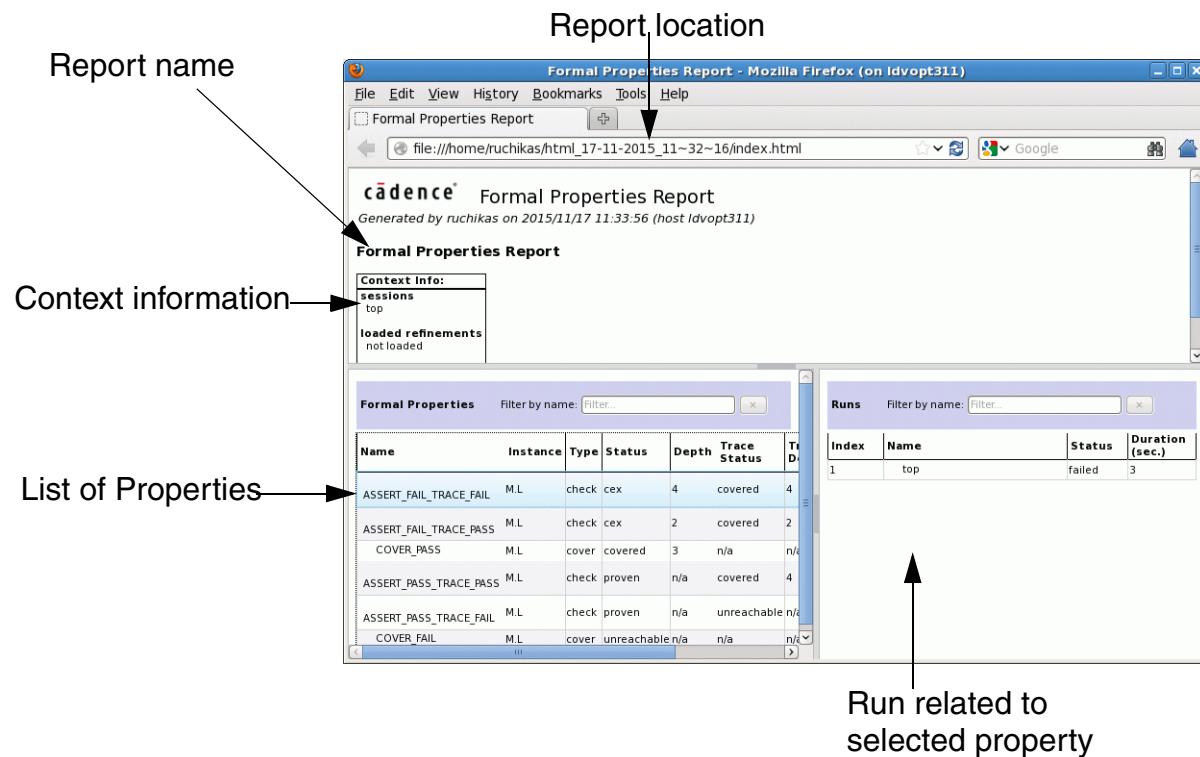
This dialog box shows the location where the report is generated.

6. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 3-66](#) on page 179 shows the generated report.

Figure 3-66 Formal Properties Report



The report shows the information such as, who generated the report and when the report was generated. It also includes the context information, such as session and loaded refinement file.

The report lists the formal properties information. It shows following tables:

- **Formal Properties**—This table lists the formal properties within the selected session.
- **Runs**—This table shows the parent run related to the selected formal property.

The columns shown in the report, the sort order, and data shown in the report (filters) is based on the current state of the table or the view selected at the time of generating the report. If required, you can adjust the columns that you want to include or remove from the table, apply sort order, and also apply filters before generating the report.

After you have generated the report, you can filter data based on the *Name* field using the *Filter by name* text box provided at the top of each table in the report.

3.6 Grouping Runs

To find runs with common attributes, such as all runs with the same status or all runs with the same owner, you apply grouping rules to a set of runs based on those attributes. The result of grouping is a table, where each row shows all runs that share the same values for the specified attributes.

After creating groups, you can save grouping rules with a customized view.

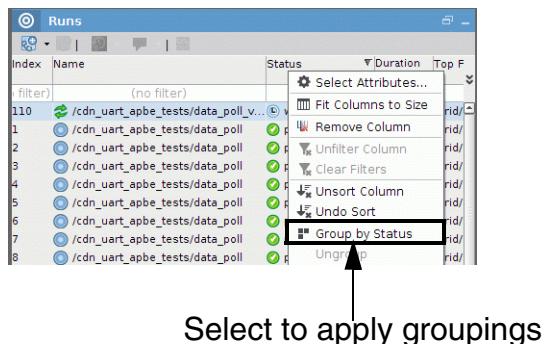
You can apply groupings based on one or more attributes.

3.6.1 Grouping by One Attribute

To apply groupings:

1. Right-click the column header. For example, to group based on run status, right-click the *Status* column, as shown in [Figure 3-67](#) on page 180.

Figure 3-67 Runs Pane—Grouping Runs



2. Select *Group by Status* from the pop-up menu.

Note: If you right-clicked the *Name* column, then the pop-up menu would have shown *Group by Name* instead of *Group by Status*.

After you select *Group by Status*, grouping is applied. [Figure 3-68](#) on page 181 shows the applied grouping.

Figure 3-68 Runs Pane—Grouping Applied

The screenshot shows the 'Groups of Runs' pane in the vManager GUI. The 'Status' column is highlighted, indicating it is the grouping attribute. An additional column 'Number Of Entities' is present. Two tables are shown: one for groups and one for individual entities within selected groups.

Index	Name	Status	Duration (sec)	Number Of Entities
(no filter)	/cdn_uart_apbe_tests/data_poll_...	(no filter)	(no filter)	(no filter)
110	/cdn_uart_apbe_tests/data_poll_...	waiting	n/a	1
MIXED	MIXED	passed	336	15
MIXED	/cdn_uart_apbe_tests/test_uart	failed	36	2

Name	Status	Duration (sec.)	See
(no filter)	(no filter)	(no filter)	(no filter)
/cdn_uart_apbe_tests/test_uart	failed	19	120
/cdn_uart_apbe_tests/test_uart	failed	17	735

Annotations:

- 'Groups created' points to the first table.
- 'Entities in selected group' points to the second table.
- 'Attribute on which the grouping is applied' points to the 'Status' column header.
- 'Additional column created' points to the 'Number Of Entities' column.

When you apply groupings:

- ❑ The column on which the grouping is based is highlighted. For example, in this case, the *Status* column shows highlighted.
- ❑ An additional column *Number of Entities* is created. This column indicates the number of entities in each group.
- ❑ An additional table is created, which lists the individual entities in each group. You can select the group in the above table and the entities within that group are shown in the additional table created.

Important

When a table is grouped, some of the attribute's values might show as `MIXED`. This means that there are various values for this attribute in the group entities. In the vManager GUI, the text is shown as `MIXED`. However, in the case of vAPI, instead of `MIXED`, there is a special value for each of the type, as stated below:

- ❑ **String:** `zzzzzzzzzzzzzzzz`
- ❑ **Boolean:** `null`
- ❑ **Long:** `MAX_LONG - 0x7fffffffffffff`
- ❑ **Integer:** `MAX_INT - 0x7fffffff`
- ❑ **Double:** `MAX_DOUBLE - 0x1.ffffffffffffP+1023`
- ❑ **Date:** `January 1st 1970`
- ❑ **ENUM:** `MIXED_GROUP_VALUE`

Note: Alternatively, you can apply groupings using the *Group by Attributes* dialog box. This dialog box is shown when you select *Group by Attributes* from the *Tools & Options* button. For more details, see [Grouping by Multiple Attributes](#) on page 182.

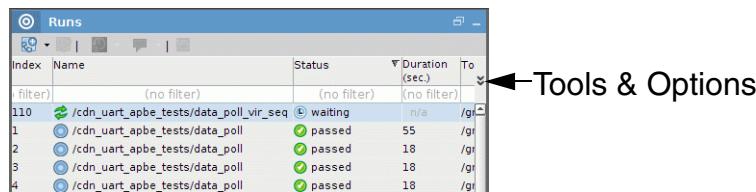
3.6.2 Grouping by Multiple Attributes

You can accumulate grouping actions one on top of the other. In that case, the grouping value is a logical AND of the previous attributes and the new selected attribute. When you group by multiple attributes, each row in the table represents a group of runs that have the same value for each of the grouping attributes.

To group by multiple attributes:

1. Click the *Tools & Options* drop-down in the *Runs* pane, as shown in [Figure 3-69](#) on page 182.

Figure 3-69 Runs Tab Page—Grouping Runs

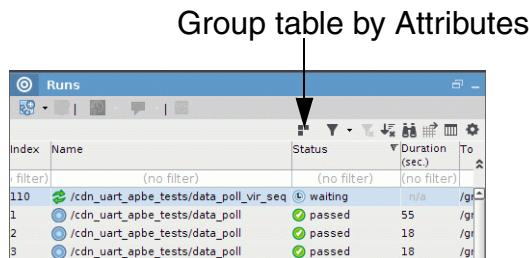


A screenshot of the Incisive vManager interface showing the 'Runs' tab. At the top right of the table header, there is a 'Tools & Options' dropdown menu. The table itself contains five rows of test run data, with columns for Index, Name, Status, Duration (sec.), and To. The first row is highlighted in blue.

Index	Name	Status	Duration (sec.)	To
110	/cdn_uart_apbe_tests/data_poll_vir_seq	waiting	n/a	/gi
1	/cdn_uart_apbe_tests/data_poll	passed	55	/gi
2	/cdn_uart_apbe_tests/data_poll	passed	18	/gi
3	/cdn_uart_apbe_tests/data_poll	passed	18	/gi
4	/cdn_uart_apbe_tests/data_poll	passed	18	/gi

2. Select *Group table by Attributes* icon, as shown in [Figure 3-70](#) on page 182.

Figure 3-70 Runs Tab Page—Grouping Runs



The *Group by Attributes* dialog box is displayed, as shown in [Figure 3-71](#) on page 183.

Figure 3-71 Group by Attributes



3. Select the attributes for applying the groupings and click *OK*. For example, select *Name* and *Status* check boxes and click *OK*.

Figure 3-72 on page 183 shows the applied grouping.

Figure 3-72 Runs Pane—Grouping Applied

The figure shows the 'Groups of Runs' pane with the following data:

Index	Name	Status	Number Of Entities	Duration (sec)	See
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
110	/cdn_uart_apbe_tests/data_poll_vir_seq	waiting	1	n/a	
17	/cdn_uart_apbe_tests/data_poll_vir_seq	passed	1	18	
MIXED	/cdn_uart_apbe_tests/fill_tx_buffer	passed	4	83	
MIXED	/cdn_uart_apbe_tests/data_poll	passed	10	235	
MIXED	/cdn_uart_apbe_tests/test_uart	failed	2	36	

Below the main table, a smaller table shows the entities in the selected group:

Name	Status	Duration (sec)	See
(no filter)	(no filter)	(no filter)	(no filter)
/cdn_uart_apbe_tests/data_poll_vir_seq	waiting	n/a	169

Annotations in the figure point to specific parts of the interface:

- 'Groups created' points to the main table header.
- 'Attributes on which grouping is applied' points to the 'Name' and 'Status' column headers.
- 'Number of entities in each group' points to the 'Number Of Entities' column header.
- 'Entities in the selected group' points to the smaller table below.

In the above figure notice that:

- ❑ Columns *Name* and *Status* are highlighted. This indicates that grouping is based on these attributes.
- ❑ An additional column *Number of Entities* is created. This column indicates the number of entities in each group.
- ❑ An additional table is created, which lists the individual entities in the selected group. You can select the group in the above table and the entities within that group are shown in the additional table created.

Note: You can also apply groupings based on multiple attributes by right-clicking the column header and selecting *Group by* from the popup menu. You can repeat the steps for each attribute you want to group by.

3.6.3 Ungrouping Runs

You can ungroup the runs by any of the following methods:

- Right-click the column header on which the grouping is applied and select *Ungroup* from the popup menu.
- Click *Tools & Options*, select *Group Table by Attributes*, deselect the check boxes from which you want to remove groupings, and click *OK*.

3.7 Filtering Runs

To facilitate analysis, you can drop the runs that you are not interested in from the display.

The row below the column header is the filter row. You can add the filtering criteria in the row and apply the required filter.

[Figure 3-73 on page 184](#) shows the *Runs* pane with the filtering row.

Figure 3-73 Filtering Runs

A screenshot of the 'Runs' pane in vManager. The pane has a header with columns: Index, Name, Status, Duration (sec.), To, and a dropdown arrow. Below the header is a row labeled '(no filter)' with three small icons: a magnifying glass, a gear, and a question mark. This row is highlighted with a light blue background. The main body of the table contains five rows of data, each with a small icon, a name like '/cdn_uart_apbe_tests/data_poll_vir_seq', a status (e.g., 'waiting', 'passed'), a duration (e.g., 'n/a', '55'), and a timestamp ('/g'). An arrow points from the text 'Add filtering criteria' to the '(no filter)' row.

Index	Name	Status	Duration (sec.)	To	
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	
110	/cdn_uart_apbe_tests/data_poll_vir_seq	waiting	n/a	/g	
1	/cdn_uart_apbe_tests/data_poll	passed	55	/g	
2	/cdn_uart_apbe_tests/data_poll	passed	18	/g	
3	/cdn_uart_apbe_tests/data_poll	passed	18	/g	
4	/cdn_uart_apbe_tests/data_poll	passed	18	/g	

You can specify the filter criteria in the filter row. When you place the cursor over the filtering row, a tooltip is shown. It shows information related to specifying the filtering criteria for that particular field.

For more details on filtering, see [Filtering Table Data on page 33](#).

3.8 Ranking Runs

You can improve the coverage efficiency of a regression suite by dropping redundant runs—runs that do not add significantly to the coverage grade. When you perform ranking, vManager re-orders the runs and opens a separate analysis page to show the ranking results.

This section covers following topics:

- [Rank runs](#)

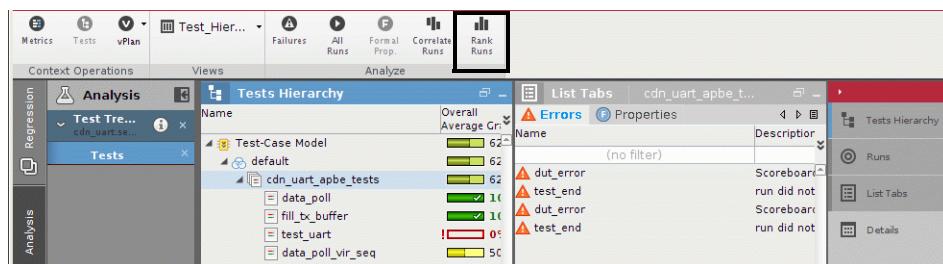
- [Report on Rank Page](#)
- [Enable Parallel Ranking](#)

3.8.1 Rank runs

To rank runs:

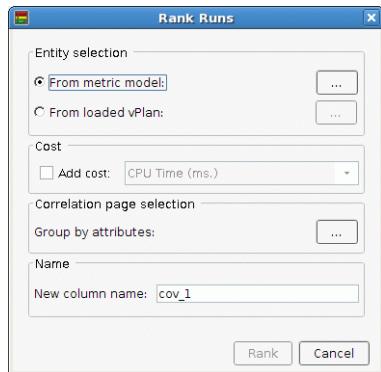
1. Select the node for which you want to rank runs (if you are in the *Tests Hierarchy* pane).
- Note:** Alternatively, you can also select the runs (you want to rank) from the *Runs* table.
- Note:** In case you are in the *Metrics* page or the *vPlan* page, you can do any of the following:
- Select single or multiple items (types, instances, bins, toggles, and so on) in the hierarchy tree, right-click and select *Rank Runs*.
 - Select multiple items in the *List tabs* pane in the *Metrics* page, right-click and select *Rank Runs*.
2. Select *Rank Runs* from the *Analyze* toolbar, as shown in [Figure 3-74](#) on page 185.

Figure 3-74 Ranking Runs



The *Rank Runs* dialog box appears, as shown in [Figure 3-75](#) on page 186.

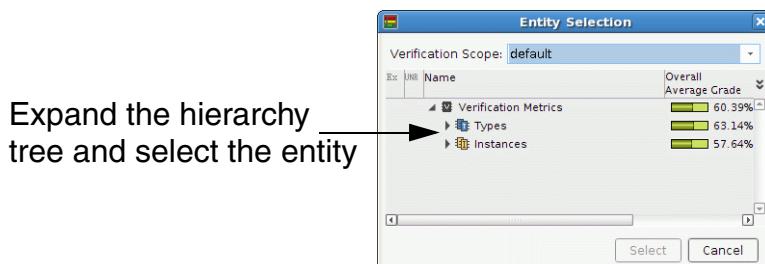
Figure 3-75 Rank Runs



3. You can rank runs based on their contribution to coverage grade or the loaded vPlan. To rank based on metric model, select the *From metric model* radio button, and click the button next to it.

The *Entity Selection* dialog box appears, as shown in [Figure 3-76](#) on page 186.

Figure 3-76 Entity Selection

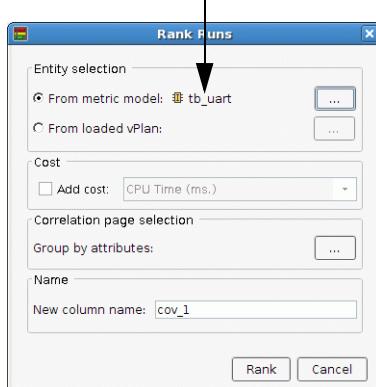


4. Expand the hierarchy tree and select the required entity. For example, select *tb_uart* from the listed instances.
5. Click *Select*.

The *Rank Runs* dialog box appears, as shown in [Figure 3-77](#) on page 187.

Figure 3-77 Rank Runs

Indicates your selection (This will show as the page name in the navigation pane)



- By default, cost is not considered while ranking the runs. To consider the cost, select the *Add cost* check box. This will activate the cost drop-down list. It has following options:
 - CPU Time (ms.) — to consider CPU time while ranking runs.
 - Simulation Time — To consider simulation time while ranking runs.
 - User Time (ms.) — To consider user time while ranking runs.
 - System Time (ms.) — To consider system time while ranking runs.

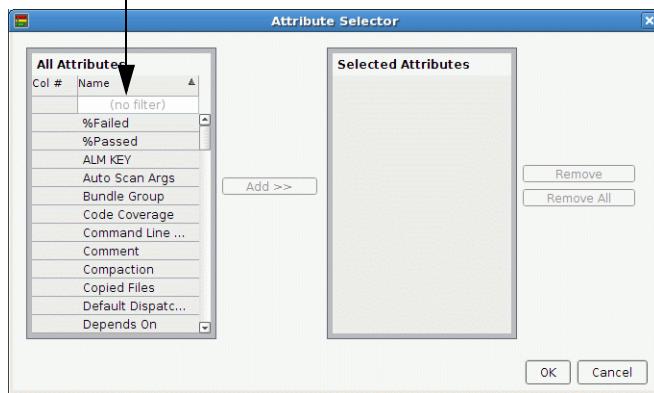
For example, to consider simulation time, select *Simulation Time* from the drop-down list.

Note: If you select *Simulation Time*, then the rank results will show *Simulation Time* column added to the resultant table.

6. In case you want to specify attributes based on which the groups will be created while ranking, click the button next to the *Group by attributes* option. This will open the *Attributes Selector* dialog box, as shown in [Figure 3-78](#) on page 188.

Figure 3-78 Attributes Selector

Select the attributes based on which the groups will be created



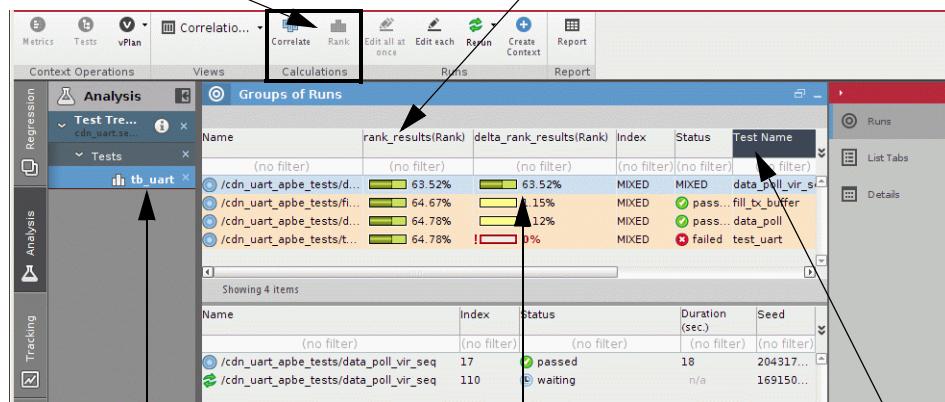
Note: If the groupings are already applied on the table data before performing the rank operation, then the *Attribute Selector* dialog box will show the attributes on which grouping was done in the *Selected Attributes* list box. You can choose to keep the same attributes for grouping during ranking or you can remove them as per your requirements.

7. Select the required attribute(s). For example, select *Test Name* attribute, and click the *Add* button.
8. Click *OK* to close the *Attributes Selector* dialog box.
9. You will now return to the *Rank Runs* dialog box. By default, an auto-generated name shows in the *Name column name* text box. You can choose to keep the auto-generated name or specify a name of your choice. This appears as the name of the column that will include rank results. For example, specify the name as *rank_results*, and click *Rank*.

[Figure 3-79](#) on page 189 shows the rank results.

Figure 3-79 Rank Results

New toolbar added Name you specified in the *Rank Runs* dialog box.



This is the name of the entity you selected in the *Entity Selection* dialog box.

This indicates the overall ranking efficiency.

Attribute specified for grouping rank results

When you rank the runs:

- ❑ A new analysis page is created to show the rank results. The page is named based on the entity you select in the *Entity Selection* dialog box. If a page already exists with the same name, 1 is appended to the name and with subsequent opening of the same page, this number is incremented automatically.
- ❑ Following new columns are created:
 - <name> (*Rank*)—Indicates the overall contribution of the run to total coverage. <name> indicates the name specified in the *Rank Runs* dialog box at the time of ranking.
 - delta_<name> (*Rank*)—Indicates the additional contribution made by the run to the overall coverage grade in comparison to the run listed above it.

Note: If the grouping was applied at the time of ranking, then the new columns stated above are shown only in the upper table. The lower table shows only the group information.

Note: Grouping actions are unavailable on the table that shows ranking results. This is because calculation of ranking grades is complex and can take long time. Other actions, such as filtering and sorting do not change the ranking grades and are therefore allowed. In addition, applying groupings after ranking is not required, and can confuse the user.

- A new toolbar *Calculations* is added. This toolbar has two buttons: *Correlate* and *Rank*. Only *Correlate* button is active if you are on the *Ranks* page. Using the *Correlate* button, you can add a new correlation column to the *Runs* table, as required.

Note: Ranking context will show disabled if the underlying coverage model is extremely large.

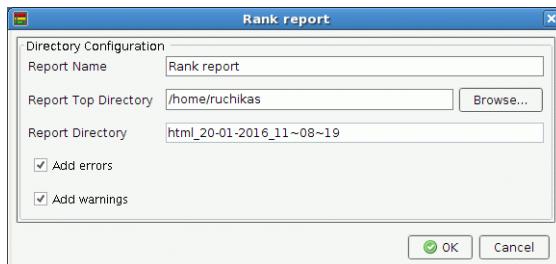
3.8.2 Report on Rank Page

To generate report on *Rank* page:

1. In the *Rank runs* analysis page, click the *Report* button in the Report toolbar.

The *Rank report* dialog box is displayed, as shown in [Figure 3-80](#) on page 190.

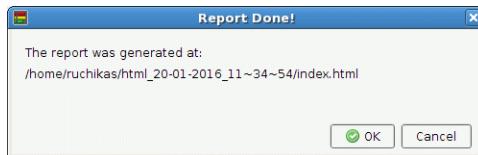
Figure 3-80 Rank Report



2. Specify the name of the report in the *Report Name* text box. This name will appear on the HTML report.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`.
5. By default, Errors table is shown in the rank report. To not show the Errors table in the report, clear the *Add errors* check box.
6. By default, Warnings table is shown in the rank report. To not show the Warnings table in the report, clear the *Add warnings* check box.
7. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 3-81](#) on page 191.

Figure 3-81 Report Done



This dialog box shows the location where the report is generated.

8. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 3-82 on page 191](#) shows the generated report.

Figure 3-82 Rank Report

The screenshot shows a web browser window titled "Rank report". The URL in the address bar is `file:///Z:/html_20-01-2016_11~21~09/index.html`. The page content includes:

- Report name:** cadence Rank report
- Report location:** Generated by ruchikas on 2016/01/20 11:21:14 (host ldvopt311)
- Context info:** sessions: cdn_uart_segal.11_01_04_23_14_18_1797; loaded refinements: not loaded
- Ranked runs:** A table showing four rows of test results:

Name	rank_results(Rank)	delta_rank_results(Rank)	Index	Status
/cdn_uart_apbe_tests	63.52%	63.52%	MIXED	MIXED_GRC
/data_poll_vir_5eq				
/cdn_uart_apbe_tests	64.67%	1.15%	MIXED	passed
- Errors associated with selected run:** A table showing errors for the selected run:

Name	Description	Tool
dut_error	Scoreboard DUV to eVC scoreboard - detected frame did not match expected frame detected frame data: 0 0 1 0 0 1 0 expected frame data: 0 0 1 0 0 1 0	specman
test_end	run did not complete properly.	specman
- Warnings associated with selected run:** A table showing warnings for the selected run:

Name	Description	Tool
	Scoreboard DUV to eVC scoreboard - detected frame did not match expected frame.	

The report shows the information such as, who generated the report and when the report was generated. It also includes the context information, such as session and loaded refinement file.

The report shows following tables:

- Runs—This table lists the runs ranked on the basis of their efficiency.
- Errors —This table lists the errors associated with the selected run.
- Warnings—This table lists the warnings associated with the selected run.

The columns shown in the report, the sort order, and data shown in the report (filters) is based on the current state of the table at the time of generating the report. If required, you can adjust the columns that you want to include or remove from the table, apply sort order, and also apply filters before generating the report.

After you have generated the report, you can filter data based on the *Name* field using the *Filter by name* text box provided at the top of each table in the report.

3.8.3 Enable Parallel Ranking

Ranking is a time-consuming process which involves significant computation and heavy disk access. In case you want to analyze many runs (thousands of runs), you might have to wait long before you can analyze the ranked results.

To save time, you can accelerate the rank calculation by utilizing multiple processes (potentially on multiple hosts). You can configure vManager such that the rank calculation is dispatched to multiple processes using the local host or to different hosts using DRM systems such as LSF and SGE.



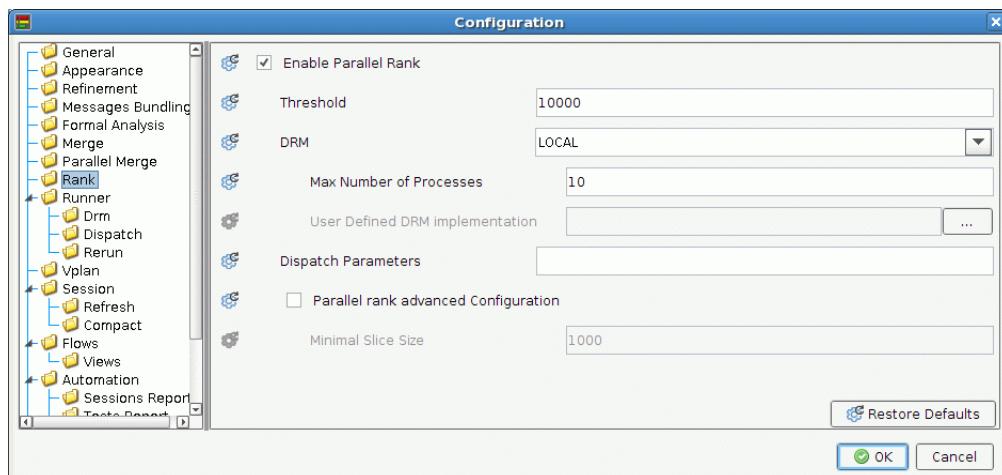
Parallel ranking is not always the best choice. There is an overhead associated with it, such as distributing the ranking operation to multiple processes (process creation, communication, synchronization, and so on.) Parallel ranking should not be used for small cases.

To enable vManager to perform Parallel ranking, perform the following steps:

1. Select *View* —> *Configuration*.
2. Select the *Rank* folder in the left pane.
3. The *Configuration* dialog box displays the *Rank* options. Select the *Enable Parallel Ranking* check box. This enables options related to Parallel ranking.

[Figure 3-83](#) on page 193 shows the *Parallel Ranking* options.

Figure 3-83 Configuration (Parallel Ranking)



4. In the *Threshold* field, specify the number of runs after which the parallel ranking should be performed. The default threshold value is 10000, which means that the parallel ranking will be performed only after the number of runs for ranking exceeds the threshold value of 10000.
5. Next, specify the DRM type to be used for launching the processes in the *DRM* drop-down list. By default, Local (*LOCAL*) is used. You can select any of the following DRMs:
 - LSF — To use the integration with IBM Platform LSF for launching the processes.
 - SGE — To use open source batch-queuing system, Grid Engine for launching the processes.
 - UGE — To use Univa Grid Engine for launching the processes.
 - OPENLAVA — To use OpenLava workload scheduler for launching the processes.
 - NC — To use NetworkComputer based job scheduler for launching the processes.
 - LOCAL — To schedule execution of runs in parallel or serial mode on the local machine for launching the processes.
 - USER_DEFINED — To use DRM that you created using the DRM API for launching the processes.
6. In the *Max Number of Processes* field, specify the maximum number of processes to be used while ranking. The default value specified is 10. For Local DRM, the valid values are 2 to 50. For any other DRM, you can specify any integer 2 or more than 2.

7. If DRM is specified as `USER_DEFINED`, then *User Defined DRM Implementation* field becomes active. In this field, specify the TCL file (that includes the DRM API) to be used. For more details on creating a DRM API for parallel ranking, see *Incisive vManager Installation and Configuration Guide*.
8. Specify the dispatch parameters for the DRM in the *Dispatch Parameters* field.
9. Select the *Advanced Configuration* check box in case you want to enable specifying the number of runs to be sent to each process. After you select this check box, the *Minimal Slice Size* option becomes active.
10. In the *Minimal Slice Size* field, specify the number of runs to be sent to a single process. The default value is 1000.
11. Click OK.

Note: You can also enable parallel ranking using the `config` CLI command. To enable parallel ranking in CLI mode, use:

```
config rank.enable_parallel_rank -set true
```

Few CLI examples are mentioned below.

Task	Command
Enable parallel ranking	<code>config rank.enable_parallel_rank -set true</code>
Set DRM to LOCAL	<code>config rank.rank_drm -set LOCAL</code>
Set the DRM dispatch parameters	<code>config rank.rank_drm_dispatch_parameters -set {-u /dev/null -R "select[(os==linux && osmajor==5)]" -W 60}</code>
Set the number of runs after which the parallel ranking should be performed	<code>config rank.parallel_rank_threshold -set 1002</code>
Set the maximum number of processes to be used while ranking as 3	<code>config rank.number_of_parallel_rank_processes -set 3</code>
Enable advanced configuration while parallel ranking	<code>config rank.parallel_rank_advanced_configuration -set true</code>
Set the number of runs to be sent to a single process as 2	<code>config rank.parallel_rank_min_slice_size -set 2</code>

Files Generated After Parallel Rank Process

After the parallel rank process, a log file named `rank_processes_logs.log_<timestamp>` is generated in the current working directory. This log file summarizes all the process activities that took place during rank.

[Figure 3-84 on page 195](#) displays a sample of processes log file.

Figure 3-84 Log File Generated After Parallel Rank (Sample)

Parallel Work Summary					
Configurations used:					
Threshold: 10000					
Number of process: 10					
DRM Type: DRM_CONTROL_LSF					
DRM Config dir: <DRM_Config_dir>					
DRM Dispatch Parameters:					
Index	Start Time	End Time	Running Time	Pending Time	Number Of Runs
1	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:27	1240
2	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:26	1240
3	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:57:30 IDT 2014	00:05:34	00:00:26	1240
4	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:25	1240
5	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:24	1240
6	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:24	1240
7	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:23	1240
8	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:57:30 IDT 2014	00:05:34	00:00:22	1240
9	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:21	1240
10	Sun Aug 03 09:51:55 IDT 2014	Sun Aug 03 09:56:49 IDT 2014	00:04:53	00:00:20	1245
12405 ucds were processed in: 00:06:03					

The above screen shows the summary of the rank process. Below the summary (in the `rank_processes_logs.log` file), a unified unordered sequential list of all the outputs coming from the running processes is shown.

3.9 Correlating Runs

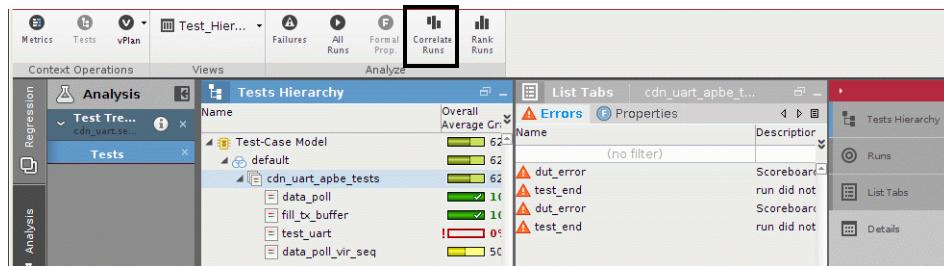
Correlating runs helps you see which runs in the current context contribute how much to an attribute such as grade, sampled space or checks for a particular element in the vPlan, including a section, coverage entity, scenario, or check.

3.9.1 Correlate Runs

To correlate runs:

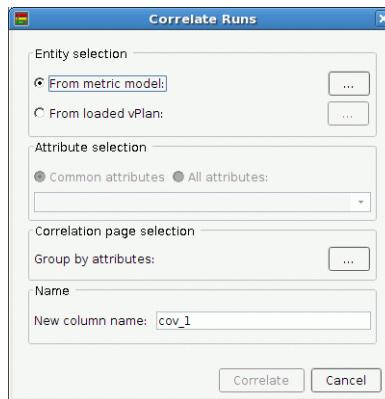
1. In the *Tests Hierarchy* pane, select the node for which you want to correlate runs.
- Note:** Alternatively, you can also select the runs (you want to correlate) from the *Runs* table.
2. Select *Correlate Runs* from the *Analyze* toolbar, as shown in [Figure 3-85](#) on page 196.

Figure 3-85 Correlating Runs



The *Correlate Runs* dialog box appears, as shown in [Figure 3-86](#) on page 196.

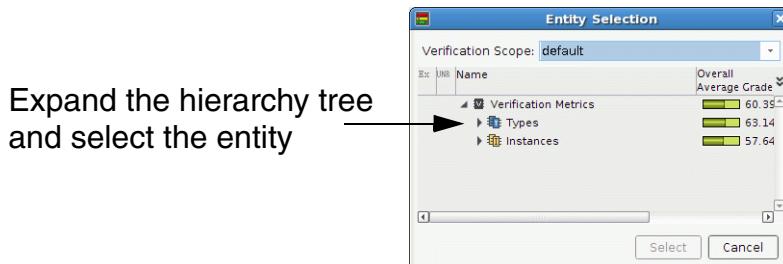
Figure 3-86 Correlate Runs



3. You need to select the entity for correlation. You can select the entity from the metric model or the loaded vPlan. To correlate runs based on metric model, select the *From metric model* radio button, and click the button next to it.

The *Entity Selection* dialog box appears, as shown in [Figure 3-87](#) on page 197.

Figure 3-87 Entity Selection

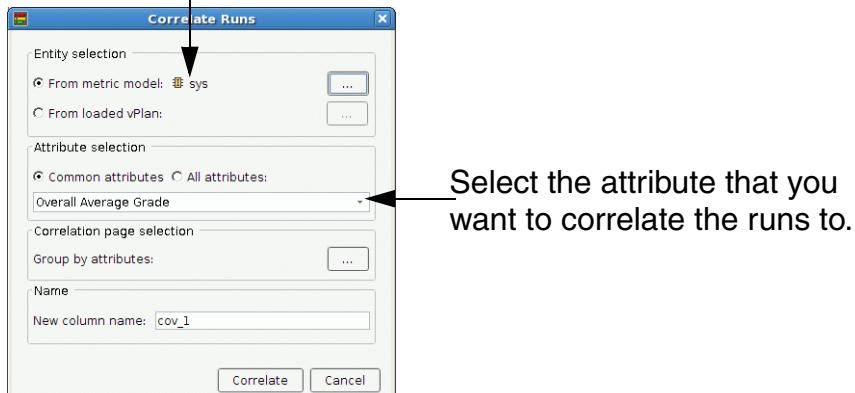


4. Expand the hierarchy tree and select the required entity. For example, select sys from the list of instances.
5. After you select the required entity from the domain hierarchy tree, click the *Select* button.

The *Correlate Runs* dialog box appears, as shown in [Figure 3-88](#) on page 197.

Figure 3-88 Correlate Runs

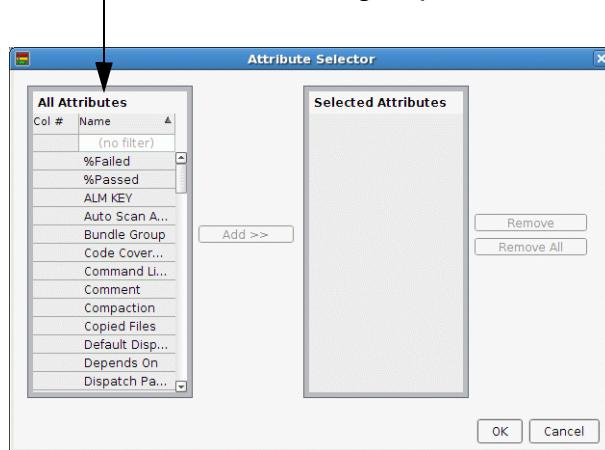
Indicates your selection (This will show as the page name in the navigation pane)



6. After making the entity selection, you need to select the attribute that you want to correlate the runs to. By default, the list box shows the most commonly used attributes. You can show all of the attributes in the list box by selecting *All Attributes* radio button. For example, to correlate the runs to overall average grade, select *Overall Average Grade* in the list box.
7. In case you want to specify attributes based on which the groups will be created while correlation, click the button next to the *Group by attributes* option. This will open the *Attributes Selector* dialog box, as shown in [Figure 3-89](#) on page 198.

Figure 3-89 Attributes Selector

Select the attributes based on which the groups will be created

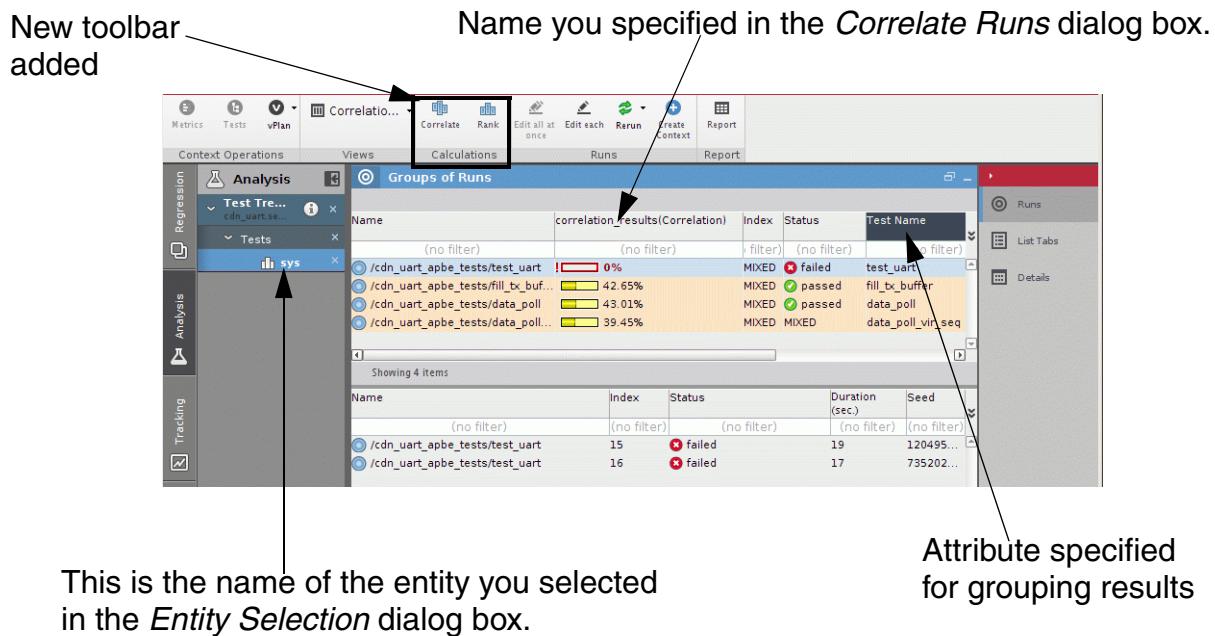


Note: If the groupings are already applied on the table data before performing the correlation operation, then the *Attribute Selector* dialog box will show the attributes on which grouping was done in the *Selected Attributes* list box. You can choose to keep the same attributes for grouping during correlation or you can remove them as per your requirements.

8. Select the required attribute(s). For example, select *Test Name* attribute, and click the *Add* button.
9. Click *OK* to close the *Attributes Selector* dialog box.
10. You will now return to the *Correlate Runs* dialog box. By default, an auto-generated name shows in the *Name* text box. You can choose to keep the auto-generated name or specify a name of your choice. This appears as the name of the column that will include correlation results. For example, specify the name as *correlation_results*, and click *Correlate*.

[Figure 3-90 on page 199](#) shows the correlation results.

Figure 3-90 Correlation Results



When you correlate the runs:

- A new analysis page is created to show the correlation results. The page is named based on the entity you select in the *Entity Selection* dialog box. If a page already exists with the same name, 1 is appended to the name and with subsequent opening of the same page, this number is incremented automatically.
- A new column is created. The column is named as <name> (*Correlation*), where <name> indicates the name specified in the *Correlate Runs* dialog box at the time of correlation. This column shows the contribution of the run to the attribute selected at the time of correlation.
- A new toolbar *Calculations* is added. This toolbar has two buttons: *Correlate* and *Rank*. Using these buttons, you can add a new correlation or rank column to the *Runs* table, as required.
- If the grouping was applied at the time of correlation, then the new column (<name> (*Correlation*)) stated above is shown only in the upper table. The lower table shows only the group information.

Note: Grouping actions are unavailable on the table that shows correlation results. This is because calculation of correlation grade is complex and can take long time. Other actions, such as filtering and sorting do not change the correlation grades and are therefore allowed. In addition, applying groupings after correlation is not required, and can confuse the user.

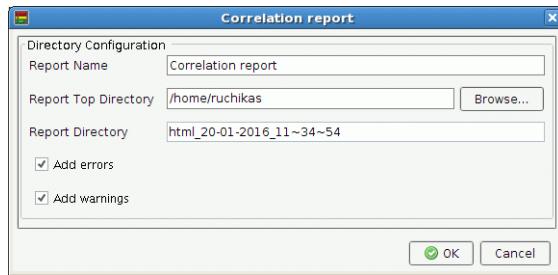
Note: Correlation context will show disabled if the underlying coverage model is extremely large.

3.9.2 Report on Correlation Page

To generate report on *Correlation* page:

1. In the *Correlate runs* analysis page, click the *Report* button in the Report toolbar.
The *Correlation report* dialog box is displayed, as shown in [Figure 3-91](#) on page 200.

Figure 3-91 Correlation Report



2. Specify the name of the report in the *Report Name* text box. This name will appear on the HTML report.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 3-92](#) on page 200.

Figure 3-92 Report Done



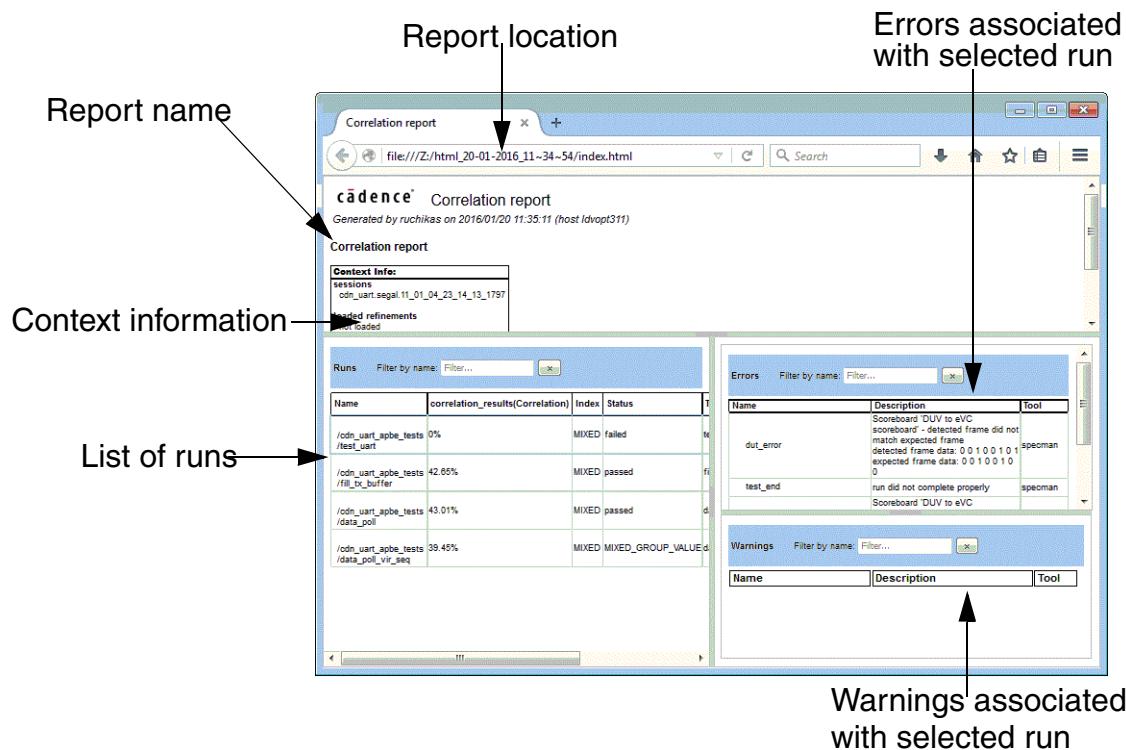
This dialog box shows the location where the report is generated.

6. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 3-93 on page 201](#) shows the generated report.

Figure 3-93 Correlation Report



The report shows the information such as, who generated the report and when the report was generated. It also includes the context information, such as session and loaded refinement file.

The report shows following tables:

- Runs—This table lists the correlation results.
- Errors —This table lists the errors associated with the selected run.
- Warnings—This table lists the warnings associated with the selected run.

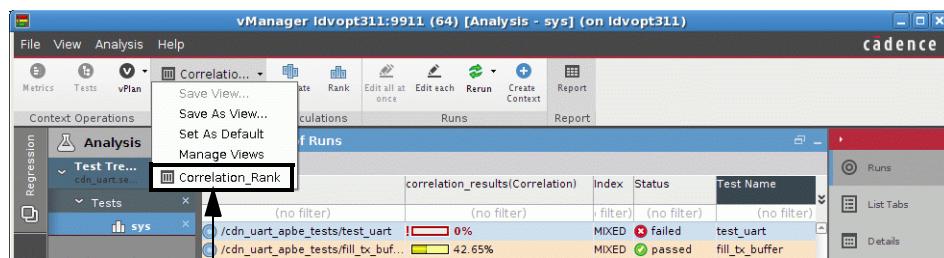
The columns shown in the report, the sort order, and data shown in the report (filters) is based on the current state of the table at the time of generating the report. If required, you can adjust the columns that you want to include or remove from the table, apply sort order, and also apply filters before generating the report.

After you have generated the report, you can filter data based on the *Name* field using the *Filter by name* text box provided at the top of each table in the report.

Ranking and Correlation Views

Both ranking and correlation operations create a new analysis page. This page has a predefined view *Correlation_Rank*, as shown in [Figure 3-94](#) on page 202.

Figure 3-94 Ranking and Correlation View



Predefined view

This view has following columns other than the ones created during the ranking or correlation operation:

- Name
- Index
- Status
- Duration
- Seed

Note: The table is sorted based on the column that is created automatically during ranking or correlation operation.

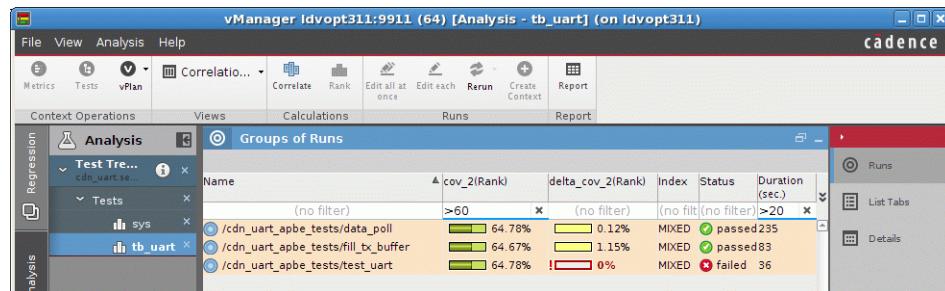
Similar to other pages, you can define a new view (based on your requirements), set it as default, and use it for reporting. For example, you can apply filters, change the order of columns, sort the table data, and so on.

For more details on defining views, see [Defining and Organizing Views](#) on page 43.

Note: Groupings applied at the time of ranking or correlation are not saved to the view. In addition, the automatically generated columns are not saved to the view.

For example, consider the groupings and filters applied on table shown in [Figure 3-95](#) on page 203.

Figure 3-95 Ranking and Correlation View Table



If the above settings are saved to a view, then the following items get saved to the view:

- Sorting on the *Name* column
- Filter on *Duration* column

Note: The automatically generated columns (in this case, cov_2 and delta_cov2) are not saved to the view.

Next time you apply this saved view to another rank or correlation page, all settings except the following items are applied:

- Grouping on *Status* column
- Availability of automatically generated columns

Note: The view once saved can also be used in the CLI mode using the `-view` option of the `rank` or `correlate` command. For example, you saved the view as `my_rank_r1`. You can apply it on other rank pages in CLI mode using the following command:

```
rank -inst uvm_pkg -view my_rank_r1
```

For more details on `rank` command, see [rank](#) on page 521.

For more details on `correlate` command, see [correlate](#) on page 530.

Important Considerations While Ranking and Correlation

While ranking or correlating runs, remember that:

- The excluded entities impact the ranking and correlation results. The excluded entities are not considered while ranking or correlating runs. For example, you excluded a type

`tb_uart`. After exclusion, if you perform a rank or correlate operation by selecting the *Type* node, then `tb_uart` and all its underlying entities are ignored.

- Ranking and correlation operations can be performed on runs that come from different verification scopes.

3.10 Generating Tests Report

Using vManager you can generate tests report in an HTML format and then publish it on the internet. Generating a report in an HTML format is useful for users who want to make the results available for viewing and analysis to people who might not have vManager installed with them.

The *Report* drop-down button on the *Report* toolbar has the following options for generating reports:

- All (Default) —To generate a tests report for all the tests.
- On Selected Entity—To generate a tests report for the selected test group or test.
- Summary Report—To generate a summary report. For more details, see [Generating Summary Report](#) on page 51.

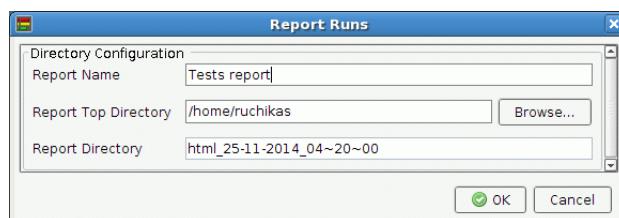
Note: You can also select these options from the *Analysis* menu.

For example, to generate a report for selected test group or test, perform the following steps:

1. Select the test group or the test whose report you want to generate. For example, select the test group `cdn_uart_apbe_tests`.
2. Right-click and select *Report on Selected Entity* from the pop-up menu.

The *Report Runs* dialog box is displayed, as shown in [Figure 3-96](#) on page 204.

Figure 3-96 Report Runs



3. Specify the name of the report in the *Report Name* text box. By default, the name is specified as *Tests report*. This name appears on the HTML report.

4. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
5. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`. For example, change the name as `all_tests`.
6. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 3-97](#) on page 205.

Figure 3-97 Report Done



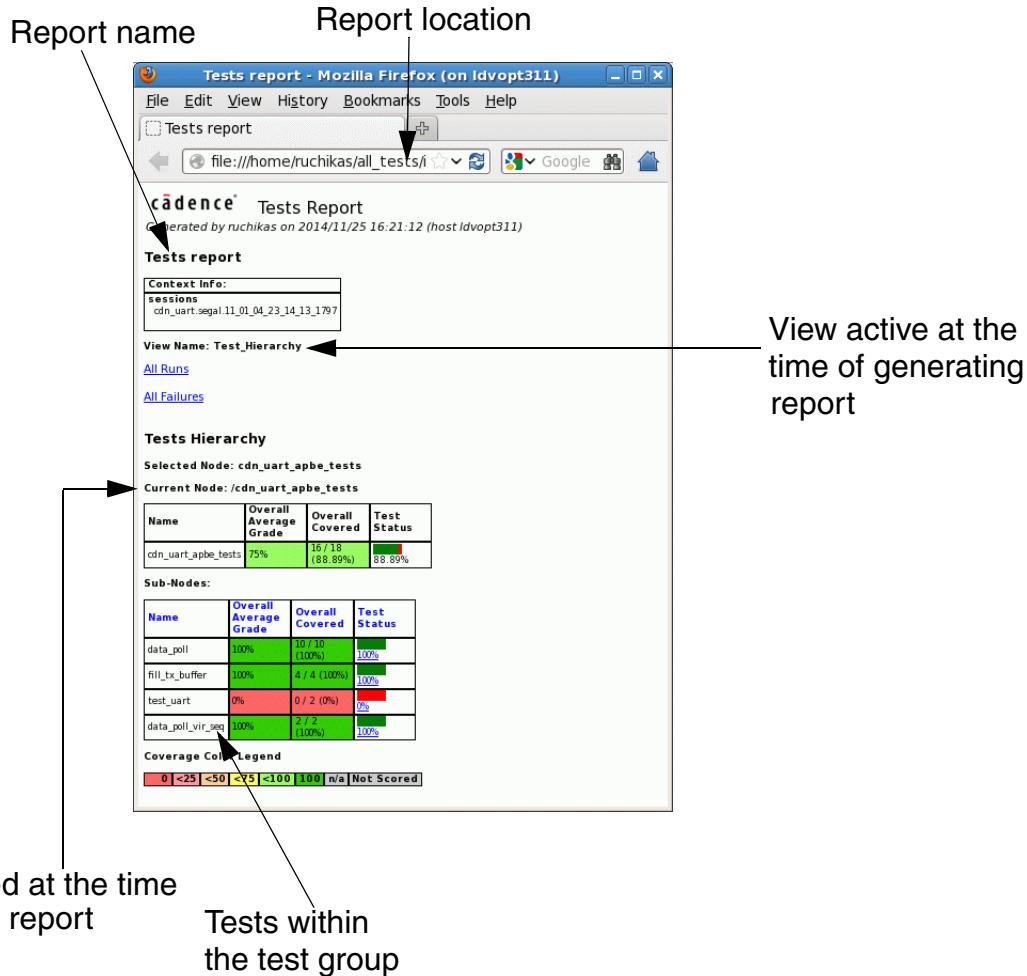
This dialog box shows the location where the report is generated.

7. Click *OK*.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 3-98](#) on page 206 shows the top-level page of the report.

Figure 3-98 Tests Report



This page shows snapshot summary of the selected test group or the test. It shows the details of tests within the selected test group and number of runs within each test.

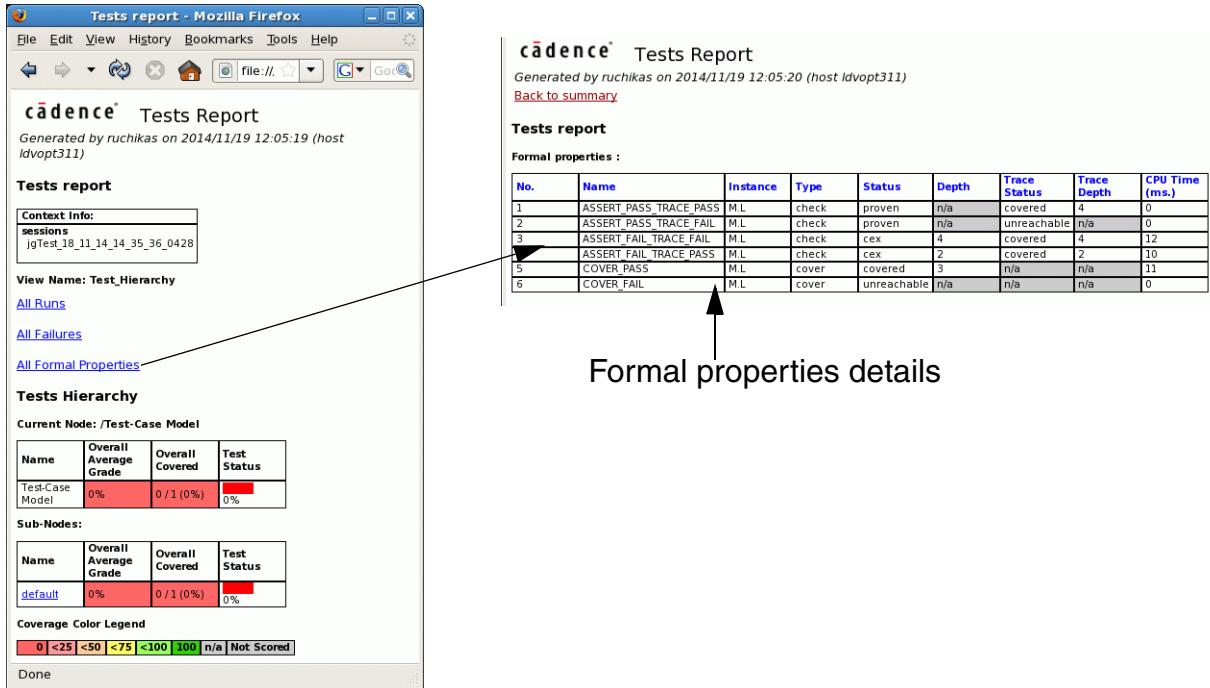
It also shows information such as, who generated the report and when the report was generated. It also shows the view used to generate the report and the node selected at the time of generating the report.

You can click the link named *All Runs* to view a flat list of runs from all of the tests. The runs are presented based on the settings of the Runs table in the selected view.

You can click the link named *All Failures* to view a flat list of failures from all runs in all of the tests. The failures are presented based on the settings of the Errors table in the selected view.

Note: In case you generated a report from a session that has formal properties, then you will see an additional link named *All Formal Properties*. You can click this link to view the view the formal properties details, as shown in [Figure 3-99](#) on page 207.

Figure 3-99 Tests Report (Formal Properties)



The screenshot shows two windows side-by-side. The left window is titled 'Tests report - Mozilla Firefox' and displays a 'Tests report' page. It includes sections for 'Context info', 'View Name: Test_Hierarchy', and 'Tests Hierarchy'. A link labeled 'All Formal Properties' is visible in the 'Tests Hierarchy' section. The right window is titled 'cadence® Tests Report' and shows a 'Formal properties' table. An arrow points from the 'All Formal Properties' link in the left window to the 'Formal properties' table in the right window. The table has columns: No., Name, Instance, Type, Status, Depth, Trace Status, Trace Depth, and CPU Time (ms.). The data in the table is as follows:

No.	Name	Instance	Type	Status	Depth	Trace Status	Trace Depth	CPU Time (ms.)
1	ASSERT PASS TRACE PASS	M.L	check	proven	n/a	covered	4	0
2	ASSERT PASS TRACE FAIL	M.L	check	proven	n/a	unreachable	n/a	0
3	ASSERT FAIL TRACE FAIL	M.L	check	cex	4	covered	4	12
4	ASSERT FAIL TRACE PASS	M.L	check	cex	2	covered	2	10
5	COVER PASS	M.L	cover	covered	3	n/a	n/a	11
6	COVER FAIL	M.L	cover	unreachable	n/a	n/a	n/a	0

Note: The report output and the columns shown in the report are based on the view selected at the time of generating the report. You can adjust the columns that must be included in the coverage report by adding or removing the attributes in GUI and save it as a view before generating the report.

Note: You can sort data in the table by clicking a column header.

To view the runs within the selected test or test group, click in the link shown in the *Runs#* column. For example, click in the Runs# column of test named *test_uart*.

[Figure 3-100](#) on page 208 shows the *Runs* table of the Tests report.

Figure 3-100 Tests Report (Runs Table)

Click to return to the top-level summary page → [Back to summary](#)

Runs table →

No.	Failures	Warnings	Index	Name	Status	Durat (sec.)
1	2	3	16	/cdn_uart_apbe_tests/test_uart	failed	17
2	2	3	15	/cdn_uart_apbe_tests/test_uart	failed	19

The *Runs* table shows the runs within the selected test or test group. The columns shown in the table are based on the columns available at the time of generating the report. For example, if a particular column is removed from the table in vManager, then the HTML report will also not include that column.

You can click on the links available in the Failures and Warnings column to view the details on failures and warning associated with the selected run.

[Figure 3-101](#) on page 209 shows the failures associated with the selected run.

Figure 3-101 Tests Report (Failures)

The table shows the failures associated with selected run.

No.	Name	Description	Severity	Category
1	dut_error	Scoreboard 'DUV to eVC scoreboard' - detected frame did not match expected frame detected frame data: 0 0 1 0 0 1 1 1 expected frame data: 0 0 1 0 0 1 0 0	error	first
2	test_end	run did not complete properly	error	other

The table shows the failures associated with selected run.
Similarly, you can generate more reports and navigate through them, as required.

3.11 Integrating vManager with Indago Debug Analyzer

vManager and INDAGO integration helps the INDAGO users to run their regression using vManager, and then use INDAGO to debug the failures. With the integration of vManager and INDAGO, the INDAGO users can perform the following tasks:

- [Opening INDAGO GUI from vManager on Exact Location of Error](#)
- [Setup Regression to Automatically Record IDA Data in Rerun of Failed Runs](#)

3.11.1 Opening INDAGO GUI from vManager on Exact Location of Error

vManager supports following test/group attributes to allow recording of INDAGO data during regression:

- `use_ida_record`—To enable INDAGO recording.
- `ida_db_start_time`—To set the start time for recording.
- `ida_probe_options`—To specify probe options for recording INDAGO data.

- `ida_recording_window`—To define recording window for INDAGO recording. It is used only when rerunning runs to determine start time based on failure time in the original run. (Default is 1000 FS).

Important

The *Incisive Debug Analyzer (IDA)* is now *Indago Debug Analyzer*. For backward compatibility, the names of attributes are not changed from ida to indago.

Note: These attributes must be used in a vsif file. If you define these attribute in the vsif, vManager exports that definition as an environment variable (`BRUN_*`). However, the simulator does not automatically pick these up. You must use these in your run script. For more details on using these attributes, see [Incisive vManager Installation and Configuration Guide](#).

To open INDAGO GUI on exact location of the error, perform the following steps:

1. Create a vsif and a run script with INDAGO related attributes.

[Figure 3-102 on page 210](#) shows a sample vsif file.

Figure 3-102 vsif File

vsif file

```
// ex_brun.vsif: an vsif that runs the vm_basic example

session vm_basic {
    top_dir : ${ENV(DEMO_HOME)}/clab_sv/sessions;
    output_mode: terminal;
    session_description: <text>This example runs 30 basic runs using Specman</text>;
    post_session_script: vm_analyze.pl;
    ida_recording_window: 9000NS;
};

group basic {
    count: 5;
    run_script: <text>${ENV(DEMO_HOME)}/clab_sv/scripts/run.csh:${ENV(DEMO_HOME)}/clab_sv/scripts/run.csh</text>;
    scan_script: "vm_scan.pl ius.flt";
    test_smokes {
        timeout: 600;
        top_files: "random_test";
    };
};
```

↑

Calls a run script in which INDAGO attributes are enabled

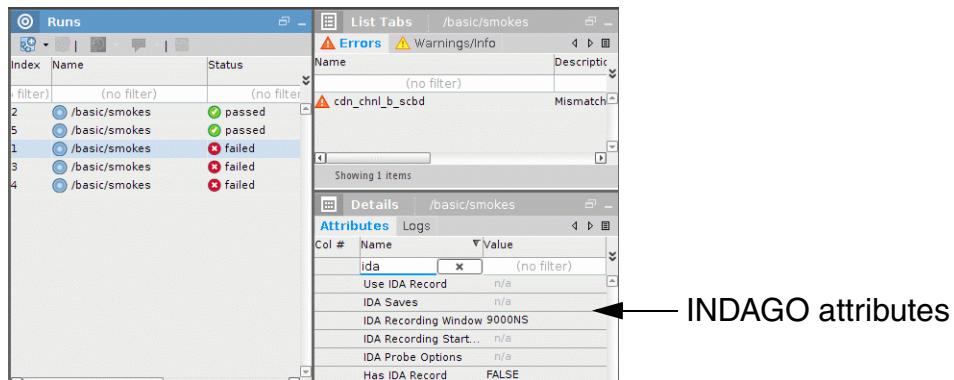
2. Invoke vManager.
3. Launch vsif (that includes IDA related attributes).

Note: By default, IDA recording is not done because the `use_ida_record` attribute is disabled by default. You can enable IDA recording at the time of rerun by selecting `IDA_BATCH_DEBUG` run mode.

4. Open failure analysis. For example, select *Analyze Failed Runs* from the *Analyze* toolbar.
5. Now select runs for a rerun.

[Figure 3-103 on page 211](#) shows INDAGO related attributes before enabling IDA recording.

Figure 3-103 INDAGO Attributes Before IDA Recording



6. To enable IDA recording, click *Rerun* button in the *Runs* toolbar.
7. The *Rerun* dialog box will be displayed. Select *INDAGO_BATCH_DEBUG* from the *Select scheme* drop-down list. This will enable INDAGO recording.

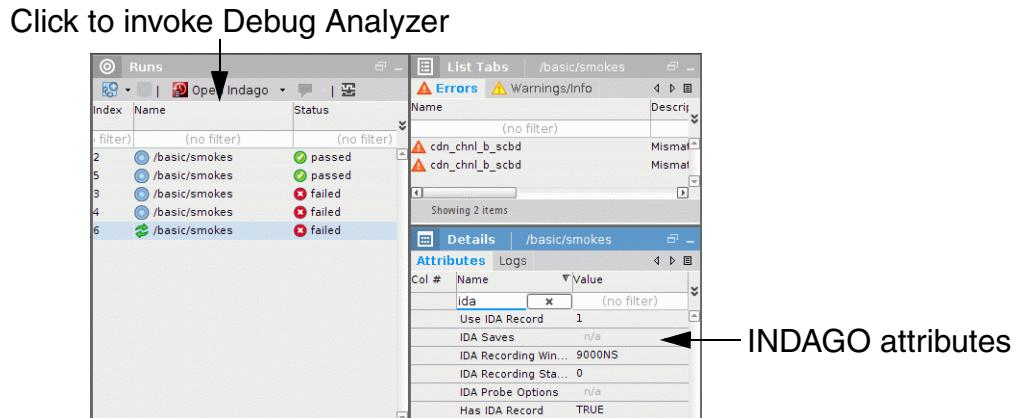
The following tasks will be performed:

- ❑ The value of `use_ida_record` attribute is set to 1. (The run script that was created earlier is executed.)
- ❑ The value of `ida_db_start_time` is calculated based on the `ida_recoding_window` and the simulation time of the original run:
`test.ida_db_start_time =max(0, run.simulatio_time - run.ida_recording_window)`
- ❑ The timeout attribute value is multiplied by 5 (run with INDAGO recording is expected to be slower).

8. The IDA data is now recorded.

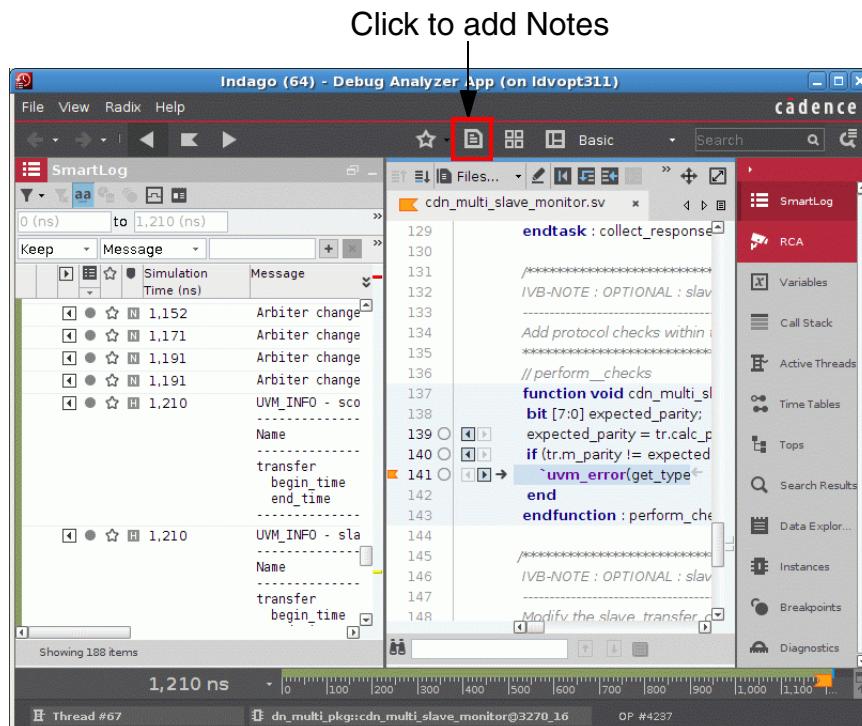
[Figure 3-104 on page 212](#) shows INDAGO related attributes after INDAGO recording.

Figure 3-104 INDAGO Attributes After IDA Recording



9. After the INDAGO data is recorded, the *Open Indago* button is enabled.
10. To invoke *Debug Analyzer*, click the *Open Indago* button in the Runs table. This will launch *Debug Analyzer* at the exact location of the error, as shown in [Figure 3-105](#) on page 212.

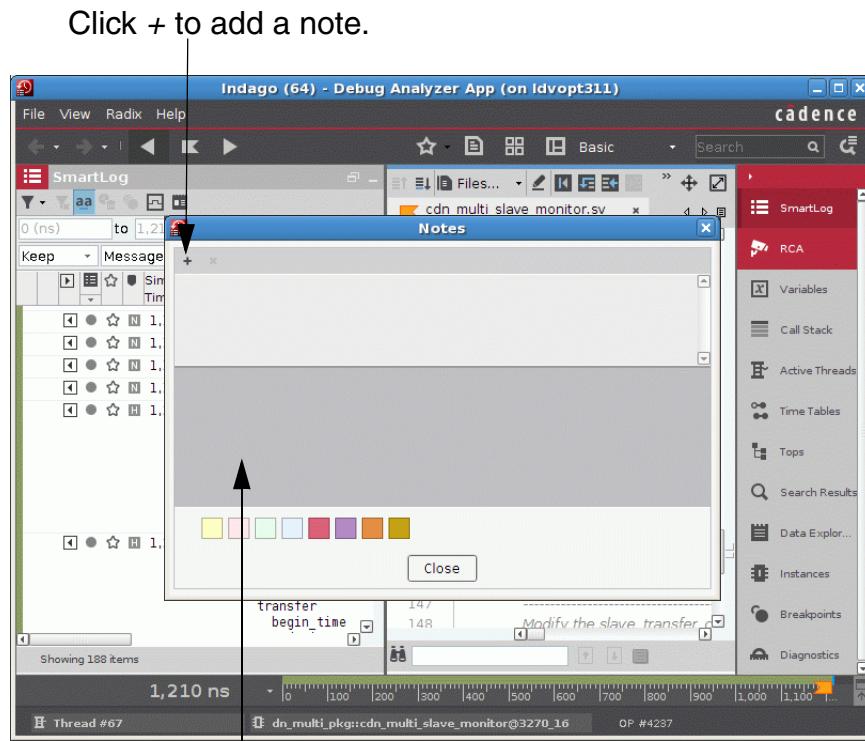
Figure 3-105 Debug Analyzer



11. You can now add notes, as required. For this, click the *Notes* button in the toolbar.

The *Notes* dialog box is displayed, as shown in [Figure 3-106](#) on page 213.

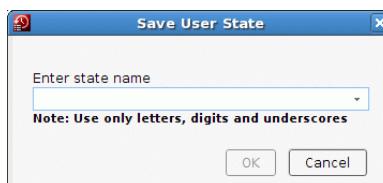
Figure 3-106 Debug Analyzer



12. Click the + sign to add a note. This will enable the text box and allow adding of notes. For example, add the text `Mike, fix this line` as the note.
13. Close the *Notes* dialog box by clicking the X sign on the right-side top of the *Notes* dialog box. The note gets added.
14. Now, save the current user state by selecting *Save User State* from the *File* menu of *Indago Debug Analyzer*.

The *Save User State* dialog box is displayed, as shown in [Figure 3-107](#) on page 213.

Figure 3-107 Save User State



- 15.** Specify the state name. For example, specify state name as *For_mike*, and click *OK*.

Note: The state name you specify in this dialog box will appear in the drop-down list in *Runs* table of vManager using which you will be able to launch *Indago Debug Analyzer*. It will also be available in drop-down of Notes icon button of Runs table using which you will be able to view the notes associated with a particular record.

- 16.** Now, return to the vManager window. Select *Indago_Debug_Analyzer* view from the Views drop-down list. This step is optional. This will show you records with appropriate INDAGO fields in the *Runs* table, as shown in [Figure 3-108](#) on page 214.

Figure 3-108 Indago_Debug_Analyzer View

INDAGO fields

Name	Status	Duration (sec)	Top Files	Has IDA Record	IDA Saves
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
1 /basic/smokes	Failed	29	random_test	n/a	
2 /basic/smokes	Failed	11	random_test	n/a	
3 /basic/smokes	Failed	11	random_test	n/a	
4 /basic/smokes	Failed	9	random_test	n/a	
6 /basic/smokes	Failed	16	random_test	For_mike	For_mike

Errors		Warnings/Info	
Name	Description	Name	Description
(no filter)	(no filter)	(no filter)	(no filter)
cdn_multi_slave_...	recieved transfer with Wrong Pa...	cdn_multi_slave_...	recieved transfer with Wrong Pa...

Showing 2 items

Note: You can also apply filters on INDAGO fields, as required.

- 17.** You can now perform the following actions:

- View the notes associated with the selected record
- Launch *Indago Debug Analyzer* using saved state and continue debugging

- 18.** To view the notes associated with the selected record:

- a. Click the *Note* icon drop-down button.
- b. The *Note* drop-down, lists the user states that were saved in *Indago Debug Analyzer*. For example, in this case, it shows *For_mike*, as shown in [Figure 3-109](#) on page 214.

Figure 3-109 Notes drop-down list



- c. Select *For_mike* to view the notes.

A dialog box appears and the notes associated with selected records are shown. You can take appropriate action, as required.

19. To launch *Indago Debug Analyzer* using saved state:

- d. Click the *Open_Indago* drop-down button.
- e. The *Open INDAGO* drop-down, lists the user states that were saved in *Indago Debug Analyzer*. For example, in this case it shows *For_mike*, as shown in [Figure 3-110](#) on page 215.

Figure 3-110 Open Indago drop-down list



- f. In case you want to start debugging from the point where Mike stopped, select *For_mike* from the drop-down list.

This will launch *Indago Debug Analyzer* at a point where the user left debugging.

3.11.2 Setup Regression to Automatically Record IDA Data in Rerun of Failed Runs

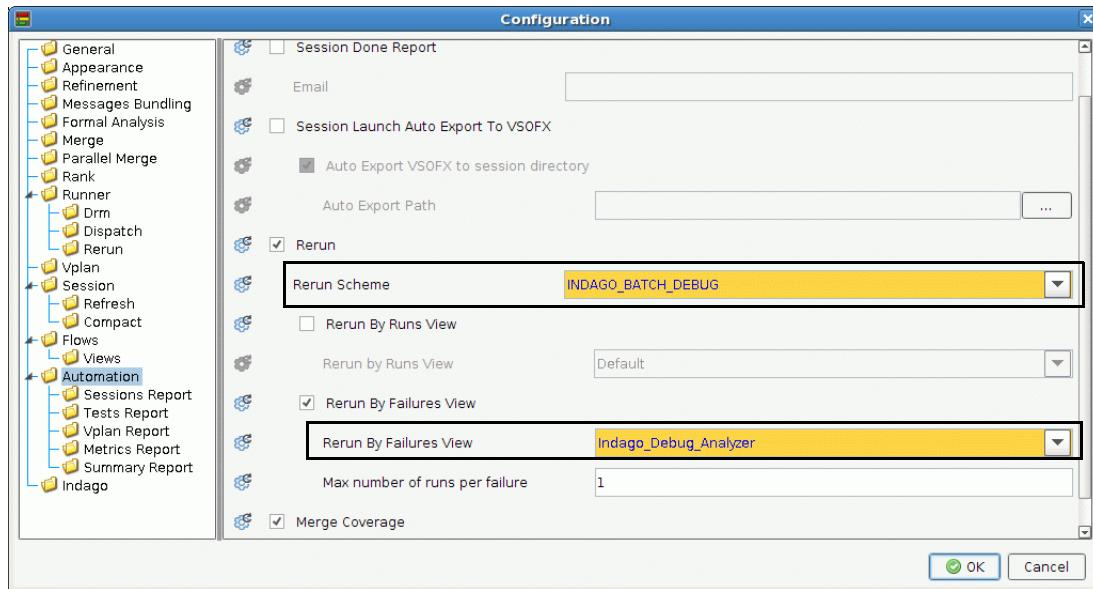
You can set up the regression to automatically record IDA data at the time of rerun in the case of failed runs. For this:

1. Select *View* —> *Configuration*.
2. Select the *Automation* folder in the left pane.
3. Select the *Rerun* check box.
4. Select *INDAGO_BATCH_DEBUG* from the *Rerun Scheme* drop-down list.
5. Select the *Rerun by Failure View* check box.
6. Select *INDAGO_Debug_Analyzer* from the *Rerun by Failure View* drop-down list.

[Figure 3-111](#) on page 216 shows the required options selected.

Incisive vManager User Guide

Figure 3-111 Configuration (Automation Options)



For more details, on *Configuration* dialog box, see [Incisive vManager Installation and Configuration Guide](#).

Analyzing Metrics

After loading a session or set of runs, you can analyze metrics data scored in that run. This chapter describes how to analyze the metrics data, and how to apply refinements on the metrics data.

This chapter covers the following topics:

- [Launching Metrics Analysis From Regression Center](#) on page 217
- [Launching the Metrics Analysis From Analysis Center](#) on page 218
- [Metrics Summary Page Overview](#) on page 220
- [Analyzing Code Coverage](#) on page 229
- [Analyzing FSM Coverage](#) on page 229
- [Analyzing Functional Coverage](#) on page 229
- [Analyzing Coverage Coming from Palladium](#) on page 235
- [Analyzing Coverage Coming from JasperGold](#) on page 237
- [Refining Metrics Data](#) on page 238
- [Coverage Grading](#) on page 239
- [Generating Metrics Report](#) on page 249

4.1 Launching Metrics Analysis From Regression Center

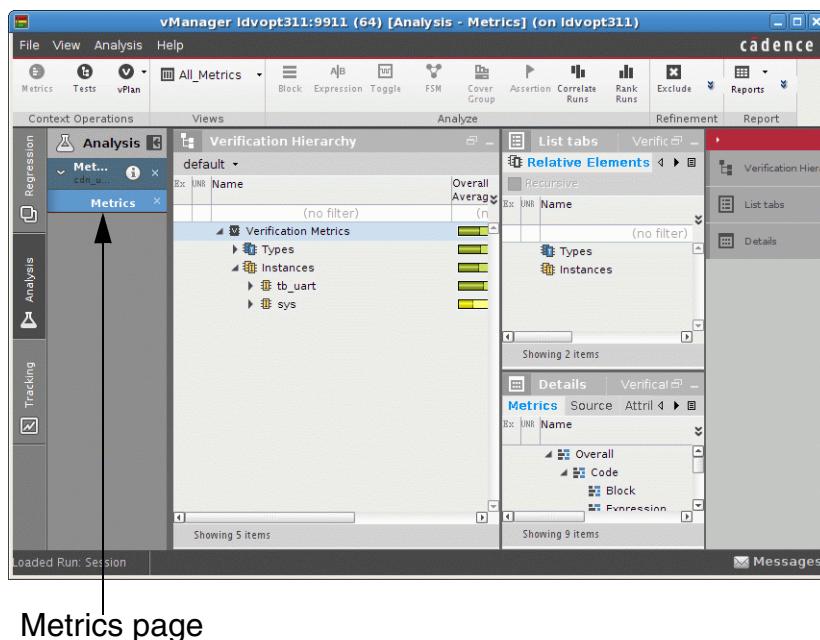
To launch the metrics analysis from the *Regression* center:

1. In the *Regression* center, select the session whose metrics you want to analyze.
2. Select *Metrics* from the *Analyze* toolbar.

Note: Alternatively, you can right-click the session and select *Analyze Metrics* from the pop-up menu. You can also select *Analyze Metrics* from the *Regression* menu.

The *Analysis* center is invoked, as shown in [Figure 4-1](#) on page 218.

Figure 4-1 Analysis Center—Metrics Page



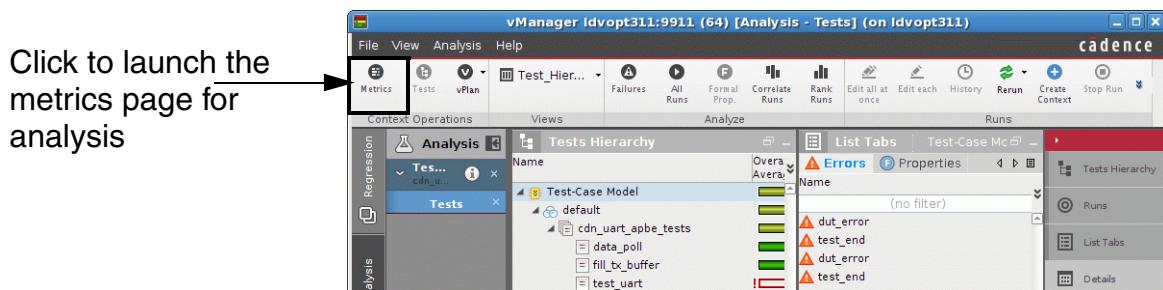
When you select the *Analyze Metrics*, the *Analysis* center launches the *Metrics Summary* page.

Note: Metrics context will show disabled if the underlying coverage model is extremely large.

4.2 Launching the Metrics Analysis From Analysis Center

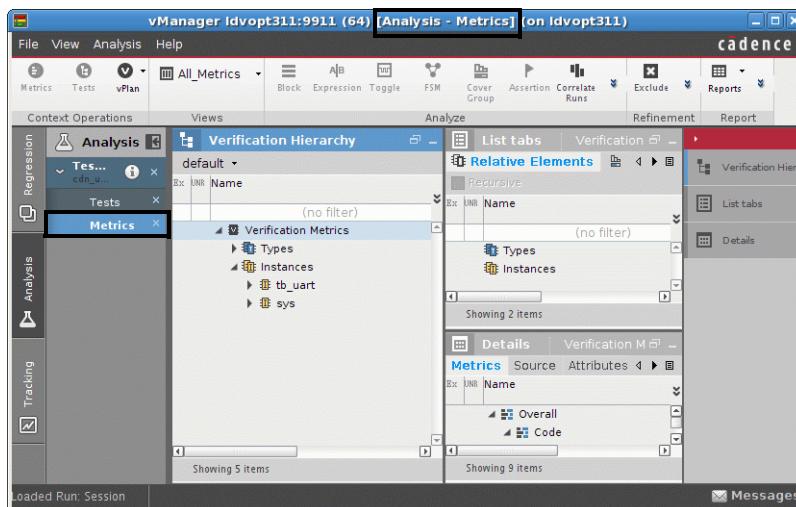
If you are already in the *Analysis* center, you can quickly launch metrics analysis by clicking the *Metrics* button in the *Context Operations* toolbar, as shown in [Figure 4-2](#) on page 219.

Figure 4-2 Analysis Center—Launching Metrics Page



This will launch the *Metrics Summary* page, as shown in [Figure 4-3](#) on page 219.

Figure 4-3 Analysis Center— Metrics Summary Page



By default, when you launch *Metrics* analysis, the *All_Metrics* view is launched. However, you can change the default view in the *Configuration* dialog box by:

1. Select *View* —> *Configuration*.
2. Select *Views* under the *Flows* folder in the left pane.
3. Select the *Override default view for Metrics analysis* check box.
4. Select the desired view from the *Metrics Analysis Default View* drop-down.
5. Click *OK*.

This will set the desired view as the default view when you launch the *Metrics* analysis.

4.3 Metrics Summary Page Overview

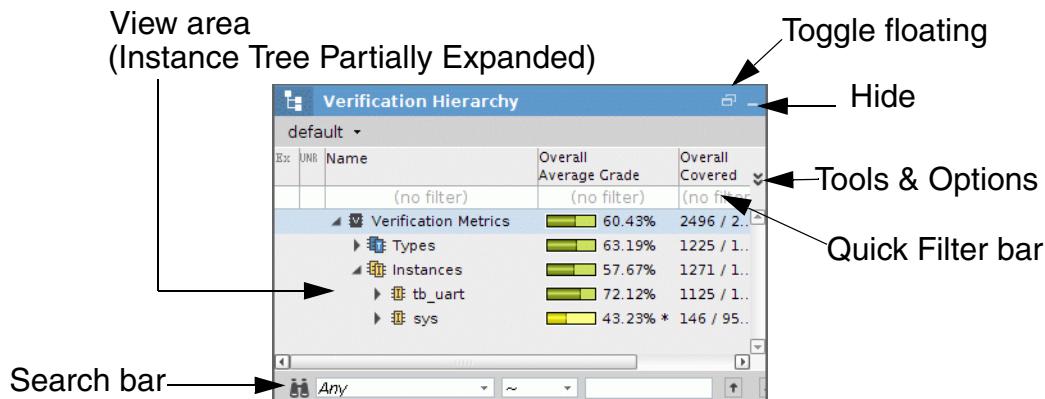
The *Metrics Summary* page is divided into the following areas:

- [Verification Hierarchy Pane](#)
- [List tabs Pane](#)
- [Details Pane](#)

4.3.1 Verification Hierarchy Pane

Figure 4-4 on page 220 displays the verification hierarchy pane of the *Metrics* page. Using the verification hierarchy pane, you can navigate through the design hierarchy. The *Verification Hierarchy* pane by default, displays the instance hierarchy partially expanded, and information such as overall average grade and overall covered.

Figure 4-4 Summary Page—Verification Hierarchy Pane



The main components of the *Verification Hierarchy* pane are:

- [View Area](#)
- [Tools & Options](#)
- [Toggle Floating](#)
- [Hide](#)
- [Quick Filter Bar](#)
- [Search Bar](#)

4.3.1.1 View Area

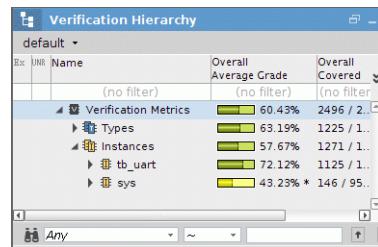
In the view area, at the top of the pane, an artificial node, Verification Metrics, is displayed.

Below the Verification Metrics node, there are following two nodes:

- *Types* node—Displays a list of all of the types scored within the design.
- *Instances* node—Displays a list of all of the instances scored within the design.

By default, when you load a run, the view area displays the *Instances* node partially expanded, as shown in [Figure 4-5](#) on page 221.

Figure 4-5 Instance Node Expanded



To expand the *Types* node, click the + symbol next to *Types*.

In the view area, you can navigate through the various instances and types scored in the design.

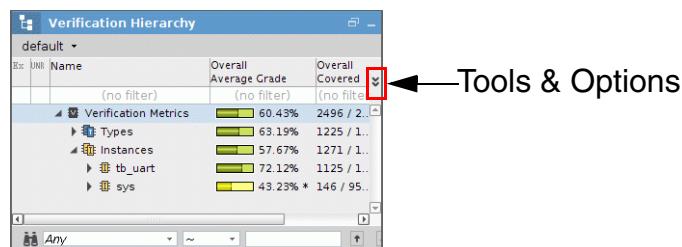
Note: The attributes or columns shown in the *Verification Hierarchy* pane are based on the selected view. You can select a view based on your requirements from the *Views* toolbar. For more details on available views, see the [Incisive Metrics Center User Guide](#).

4.3.1.2 Tools & Options

The *Tools & Options* is a double arrow button which when clicked shows the options that allows you to add or remove columns in the view area, show or hide the search bar, and so on.

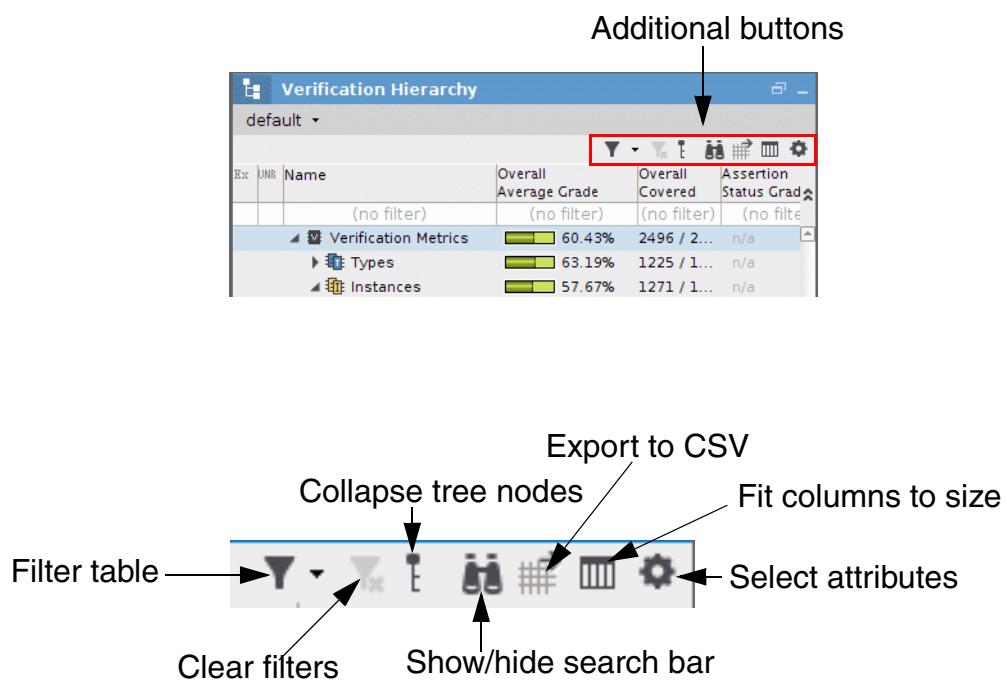
[Figure 4-6](#) on page 222 displays the *Tools & Options* drop-down button.

Figure 4-6 Tools & Options



When you click the Tools & Options button, additional buttons are displayed in the pane, as shown in [Figure 4-7](#) on page 222.

Figure 4-7 Tools & Options



The *Tools & Options* has following options:

- Filter Table—To apply quick filter, advanced filter, and show hide quick filter bar. For more details, see [Filtering Table Data](#) on page 33.
- Clear Filters—To remove filters from all fields.
- Collapse All—To collapse the instance hierarchy tree and show only the Verification Metrics node.

- Show Search Bar—To show or hide the Search Bar shown at the bottom of the verification hierarchy pane.
- Export to csv file —To export the metrics tree to the CSV file.
- Fit Columns to Size—To resize the width of the columns to its original default size.
- Select Attributes—To add or remove columns.

4.3.1.3 Toggle Floating

The toggle floating icon on the title bar of the pane is used to detach panes from the view area and move it, as required.

For more details, see [Detaching Panes](#) on page 39.

4.3.1.4 Hide

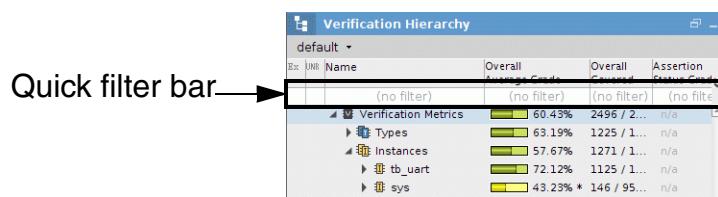
The toggle floating icon on the title bar of the pane is used to hide panes, as required.

For more details, see [Hiding Panes](#) on page 41.

4.3.1.5 Quick Filter Bar

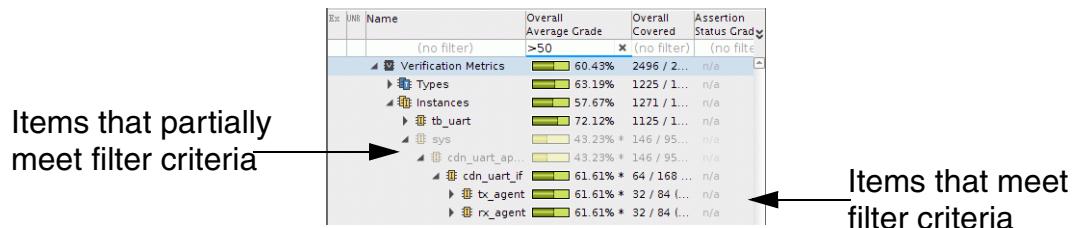
[Figure 4-8](#) on page 223 displays the Quick filter bar shown at the top of the verification hierarchy pane of the *Metrics* page. The Quick filter bar helps you to filter the verification hierarchy pane based on the filter criteria.

Figure 4-8 Verification Hierarchy Pane—Filter Bar



For example, to filter the verification hierarchy tree to show items for which *Overall Average Grade* is more than 50, specify >50 in the filter row and press Enter. After you press Enter, the verification hierarchy tree is filtered, as shown in [Figure 4-9](#) on page 224.

Figure 4-9 Verification Hierarchy Pane—Filtered



The filtered hierarchy tree shows following kinds of items:

- Items that meet the filter criteria. These items are shown as regular items.
- Items that partially meet the filter criteria. These items are shown as partially transparent. These are the items that do not meet the filter criteria completely, but appear because of their children.

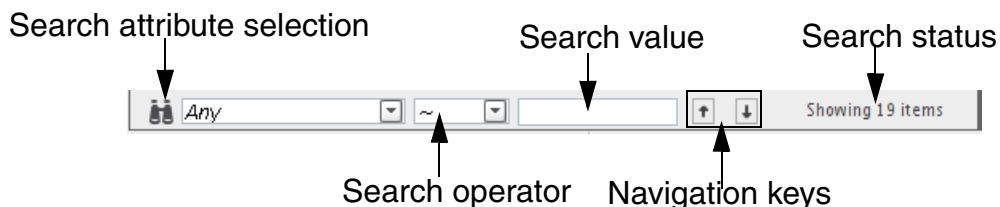
The items that do not meet the filter criteria are removed from the hierarchy tree.

vManager also enables you to apply advanced filters. For more details, see [Advanced Filtering](#) on page 37.

4.3.1.6 Search Bar

[Figure 4-10](#) on page 224 displays the search bar shown at the bottom of the verification hierarchy pane of the *Summary* page. The search bar helps you to quickly navigate through the instances or types that meet the specified search criteria.

Figure 4-10 Verification Hierarchy Pane—Search Bar

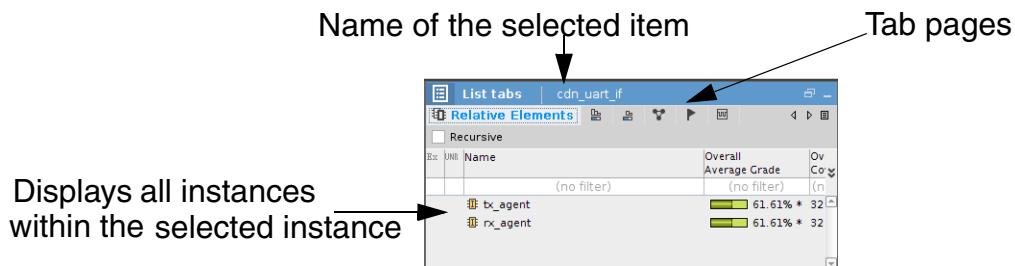


For more details on using the search bar, see [Searching Data in Tables](#) on page 31.

4.3.2 List tabs Pane

[Figure 4-11](#) on page 225 displays the *List tabs* pane of the *Metrics* page. The *List tabs* pane has different tab pages to list relative elements, covergroups, FSMs, covergroup items, toggles, and assertions within the instance or type selected in the *Verification Hierarchy* pane of the *Summary* page.

Figure 4-11 List tabs Pane



The *List tabs* pane has the following components:

- Tab Pages to list relative elements, covergroups, FSMs, covergroup items, toggles and assertions
- Recursive check box to display or hide all objects within the selected instance

Note: When you select the recursive check box, an additional column is added to the table to display the enclosing entity for the listed items.

4.3.2.1 Tab Pages

Using the different tab pages, you can list relative elements, covergroups, FSMs, covergroup items, and assertions within an instance or type selected in the *Verification Hierarchy* pane of the *Metrics* page. By default, the *Relative Elements* tab page is displayed.

To view a list of FSMs within instance:

1. Select an instance in the verification hierarchy pane.
2. Click the *FSMs* tab in the *List tabs* pane.

[Figure 4-12](#) on page 226 lists the FSMs in the selected instance.

Figure 4-12 List tabs—FSMs

The screenshot shows the 'List tabs' pane with the title 'tb_uart'. A legend at the top indicates that green bars represent 'Covered' metrics and yellow bars represent 'Not covered'. The table has columns: Index, Name, Overall Average Grade, Overall Covered, and Enclosing. There are three rows: 'wbstate' (75%, 6 / 8 (75%), tb_uart.u), 'tstate' (92.86%, 12 / 13 (... tb_uart.u)), and 'rstate' (90%, 23 / 26 (... tb_uart.u)). A callout arrow points from the text 'List of FSMs' to the first row.

Index	Name	Overall Average Grade	Overall Covered	Enclosing
	(no filter)	(no filter)	(no filter)	
	wbstate	75%	6 / 8 (75%)	tb_uart.u
	tstate	92.86%	12 / 13 (...	tb_uart.u
	rstate	90%	23 / 26 (...	tb_uart.u

Similarly, you can list covergroups, covergroup items, FSMs, and child instances for the selected instance or type.

Note: By default, the *List tabs* pane of the *Metrics* page shows Relative Elements, Cover groups, Items, FSMs, Assertions, and Toggle tab pages. If the configuration option *Show extend metrics tree* is set, then only Relative Elements, Bins, and Toggle tab pages are shown in the *List tabs* pane. For more details on this option, see [Configuring Appearance Options](#) on page 56.

4.3.3 Details Pane

The *Details* pane shows additional information, such as the metrics scored, attributes, and source, for the item last selected in the *Verification Hierarchy* pane or the *List tabs* pane. The *Details* pane has the following tabs:

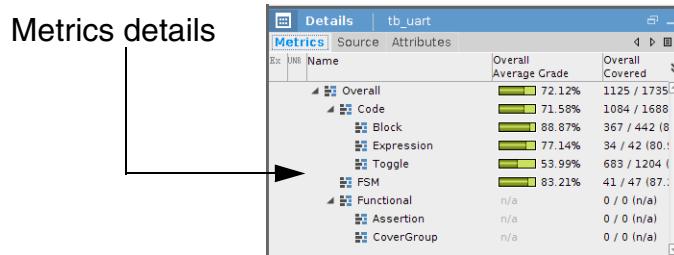
- [Metrics](#)
- [Attributes](#)
- [Source](#)

4.3.3.1 Metrics

The *Metrics* tab displays the overall average grade and overall covered details for each metric type relevant to the instance or type selected in the verification hierarchy pane.

[Figure 4-13](#) on page 227 displays the *Metrics* tab for the selected instance.

Figure 4-13 Details Pane—Metrics



The screenshot shows a table titled "Metrics" with columns "Name", "Overall Average Grade", and "Overall Covered". The data is organized into a hierarchical tree:

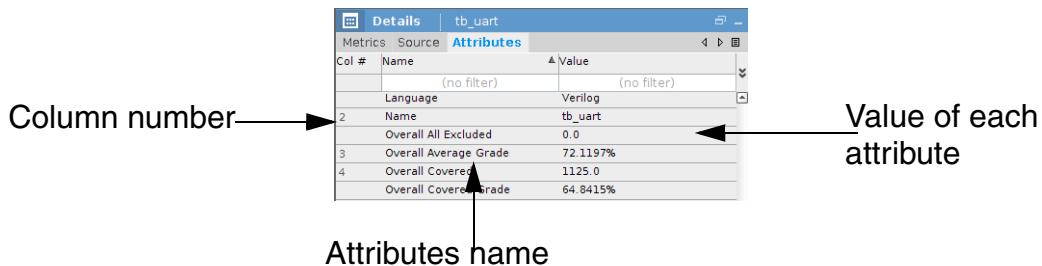
Name	Overall Average Grade	Overall Covered
Overall	72.12%	1125 / 1735
Code	71.58%	1084 / 1688
Block	88.87%	367 / 442 (8)
Expression	77.14%	34 / 42 (80%)
Toggle	53.99%	683 / 1204 (57%)
FSM	83.21%	41 / 47 (87%)
Functional	n/a	0 / 0 (n/a)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	n/a	0 / 0 (n/a)

The value *n/a* for a metric type indicates that either the metric was not scored for that instance or type, or items corresponding to that metric type were not included in the selected instance or type.

4.3.3.2 Attributes

The *Attributes* tab displays the characteristics recorded for the item selected in the *Verification Hierarchy* pane or the *List tabs* pane. [Figure 4-14](#) on page 227 displays the *Attributes* tab of selected instance.

Figure 4-14 Details Pane—Attributes



The screenshot shows a table titled "Attributes" with columns "Col #", "Name", and "Value". The data is as follows:

Col #	Name	Value
(no filter)	(no filter)	
2	Language	Verilog
2	Name	tb_uart
3	Overall All Excluded	0.0
3	Overall Average Grade	72.1197%
4	Overall Covered	1125.0
4	Overall Coverage Grade	64.8415%

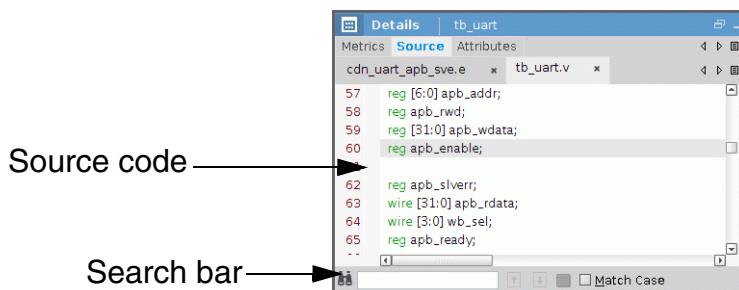
The attributes listed on the *Attributes* tab varies based on the selection made in the *Verification Hierarchy* pane or the *List tabs* pane.

For more details on attributes associated with each item, see IMC User Guide.

4.3.3.3 Source

The *Source* tab displays the actual source code for the item selected in verification hierarchy pane or the *List tabs* pane. [Figure 4-15](#) on page 228 displays the *Source* tab for FSM wbstate.

Figure 4-15 Details Pane—Source



The *Source* tab also displays the source of previously accessed items for easy access in different tabs. You can close the tab pages that show the source of previously accessed items by clicking the cross (X) symbol next to source file name. Using the search bar on the *Source* tab, you can quickly search for a specific text in the source code.

You can also right-click anywhere in the source code and select any of the following options:

- Copy —To copy the selected text.
- Copy Path—To copy full path of the source file to the system clipboard.
- Open in editor—To open the source file in the editor specified in the Configuration dialog box.

Note: In GUI, the *Source* tab can also show the source code from the compressed files provided the following is true:

- The compressed files are of extensions .gz or .Z.
- Each compressed file contains only one file.
- Compressed files are used at the time of coverage data generation.

Important

Currently, the batch mode of vManager cannot read the compressed files, and therefore, the *Source Code* column of the report of entities from the compressed files will be blank. This is also applicable for reports generated from GUI. The *Source Code* column of reports generated from GUI will be blank for entities from the compressed files.

4.4 Analyzing Code Coverage

Code coverage measures how thoroughly a testbench exercises the lines of HDL code that describe a design. The coverage results reveal areas of the design that have not been fully tested, or that did not meet desired coverage criteria. Code coverage analysis includes:

- Block coverage
- Expression coverage
- Toggle coverage

For more details, see the [Incisive Metrics Center User Guide](#).

4.5 Analyzing FSM Coverage

FSM coverage interprets the synthesis semantics of the HDL design and monitors the coverage of the FSM representation of control logic blocks in the design. It answers the question “Did I reach all of the states and cover all possible transitions or arcs in a given state machine?”

To analyze FSM coverage:

1. Navigate through the design hierarchy on the *Analysis - Summary* page and identify overall coverage of different state machines in the loaded run.
2. Launch the *FSM* page to perform a detailed analysis of different state machines.

For more details, see the [Incisive Metrics Center User Guide](#).

4.6 Analyzing Functional Coverage

Functional coverage is generated by inserting PSL, SystemVerilog assertions, or SystemVerilog covergroup statements into the code and simulating the design. vManager also supports analysis of functional coverage defined in e language and collected by Specman. The functional coverage points specify scenarios, error cases, corner cases, and protocols to be covered and also specifies analysis to be done on different values of a variable.

Functional coverage is of the following types:

- Assertion coverage—Is an extension of assertion-based verification and identifies interesting functions directly. Assertion coverage points are specified using PSL or SVA

assert, assume, and cover directives. The coverage to be measured is directly specified using the PSL/SVA statements or is interpreted from them.

- Covergroup coverage—Focuses on tracking data values. It includes coverage of variable values, binning, specification of sampling, and cross products. It helps design engineers to identify untested data values or subranges. Covergroup coverage is specified using SystemVerilog constructs.

Using vManager, you can analyze both assertion as well as covergroup coverage.

For more details, see the [Incisive Metrics Center User Guide](#).

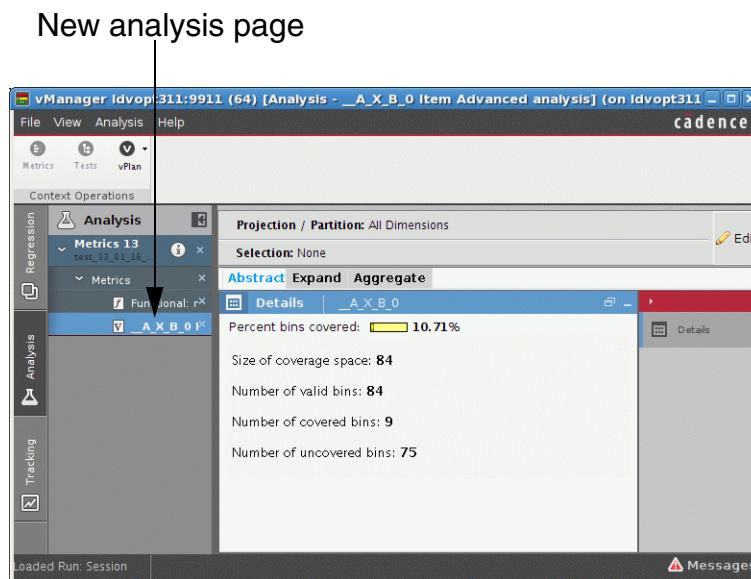
Advanced Item Analysis of Crosses

To perform advanced item analysis of crosses, do any of the following:

- Select the cross, right-click and select *Item Advanced analysis*.
- Select the cross and select *Item Advanced* in the *Analyze* toolbar.

This will open a new analysis page, as shown in [Figure 4-16](#) on page 230.

Figure 4-16 Item Advanced Analysis



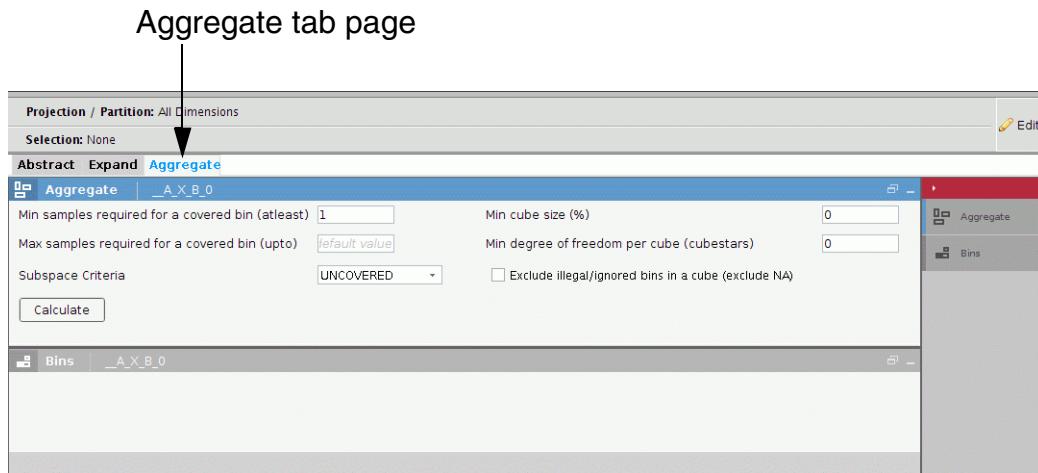
This page has following tab pages:

- The *Abstract* tab page shows the high-level summary of the selected item. For more details, see the [Incisive Metrics Center User Guide](#).

- The *Expand* tab page shows the list of bins and their detailed information. For more details, see the [Incisive Metrics Center User Guide](#).
- The *Aggregate* tab page enables generating aggregated results based on certain parameters.

[Figure 4-17](#) on page 231 shows the *Aggregate* tab page.

Figure 4-17 Item Advanced Analysis (Aggregate)



In the *Aggregate* tab page, you can set the following options:

- *Min samples required for a covered bin (atleast)*—Set the minimum number of samples required for a bin to be considered covered.
- *Max samples required for a covered bin (upto)*—Set the maximum number of samples required for a bin to be considered covered.
- *Subspace Criteria*—Specify the criteria on which the subspace creation must be based. It can be either COVERED or UNCOVERED.
- *Min cube size (%)*—Set the minimum size of reported cubes. It is measured in % of physical space size.
- *Min degree of freedom per cube (cubestar)*—Set the minimum number of stars in the reported cubes.
- *Exclude illegal/ignored bins in a cube (exclude NA)*—If selected, the NA bins are not included in reported holes/covered cubes.

After specifying the required options, click the *Calculate* button.

Figure 4-18 on page 232 shows the *Aggregate* tab page with aggregated results.

Figure 4-18 Item Advanced Analysis (Aggregated Results)

The screenshot shows the 'Aggregate' tab page. At the top, there are settings for 'Min samples required for a covered bin (atleast)' (1), 'Max samples required for a covered bin (upto)' (default value), 'Min cube size (%)' (0), 'Min degree of freedom per cube (cubestars)' (0), and 'Subspace Criteria' (UNCOVERED). There is also a checkbox for 'Exclude illegal/ignored bins in a cube (exclude NA)'. A 'Calculate' button is present. Below this is a table titled 'Bins' with columns A, B, Size, Physical Size, and Score. The table lists various cross bins and their properties. An annotation 'Aggregated results' with an arrow points to the table.

A	B	Size	Physical Size	Score
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
{Ab1[0],Ab1[3],Ab1[5],Ab1[6],Ab1[10],Ab...*}	{Bb1[0],Bb1[2],Bb1[5]}	48	48	0
{Ab1[7],Ab1[9]}	{Bb1[0],Bb1[5]}	10	10	0
{Ab1[1],Ab1[4],Ab1[7],Ab1[9]}	{Bb1[2],Bb1[3]}	8	8	0
{Ab1[4],Ab1[7],Ab1[9]}	{Bb1[0]}	8	8	0
{Ab1[1],Ab1[2],Ab1[4],Ab1[7],Ab1[9]}	{Bb1[4]}	5	5	0
{Ab1[1],Ab1[2],Ab1[7],Ab1[9]}	{Bb1[4]}	5	5	0

In the aggregated results, the similar bins are grouped together (to create cubes).

In the above results:

- Each row indicates the group of similar cross bins.
- Columns *A* and *B* indicate the coverpoints participating in the cross.

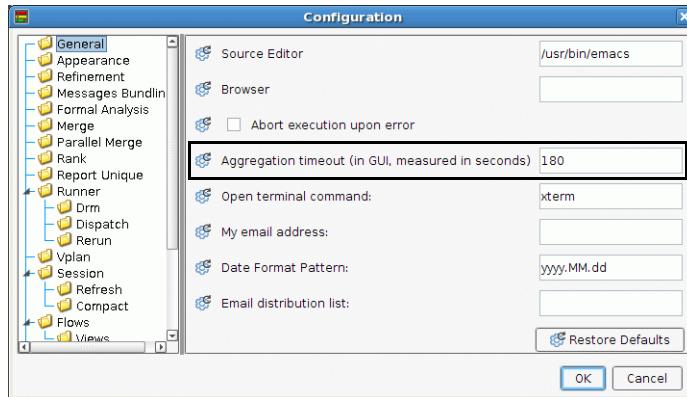
Note: These columns might sometimes show a * to indicate all values of that coverpoint. When you place the cursor over *, bins of that coverpoint (that are associated with *) are shown.

- The *Size* column shows the number of cross bins grouped together.
- The *Physical Size* column shows the physical space size taken by each group.
- The *Score* column shows the number of times that group was hit.

The aggregated results enable you to quickly analyze bins. The calculation of aggregated results might increase computational overhead; however, it makes data analysis more efficient and less time-consuming.

As the calculation of aggregated results might increase the operational overhead, you can timeout the calculation after the specified time has elapsed. You can specify the timeout using the *Aggregation timeout* configuration option, as shown in Figure 4-19 on page 233.

Figure 4-19 Aggregation Timeout Option



By default, the aggregation calculation is stopped after 180 seconds. However, you can change this limit, as required.

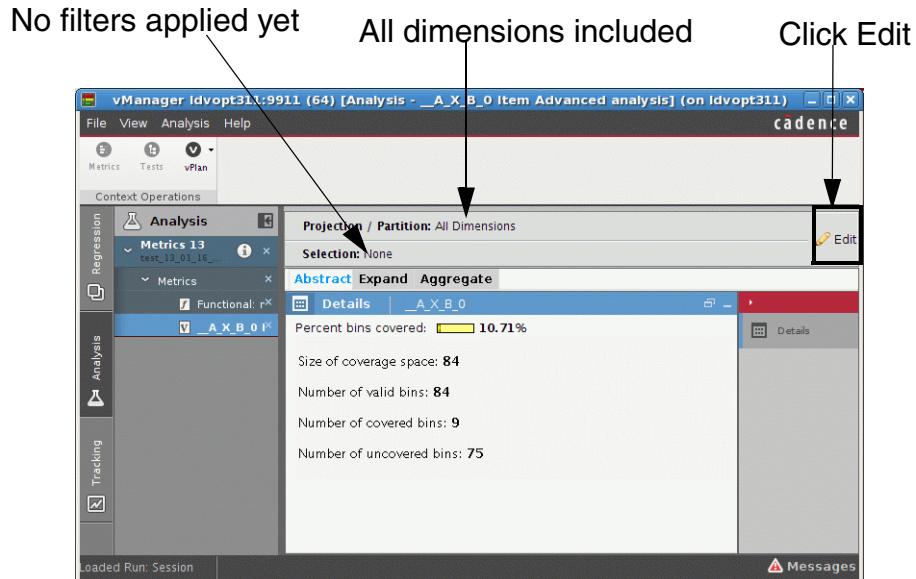
Applying Projections and Selection on Cross Items

By default, all dimensions are projected and no filters are applied on the cross items. vManager allows you to edit projections (select dimensions for data projection) and also filter out bins (selections) for data presented in the *Abstract*, *Expand*, and *Aggregate* tab pages.

To modify projections and selections, perform the following steps:

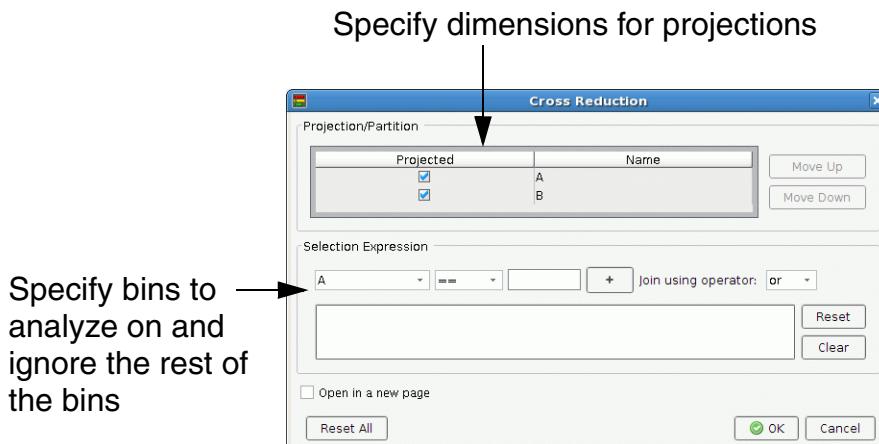
1. Click the *Edit* button, as shown in Figure 4-20 on page 234.

Figure 4-20 Applying Projections and Selections



This opens the *Cross Reduction* dialog box, as shown in [Figure 4-21](#) on page 234.

Figure 4-21 Cross Reduction

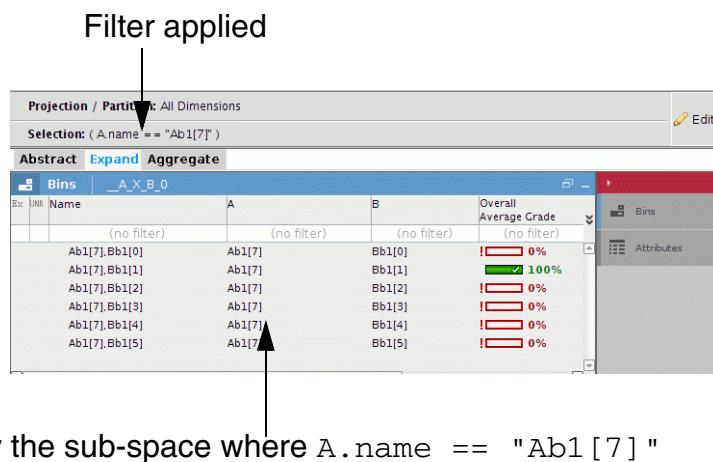


2. In the *Projection/Partition* section, the projected items are listed. If required, you can select or deselect the items from analysis.
3. In the *Selection Expression* section, build a query such that it shows a sub-space for analysis. For example, to include only bins with `A1.name` is `Ab1[7]`, specify filter query as `(A1.name == "Ab1[7]")`. Only bins that meet filter criteria are shown and rest of the bins are removed from the space and are not considered as part of valid space of the reported item.

4. In case you want to open a separate page to show the results after applying projections and selections, select the *Open in a new page* check box.
5. Click *OK*.

After you click *OK*, the results shown in the different tab pages are recalculated. For example, in this case, the *Expand* tab page now only shows the sub-space where `A.name == "Ab1[7]"`, as shown in [Figure 4-22](#) on page 235.

Figure 4-22 Projections/Selections Applied



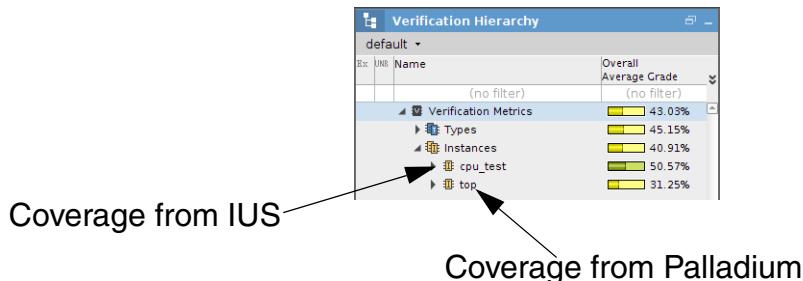
The data in *Abstract* tab page and *Aggregate* tab page is also recalculated.

Note: Projection and selection does not modify the item itself, but it creates an on the fly item, based on your inputs in the *Cross Reduction* dialog box.

4.7 Analyzing Coverage Coming from Palladium

By default, coverage coming from Palladium is not merged with coverage coming from IUS. It is listed side-by-side in the verification hierarchy pane, as shown in [Figure 4-23](#) on page 236.

Figure 4-23 Coverage From IUS and Palladium



In case you want to see the coverage as a merged output from IUS and Palladium, perform the following steps:

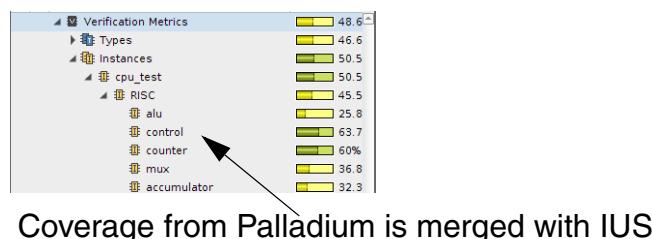
1. Merge the coverage coming from Palladium with coverage coming from IUS using the `-cross_domain` option of the `merge` command of IMC. For more details on the usage of this option, see the [Incisive Metrics Center User Guide](#).

Note: Alternatively, you can enable the *ICC-UXE cross domain* option in the *Configuration* dialog box. After you set this option, any merge operation done in vManager (no matter if it comes from collect runs / analyze session coverage / compaction / runs come from same session or different sessions) will merge ICC and UXE coverage together. For more details, see [Configuring Merge Options](#) on page 67.

2. After merging the results, open vManager.
3. Perform the *Collect Runs* action on the coverage merged in Step 1.
4. A new session is created. Right-click the session and select *Analyze Metrics*.

You will now see the merged results, as shown in [Figure 4-24](#) on page 236.

Figure 4-24 Coverage From Palladium Merged with IUS Coverage



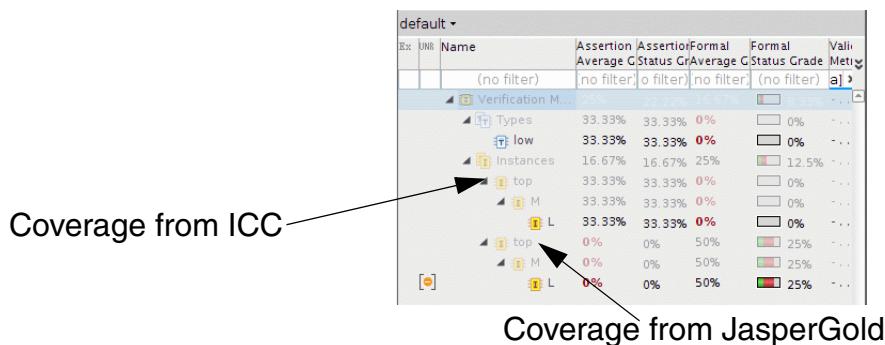
Note: Only block, toggle, and assertion coverage is merged from the UXE run to ICC run.

For more details, see the [Incisive Metrics Center User Guide](#).

4.8 Analyzing Coverage Coming from JasperGold

By default, coverage coming from JasperGold is not merged with coverage coming from IUS. It is listed side-by-side in the verification hierarchy pane, as shown in [Figure 4-25](#) on page 237.

Figure 4-25 Coverage From IUS and JasperGold



In case you want to see the coverage as a merged output from IUS and JasperGold, do any of the following:

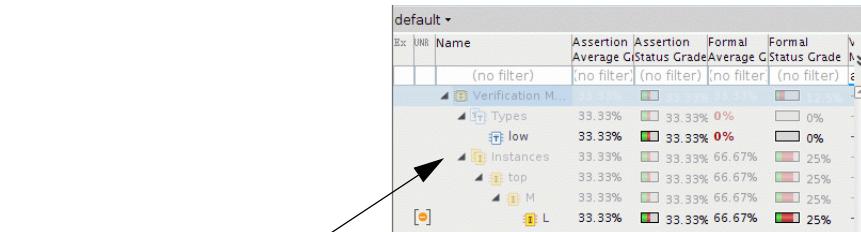
1. Merge the coverage coming from JasperGold with coverage coming from IUS using the `-cross_domain` option of the `merge` command of IMC. For more details on the usage of this option, see the [Incisive Metrics Center User Guide](#).

Note: Alternatively, you can enable the *ICC-JG cross domain* option in the *Configuration* dialog box. After you set this option, any merge operation done in vManager (no matter if it comes from collect runs / analyze session coverage / compaction / runs come from same session or different sessions) will merge ICC and JG coverage together. For more details, see [Configuring Merge Options](#) on page 67.

2. After merging the results, open vManager.
3. Perform the *Collect Runs* action on the coverage merged in Step a.
4. A new session is created. Right-click the session and select *Analyze Metrics*.

You will now see the merged results, as shown in [Figure 4-26](#) on page 238.

Figure 4-26 Coverage From JasperGold Merged with IUS Coverage



Coverage from JasperGold is merged with IUS

For more details, see the [Incisive Metrics Center User Guide](#).

4.9 Refining Metrics Data

While analyzing metrics data, you may want to exclude certain undesired instances, types, and items from coverage analysis. The process of excluding certain items from coverage analysis is called refinement.

The *Refinement* toolbar contains the options that allow you to exclude instances or types, un-exclude already excluded items, save refinements, load refinements and show already excluded items. You can choose the following items from the *Refinement* toolbar:

- Exclude—To exclude the selected item and its children
- Exclude Local Aspects only—To exclude only top-level instances
- Exclude Smart — To exclude such that any entity connected to the excluded entity will be excluded implicitly.
- Show Affected — To show the list of entities that were implicitly marked excluded because of the *Exclude Smart* action on that entity.
- Show Affecting — To show the list of entities that caused the indirect smart exclusion.
- Edit comment—To edit exclusion comment.
- Un-Exclude—To un-exclude already excluded items.
- Clear dirty rules —To clear/delete the dirty rules. This option is available recursively from the type-instance/fsm parent entity. It is also available from the dirty excluded entity itself.
- Approve dirty rules — To approve the dirty rules (make the rule completely valid). This option is available recursively from the type-instance/fsm parent entity. It is also available from the dirty excluded entity itself.

- Clear all orphan rules—To clear/delete all the orphan rules of the selected item. This option is available recursively from the type-instance/fsm parent entity.
- Apply all orphan rules —To apply all the orphan rules of the selected item (by index of originally excluded entity) and clear them. This option is available recursively from the type-instance/fsm parent entity.
- Read Refine—To load already saved exclusion file
- Save Refine—To save the exclusions to a file for later use

Note: All the above options are also available in the *Analysis* menu.

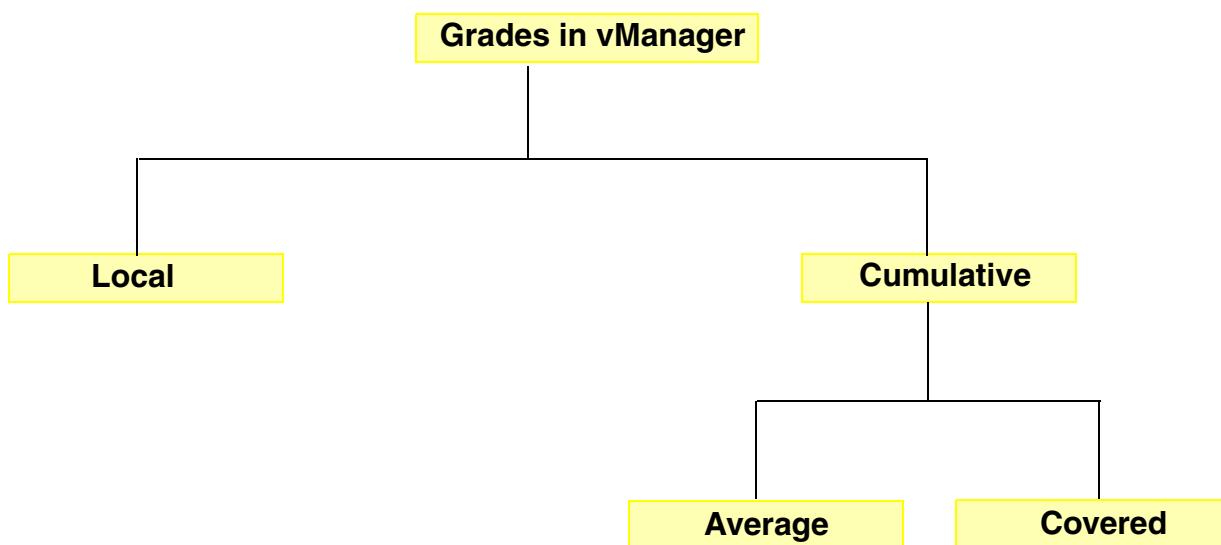
Note: Smart exclusion is not supported for vPlan refinements.

For more details, see the [Incisive Metrics Center User Guide](#).

4.10 Coverage Grading

[Figure 4-27](#) on page 239 shows the different grades calculated by vManager.

Figure 4-27 Grades in vManager



For each metrics type, vManager calculates the following grades:

- Local Grade: Local grade (also known as self coverage) is the coverage of that instance or module without considering the coverage of child instances or modules of that instance or module.

- Cumulative Grade: Cumulative grade is computed by combining the self coverage and the cumulative grade of its children, if any. Cumulative grade is of following types:
 - Cumulative Average Grade
 - Cumulative Covered Grade

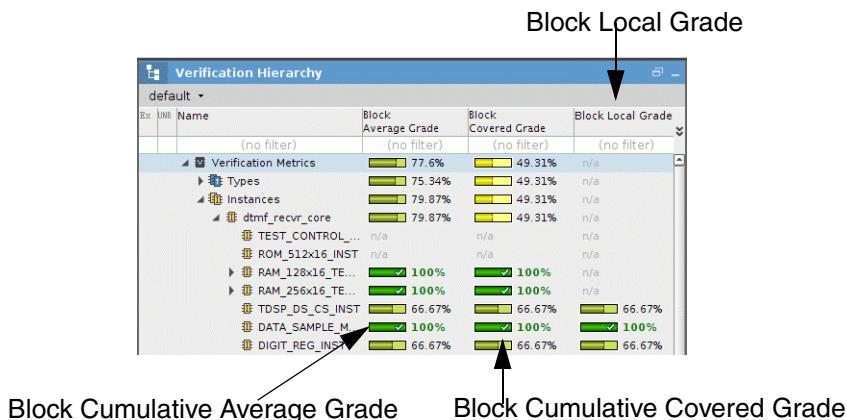
For example, for block metrics, following grades are calculated by vManager:

- Block Local Grade
- Block Average Grade
- Block Covered Grade

In vManager GUI, these grades are shown as columns/attributes with the same name as stated above.

[Figure 4-28 on page 240](#) shows the grades columns related to block metrics in vManager GUI.

Figure 4-28 Block Local and Cumulative Grades in GUI Mode

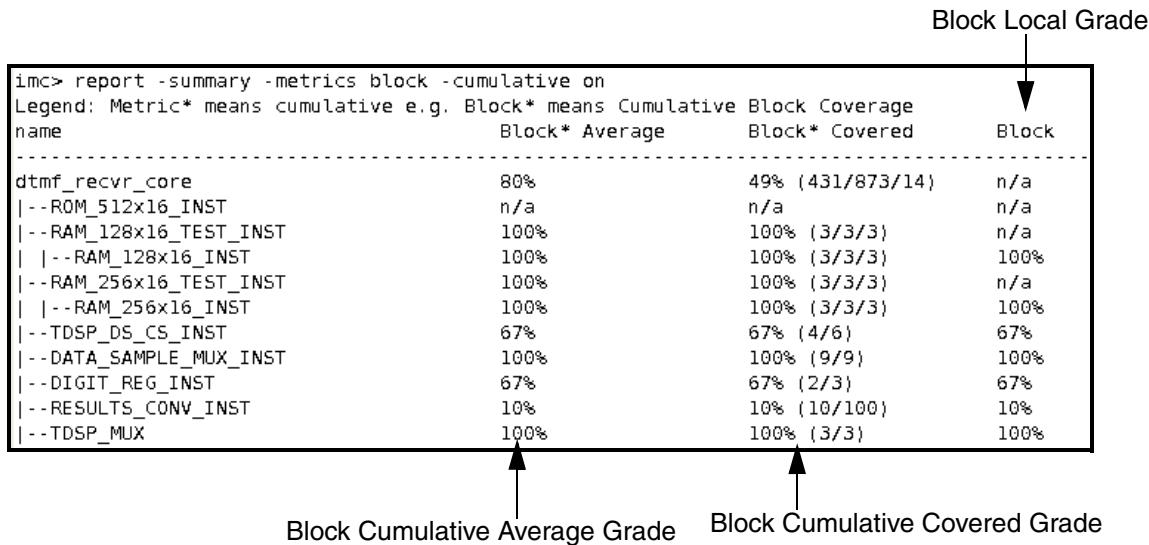


In batch mode, these grades are shown when you generate a summary report. For example, consider the following command:

```
report -summary -metrics block -cumulative on
```

[Figure 4-29 on page 241](#) shows the output of the above command.

Figure 4-29 Block Local and Cumulative Grades in Batch Mode



name	Block*	Average	Block* Covered	Block
dtmf_recv_r_core	80%	49% (431/873/14)	n/a	n/a
--ROM_512x16_INST	n/a	n/a	n/a	n/a
--RAM_128x16_TEST_INST	100%	100% (3/3/3)	n/a	n/a
--RAM_128x16_INST	100%	100% (3/3/3)	100%	100%
--RAM_256x16_TEST_INST	100%	100% (3/3/3)	n/a	n/a
--RAM_256x16_INST	100%	100% (3/3/3)	100%	100%
--TDSP_DS_CS_INST	67%	67% (4/6)	67%	67%
--DATA_SAMPLE_MUX_INST	100%	100% (9/9)	100%	100%
--DIGIT_REG_INST	67%	67% (2/3)	67%	67%
--RESULTS_CONV_INST	10%	10% (10/100)	10%	10%
--TDSP_MUX	100%	100% (3/3)	100%	100%

In the above report, * in the column name indicates cumulative grade. Cumulative grades are printed due to the use of `-cumulative` option in the `report` command. In the absence of this option, only local grade is printed.

4.10.1 Grade Calculation in vManager

This section discusses the following grade calculations for each metrics type in vManager:

- [Local Grade Calculation](#) on page 241
- [Cumulative Grade Calculation](#) on page 248

4.10.1.1 Local Grade Calculation

This section discusses local grade calculation for different metrics types.

Block Coverage

vManager uses the following formula to calculate block local grade:

$$\text{Block local grade} = \frac{\text{Number of covered blocks}}{\text{Total number of blocks}} \times 100\%$$

Note: In vManager GUI, the block local grade is printed in the *Block Local Grade* column. In batch mode, block local grade is printed when you use the `report -summary -metrics block` command.

Branch Coverage

Branch coverage is a subtype of block coverage and provides more precise coverage results by reporting coverage for each branch individually. vManager uses the following formula to calculate branch local grade:

$$\text{Branch local grade} = \frac{\text{Number of covered branches}}{\text{Total number of branches}} \times 100\%$$

Note: By default, branch coverage is not enabled. It is enabled using the `set_branch_scoring` command. See *ICC User Guide* for more details on this command. If branch coverage is enabled while scoring coverage, it is automatically reported when you generate a block summary report.

Statement Coverage

Statement coverage is a subtype of block coverage which provides information on the number of statements within a block. vManager uses the following formula to calculate statement local grade:

$$\text{Statement local grade} = \frac{\text{Number of statements in covered blocks}}{\text{Total number of statements in all blocks}} \times 100\%$$

Note: By default, statement coverage is not enabled. It is enabled using the `set_statement_scoring` command. See *ICC User Guide* for more details on this command. If statement coverage is enabled while scoring coverage, it is automatically reported when you generate a block summary report.

Expression Coverage

Expression coverage is calculated based on total number of min term rows in all of the expressions in an instance/module and the number of min term rows that are covered.

vManager uses the following formula to calculate the expression local grade:

$$\text{Expression local grade} = \frac{\text{Number of covered min term rows}}{\text{Total number of min term rows}} \times 100\%$$

In vManager, expression coverage is displayed at the following levels:

- Instance/Module
- Top expression
- Sub-expression
- Expression coverage table

For any given level, the total number of min term rows below that level is considered. For example, at the top expression level, min term rows of all the sub-expressions in the hierarchy of that expression are considered while calculating expression coverage of that top expression.

Note: In vManager GUI, the expression local grade is printed in the *Expression Local Grade* column. In batch mode, expression local grade is printed when you use the `report -summary -metrics expression` command.

Toggle Coverage

Toggle coverage is calculated based on total number of signal bits in an instance/module and the number of fully toggled signal bits. A signal bit is considered fully toggled signal if it has gone from 1 to 0 and also from 0 to 1.

vManager uses the following formula to calculate toggle local grade:

$$\text{Toggle local grade} = \frac{\text{Number of fully toggled signal bits}}{\text{Total number of signal bits}} \times 100\%$$

Note: In vManager GUI, the toggle local grade is printed in the *Toggle Local Grade* column. In batch mode, toggle local grade is printed when you use the `report -summary -metrics toggle` command.

State Coverage

vManager uses the following formula to calculate state local grade:

$$\text{State local grade} = \frac{\text{Number of visited states}}{\text{Total number of states}} \times 100\%$$

State coverage is calculated at two levels, instance/module level, and individual FSM level. While calculating state coverage at instance/module level, states of all FSMs in that instance/module are taken into consideration.

Note: In vManager GUI, the state local grade is printed in the *State Local Grade* column. In batch mode, state local grade is printed when you use the `report -summary -metrics state` command.

Transition Coverage

A transition means change of state from a start state to an end state. vManager uses the following formula to calculate transition local coverage:

$$\text{Transition local grade} = \frac{\text{Number of executed transitions}}{\text{Total number of transitions}} \times 100\%$$

Transition coverage is calculated at two levels, instance/module level, and individual FSM level. While calculating transition coverage at instance/module level, transitions of all FSMs in that instance/module are taken into consideration.

Note: In vManager GUI, the transition local grade is printed in the *Transition Local Grade* column. In batch mode, transition local grade is printed when you use the `report -summary -metrics transition` command.

Arc Coverage

Arc coverage is a subtype of transition coverage that reports all the input conditions under which a transition can take place. vManager uses the following formula to calculate arc local grade:

$$\text{Arc local grade} = \frac{\text{Number of executed arcs}}{\text{Total number of arcs}} \times 100\%$$

Arc coverage is calculated at two levels, instance/module level, and individual FSM level. While calculating arc coverage at instance/module level, the arcs of all FSMs in that instance/module are taken into consideration.

Note: If arc coverage is disabled for a particular FSM, then transition coverage is used as arc coverage for that FSM. By default, arc coverage is not enabled. It is enabled using the `set_fsm_arc_scoring` command. See *ICC User Guide* for more details on this command.

Note: In vManager GUI, the arc local grade is printed in the *Arc Local Grade* column. In batch mode, arc local grade is printed when you use the `report -summary -metrics arc` command.

Assertion Coverage

vManager uses the following formula to calculate assertion local grade:

$$\text{Assertion local grade} = \frac{\text{Number of finished assertions}}{\text{Total number of assertions}} \times 100\%$$

Note: In vManager GUI, the assertion local grade is printed in the *Assertion Local Grade* column. In batch mode, assertion local grade is printed when you use the `report -summary -metrics assertion` command.

Covergroup Coverage

A covergroup consists of one or more cover items (coverpoints or crosses) and a cover item includes one or more cover bins.

vManager uses the following formula to calculate coverage of a cover item:

$$\text{Cover item Covered coverage } (C_i) = \frac{\text{Number of covered cover bins}}{\text{Total number of cover bins}} \times 100\%$$

If $\left\{ \frac{\text{Number of covered cover bins}}{\text{Total number of cover bins}} \times 100 = 100\% \right\}$

$$\text{Cover item Average coverage } (C_i) = 100\%$$

else:

$$\text{Cover item Average coverage } (C_i) = \frac{\text{Number of covered cover bins}}{\text{Total number of cover bins} \times \text{goal}/100}$$

In the SystemVerilog language, a cover bin is considered covered if its hit count \geq the `at_least` count.

In the e language, a cover bin is considered:

- Fully covered—if its hit count \geq `at_least` count
- Partially covered—if its hit count is a non-zero value less than the `at_least` count

If a cover bin is partially covered, then the cover item coverage is calculated by replacing Number of covered bins in the above formula by hit count/at_least count.

vManager uses the following formula to calculate average coverage of a covergroup:

$$\text{Covergroup average coverage } (C_g) = \frac{\sum C_i \times W_i}{\sum W_i \times CGg / 100}$$

In the above formula, C_i and W_i are the coverage and weight, respectively, of cover items in the covergroup C_g and CGg is the coverage goal value of the Covergroup.

vManager uses the following formula to calculate covergroup average coverage of an instance/module:

$$\text{Covergroup average coverage of instance/module} = \frac{\sum C_g \times W_g}{\sum W_g \times IMg/100}$$

In the above formula, C_g and W_g are the coverage and weight, respectively, of covergroups in the instance/module and IMg is the goal value of the instance/module. For details on the at_least count and weight of cover items, see the *ICC User Guide*.

vManager uses the following formula to calculate the *Covergroup Covered* grade:

$$\text{Covergroup Covered coverage} = \frac{\text{Number of covered cover bins}}{\text{Total number of cover bins}} \times 100\%$$

In the above formula, Total number of cover bins is the number of cover bins scored that have not been excluded.

Note: vManager does not consider weight/goal while calculating Number of covered cover bins and Total number of cover bins. The weight value affects only the average grade except when the weight is 0, which means exclude.

Note: In vManager GUI, the covergroup local grade is printed in the *CoverGroup Local Grade* column. In batch mode, covergroup local grade is printed when you use the `report -summary -metrics covergroup` command.

Code Coverage

Code coverage is computed by combining block coverage, expression coverage, and toggle coverage. vManager uses the following formula to calculate code local grade:

$$\text{Code local grade} = \frac{\text{Block coverage} + \text{Expression coverage} + \text{Toggle coverage}}{3}$$

Note: In vManager GUI, the code local grade is printed in the *Code Local Grade* column. In batch mode, code local grade is printed when you use the `report -summary -metrics code` command.

FSM Coverage

FSM coverage is computed by combining state coverage and transition coverage. vManager uses the following formula to calculate FSM local grade:

$$\text{FSM local grade} = \frac{\text{State coverage} + \text{Transition coverage}}{2}$$

Note: If arc coverage is enabled, then it is used in the above computation instead of transition coverage.

Note: In vManager GUI, the FSM local grade is printed in the *FSM Local Grade* column. In batch mode, FSM local grade is printed when you use the `report -summary -metrics fsm` command.

Functional Coverage

Functional coverage is computed by combining covergroup coverage and assertion coverage. vManager uses the following formula to calculate functional coverage:

$$\text{Functional local grade} = \frac{\text{Covergroup coverage} + \text{Assertion coverage}}{2}$$

Note: In vManager GUI, the functional local grade is printed in the *Functional Local Grade* column. In batch mode, functional local grade is printed when you use the `report -summary -metrics functional` command.

Overall Coverage

Overall coverage is computed by combining code coverage, FSM coverage, and functional coverage. vManager uses the following formula to calculate overall coverage:

$$\text{Overall local grade} = \frac{\text{Code coverage} + \text{FSM coverage} + \text{Functional coverage}}{3}$$

Note: In vManager GUI, the overall local grade is printed in the *Overall Local Grade* column. In batch mode, overall local grade is printed when you use the `report -summary -metrics overall` command.

4.10.1.2 Cumulative Grade Calculation

Cumulative grade is computed by combining the self metrics of that instance/type and the cumulative metrics of its children, if any. It is calculated by averaging self metric and cumulative metric of all direct children.

The process by which two or more grades are combined is known as roll-up. vManager provides following grade roll-up methods:

- Average Rollup
- Covered Rollup

Average Rollup

The *Average* rollup method is used to calculate the Cumulative Average Grade of a metrics type. vManager uses the following formula to calculate cumulative average grade:

$$\text{Cumulative Average Grade} = \frac{M_{\text{self}} + \sum M_{\text{cum}}}{1 + \text{Number of immediate child instances}}$$

where M_{self} is the self metric (taking into consideration goal value, if defined) and M_{cum} is the cumulative metric of each of the immediate children while taking into consideration each weight and goal values.

Note: *Average* rollup method was earlier known as the *Top Elements* scheme.

Covered Rollup

The *Covered* rollup method is used to calculate the Cumulative Covered Grade of a metrics type. vManager uses the following formula to calculate cumulative covered grade:

$$\text{Cumulative Covered Grade} = \frac{\sum_{\text{Me_covered}}}{\sum_{\text{Me_total}}} + \frac{\text{Descendents covered (i)}}{\text{Descendents total (i)}}$$

where:

- Covered coverage of item i ($\text{covered}(i)$) is calculated as:

$$\text{covered}(i) = \sum_{\text{Children}} \text{Covered_bin}(i)$$

Here, $\text{covered_bin}(i)$ is the number of covered bins of item i .

- Total coverage of item i ($\text{total}(i)$) is calculated as:

$$\text{total}(i) = \sum_{\text{Children}} \text{bin}(i)$$

Here, $\text{bin}(i)$ is the number of bins of element i .

vManager does not consider weight while calculating covered grades.

Covered rollup method was earlier known as the *All Buckets* scheme.

4.11 Generating Metrics Report

Using vManager you can generate metrics report in an HTML format and then publish it on the internet. Generating a report in an HTML format is useful for users who want to make the results available for viewing and analysis to people who might not have vManager installed with them.

The *Report* toolbar has a drop-down button named *Report* using which you can generate reports from GUI. The *Reports* option is also available in the *Analysis* menu using which you can generate reports.

The *Report* drop-down button on the *Report* toolbar has the following options for generating reports:

- All (Default) —To generate a metrics report for the complete hierarchy.

- On Selected Entity—To generate a sessions report for a selected entity.
- Summary Report—To generate a summary report. For more details, see [Generating Summary Report](#) on page 51.

Note: Only HTML reports can be generated from GUI. You cannot generate ASCII reports from GUI.

For more details, see the [Incisive Metrics Center User Guide](#).

Analyzing Verification Plans

Using vManager, you can analyze verification plans in detail.

This chapter covers the following topics:

- [Loading Verification Plans](#) on page 251
- [Creating a New vPlan](#) on page 258
- [Editing vPlans](#) on page 259
- [Organizing and Defining vPlan Views](#) on page 261
- [Computing Grade of a Section](#) on page 264
- [vPlan Configuration Options](#) on page 270
- [Looking for a Section or Port](#) on page 283
- [Filtering Items in vPlan Hierarchy](#) on page 288
- [Launching Detailed Analysis Page for vPlan Entity](#) on page 289
- [Refining vPlan Items](#) on page 290
- [Generating vPlan Reports](#) on page 309
- [vPlan Visitor Interface](#) on page 316

5.1 Loading Verification Plans

You can load verification plans from the *Regression* center or the *Analysis* center. This section covers the following topics:

- [Loading vPlan from Regression Center](#)
- [Loading vPlan from Analysis Center](#)

Note: Any data written in Japanese Kanji (coming from vPlan, metrics, or tests), can be displayed in vManager.

5.1.1 Loading vPlan from Regression Center

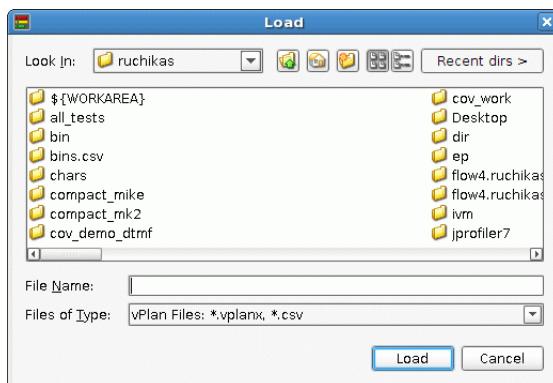
To load a vPlan from the *Regression* center:

1. Select a session(s).
2. Click *vPlan* button in the *Analyze* toolbar.

Note: Alternatively, you can select *Analyze vPlan* from the *Regression* menu.

The *Load* dialog box is displayed, as shown in [Figure 5-1](#) on page 252.

Figure 5-1 Load vPlan File

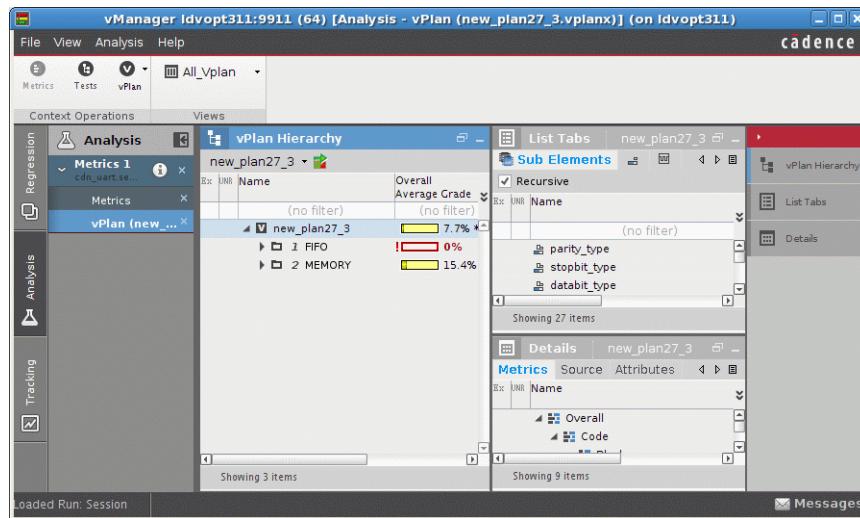


3. Navigate through the hierarchy and select the relevant vPlan file. For example, to open the vPlan file named `new_plan27_3.vplanx` select `new_plan27_3.vplanx` and click *Load*.

Note: By default, the dialog box shows the last directory you visited. In order to navigate to a different directory, click the *Recent dirs* button. If you navigate to a new directory, it is saved automatically under the list of quick links and becomes the new default.

The *Analysis* center is invoked with the selected vPlan loaded, as shown in [Figure 5-2](#) on page 253.

Figure 5-2 Analysis Center with vPlan Loaded



The selected vPlan is loaded. You can now analyze the vPlan.

Note: vPlan context will show disabled if the underlying coverage model is extremely large.

5.1.2 Loading vPlan from Analysis Center

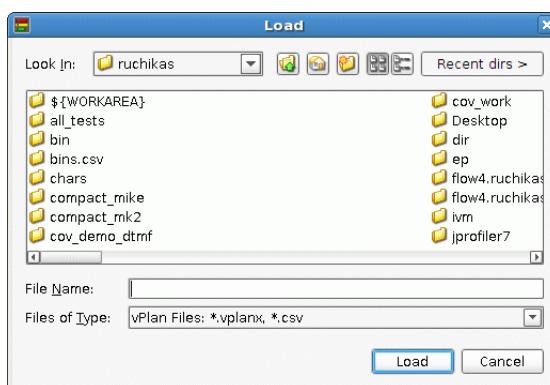
If you are already in the *Analysis* center, perform the following steps to load a vPlan:

1. Click the *vPlan* button in the *Context Operations* toolbar.

Note: Alternatively, you can select *Analyze vPlan* from the *Analysis* menu.

The *Load* dialog box is displayed, as shown in Figure 5-3 on page 253.

Figure 5-3 Load vPlan File

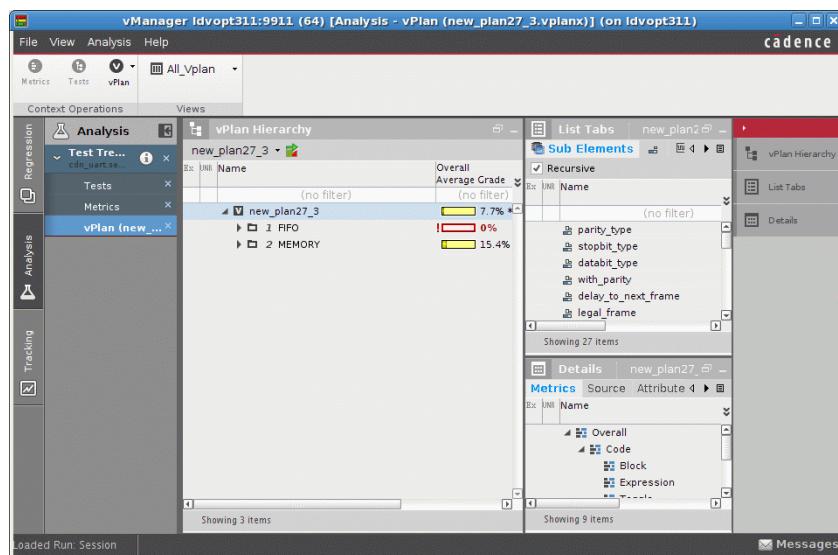


2. Navigate through the hierarchy and select the relevant vPlan file. For example, to open the vPlan file named `new_plan27_3.vplanx` select `new_plan27_3.vplanx` and click `Load`.

Note: By default, the dialog box shows the last directory you visited. In order to navigate to a different directory, click the *Recent dirs* button. If you navigate to a new directory, it is saved automatically under the list of quick links and becomes the new default.

The selected vPlan file is loaded, as shown in [Figure 5-4](#) on page 254.

Figure 5-4 Loaded vPlan



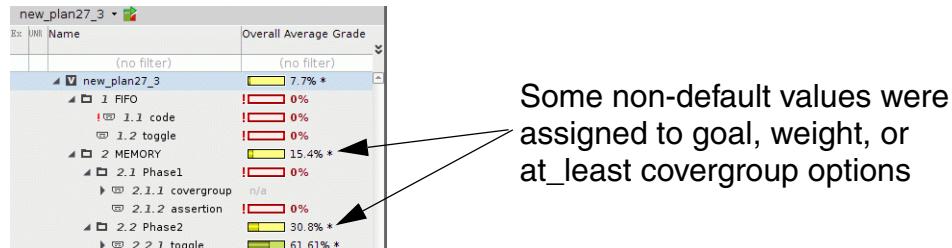
After the vPlan file is loaded, you can navigate through the vPlan in the *vPlan Hierarchy* pane.

In the *vPlan Hierarchy* pane, if you select a metric port, then the *Sub Elements* tab page of *List tabs* pane shows the elements within the selected port.

If you select a cover item in the *vPlan Hierarchy* pane, then the *Bins* tab of the *List tabs* pane becomes active and shows the bins within the selected cover item.

Note: Sometimes, you may notice a * appearing next to the grade in the *Overall Average Grade* column, as shown in [Figure 5-5](#) on page 255.

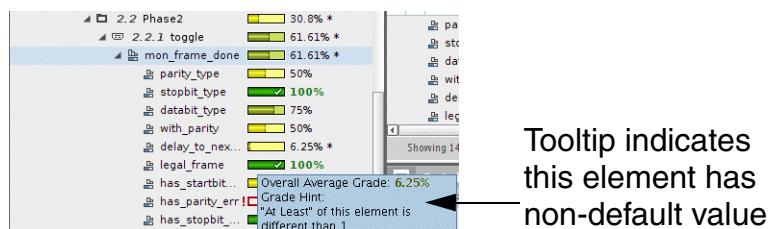
Figure 5-5 Use of Non-default Values Represented by *



A * indicates that some non-default values were assigned to the weight (default is 1), goal (default is 100), or the at_least (default is 1) covergroup options. When you place the mouse over the *Overall Average Grade* value, a tooltip appears showing the options that affected the grade calculation. The tooltip also indicates whether the options affecting the grade are related to the current entity or in one of its descendant.

[Figure 5-6 on page 255](#) shows the tooltip that appears.

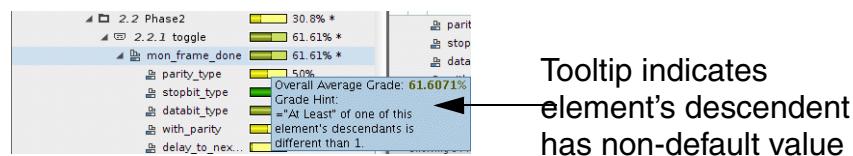
Figure 5-6 Tooltip to Indicate Options Affecting Grade Calculation



The tooltip shown indicates that non-default value was assigned to the at_least covergroup option of this selected element.

If you place the mouse over the parent item, then the tooltip, as shown in [Figure 5-7 on page 255](#) is displayed.

Figure 5-7 Tooltip to Indicate Options Affecting Grade Calculation



The tooltip shown indicates that non-default value was assigned to the at_least covergroup option of the descendent of the selected element.

Note: At the time of loading a vPlan, if an analysis context is already launched, then an additional dialog box, as shown in [Figure 5-8](#) on page 256 is displayed.

Figure 5-8 Load vPlan File



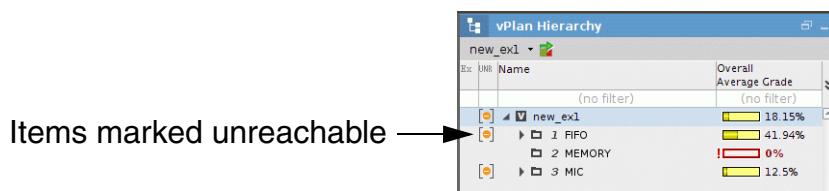
In this dialog box, you can specify if you want to load the vPlan in the current context or you want to create a new context for the vPlan analysis. If you load the vPlan in the current context and run data is loaded in the context, the vPlan that you load shows the coverage numbers. Else, if it is opened in the new context, the coverage numbers show as 0%. After you make your selection, the *Load* dialog box is shown using which you can select the vPlan and load it.

Important

If a vPlan is already loaded in a context and you load another vPlan, then the previously loaded vPlan is automatically unloaded.

Note: Sometimes, you might see an icon appearing in the *UNR* column in the vPlan hierarchy pane, as shown in [Figure 5-9](#) on page 256.

Figure 5-9 Items Marked Unreachable by IEV



An icon appears if the listed section or port includes metric items that were marked unreachable (UNR) by *Incisive Enterprise Verifier* (IEV).

Note: The color of the icon and the tooltip indicates if the item was marked as fully unreachable or partially unreachable.

- A red color icon indicates fully unreachable. This means all the items within that section or port are marked unreachable. A fully unreachable icon shows the tooltip as F_UNR.
- An orange color icon indicates partially unreachable. This means that not all the items within that section or port are marked unreachable. A partially unreachable icon shows the tooltip as P_UNR.

- A totally gray icon indicates unreachable item that is marked excluded. This item was unreachable and later marked excluded in vManager. Such items shows the tooltip as Excluded Unreachable.

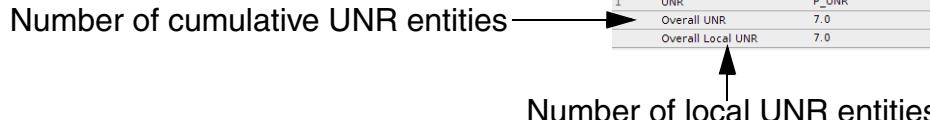
Note: Similar icons are also seen in the *Relative Elements* table of *List tabs* page.

Figure 5-10 on page 257 shows the attributes related to UNR items.

Figure 5-10 Attributes Related to UNR Items

UNR status
 (F_UNR: Fully unreachable)
 (P_UNR: Partially unreachable)
 (E_UNR: Excluded unreachable)

Col #	Name	Value
1	unr	P_UNR
	Overall UNR	7.0
	Overall Local UNR	7.0



The following attributes are related to UNR items:

- UNR — Shows the UNR status, which can be any of the following:
 - P_UNR for partially unreachable items
 - F_UNR for fully unreachable items
 - E_UNR for excluded unreachable items
- Overall Local UNR— Shows the number of local unreachable items.
- Overall UNR— Shows the number of unreachable items within the selected item. This attribute is very useful when analyzing partially unreachable items. The value of this attribute is calculated as:

$$\text{Overall UNR} = \frac{\langle \# \text{ UNR Blocks, expressions, toggles} \rangle}{\langle \# \text{ Blocks, expressions, toggles} \rangle}$$

where

- $\langle \# \text{ UNR Blocks, expressions, toggles} \rangle$ is recursive count of all blocks, expressions, and toggles that are marked as unreachable in the module/instance sub-hierarchy.

- <# Blocks, expressions, toggles> is recursive count of all blocks, expressions, and toggles in the module/instance sub-hierarchy.

Similar to other attributes, you can also add UNR related attributes in the *Verification Hierarchy* pane for analysis.

Note: The UNR markers (F_UNR and P_UNR) are also shown in the HTML-based report, as shown in [Figure 5-11](#) on page 258.

Figure 5-11 UNR Markers in HTML Reports

Current Node: /new_ex1					
Exclusion Rule Type	UNR	Name	Overall Average Grade	Overall Covered	Assertion Status Grade
None	P_UNR	new_ex1	27.22%	7 / 24 (29.17%)	n/a
Sub-Nodes:					
Exclusion Rule Type	UNR	Name	Overall Average Grade	Overall Covered	Assertion Status Grade
None	P_UNR	FIFO	41.94%	6 / 16 (37.5%)	n/a
None	n/a	MEMORY	n/a	0 / 0 (n/a)	n/a
None	P_UNR	MIC	12.5%	1 / 8 (12.5%)	n/a

↑
UNR markers

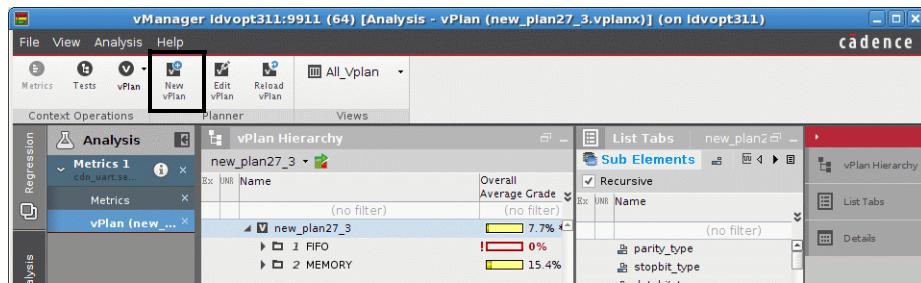
For details on generating HTML reports, see [Generating vPlan Reports](#) on page 309.

For details on the UNR flow and how items are marked UNR by IEV, see the *IEV User Guide*.

5.2 Creating a New vPlan

To create a new vPlan, click the *New vPlan* button on the toolbar, as shown in [Figure 5-12](#) on page 259.

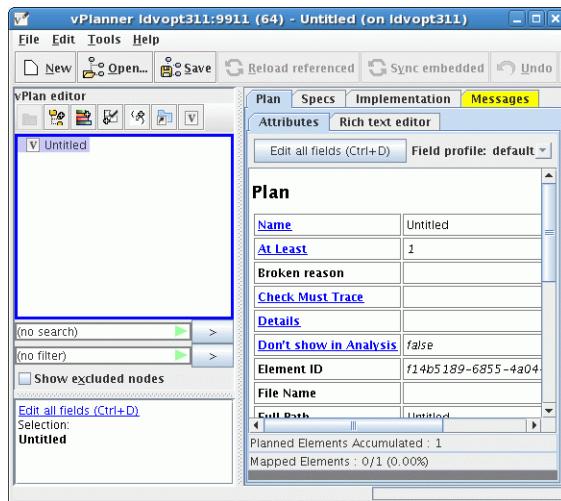
Figure 5-12 New vPlan



Note: Alternatively, you can select the *New vPlan* option from the *Analysis* menu.

This will launch vPlanner, as shown in [Figure 5-13](#) on page 259.

Figure 5-13 vPlanner



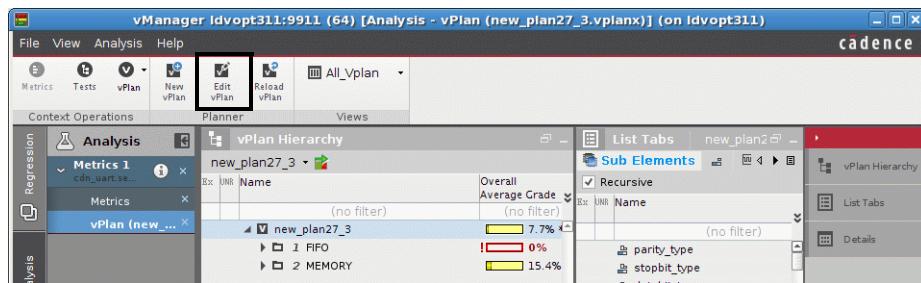
You can now create a new vPlan in vPlanner.

Note: If you launch vPlanner from an existing context where metrics data is already loaded, then the model (metrics) is also loaded to the vPlanner at the time of launching vPlanner.

5.3 Editing vPlans

To edit a vPlan, click the *Edit vPlan* button on the vPlan toolbar, as shown in [Figure 5-14](#) on page 260.

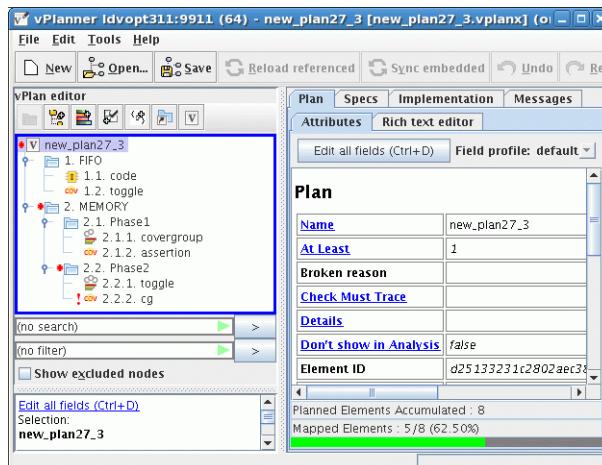
Figure 5-14 Edit vPlan



When you click the *Edit vPlan* button, vPlanner is launched and you can edit vPlans, as required.

If at the time of clicking the *Edit vPlan* button a vPlan is loaded in vManager, then when you click the *Edit vPlan* button, vPlanner is launched and the vPlan along with the coverage model is automatically loaded for editing. For example, in the above figure, vPlan named new_plan27_3.vplanx is already loaded in vManager. When you click the *Edit vPlan* button, vPlanner is launched and new_plan27_3.vplanx is already loaded, as shown in [Figure 5-15 on page 260](#).

Figure 5-15 vPlanner



Notice that the vPlan that was already launched in vManager is automatically loaded in vPlanner for editing.

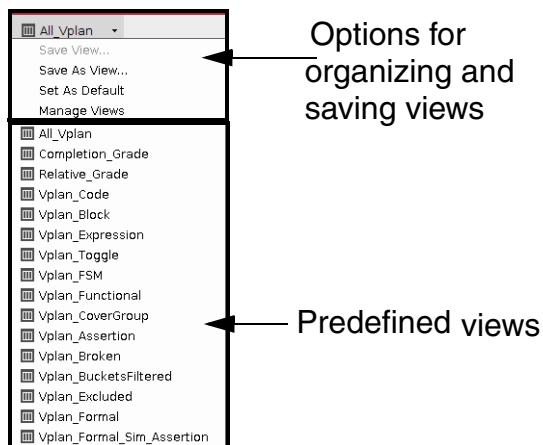
5.4 Organizing and Defining vPlan Views

vManager allows you to define views such that you include only the attributes of your interest. You can also apply sorting, filtering, and grouping as per your requirements and save it as a view.

The *Views* drop-down in the toolbar allows you to define your own views and also set available predefined views.

[Figure 5-16 on page 261](#) shows the options available in the *Views* drop-down list.

Figure 5-16 Views Drop-Down



You can select any of the following predefined views:

- All_vPlan—To show *Overall Average Grade*, *Overall Covered*, and *Assertion Status Grade* attributes in the vPlan Hierarchy pane.
- Completion_Grade—To show only the *Mapped Elements Grade* and *Overall Covered* attributes in the vPlan Hierarchy pane. For more details on Mapped Elements Grade, see [Mapped Elements Grade Calculation](#) on page 267.
- Relative_Grade—To show only *Goal Relative Grade* and *Overall Covered* attributes in the vPlan Hierarchy pane. For more details on Relative Grade, see [Goal-Relative Grade Calculation](#) on page 266.
- Vplan_Code—To show *Code Average Grade* and *Code Covered* attributes in the vPlan Hierarchy pane.
- Vplan_Block—To show *Block Average Grade* and *Block Covered* attributes in the vPlan Hierarchy pane.

- vPlan_Expression—To show *Expression Average Grade* and *Expression Covered* attributes in the vPlan Hierarchy pane.
- vPlan_Toggle—To show *Toggle Average Grade* and *Toggle Covered* attributes in the vPlan Hierarchy pane.
- vPlan_FSM—To show *FSM Average Grade* and *FSM Covered* attributes in the vPlan Hierarchy pane.
- vPlan_Functional—To show *Functional Average Grade* and *Functional Covered* attributes in the vPlan Hierarchy pane.
- vPlan_Covergroup—To show only *Covergroup Average Grade* and *Covergroup Covered* attributes in the vPlan Hierarchy pane.
- vPlan_Assertion—To show *Assertion Average Grade*, *Assertion Covered*, and *Assertion Status Grade* attributes in the vPlan Hierarchy pane.
- vPlan_Broken—To show *Overall Average Grade*, *Overall Covered*, *Assertion Status Grade*, and *Broken reason* attributes in the vPlan Hierarchy pane. This view shows all entities with broken mappings.
- vPlan_BucketsFiltered—To show *Overall Average Grade*, *Overall Covered*, *Assertion Status Grade*, and *Bin Filter* attributes in the vPlan Hierarchy pane. This view shows all entities with bin filter.
- vPlan_Excluded—To show *Overall Average Grade*, *Overall Covered*, *Assertion Status Grade*, and *Is Excluded* attributes in the vPlan Hierarchy pane. This view shows all excluded entities.
- vPlan_Formal—To show *Formal Average Grade*, *Formal Covered*, *Formal Status Grade*, and *Valid Metrics* attributes in the vPlan Hierarchy pane.
- vPlan_Formal_Sim_Assertion—To show *Assertion Average Grade*, *Assertion Covered*, *Assertion Status Grade*, *Formal Average Grade*, *Formal Covered*, *Formal Status Grade*, and *Valid Metrics* attributes in the vPlan Hierarchy pane.

Note: The views are filtered to show only relevant entities. For example, the metrics related views mentioned above are filtered based on the attribute *Valid Metrics*. The *Valid Metrics* attribute has a pre-defined filter already set based on the view you select. For example, if you select *vPlan_Block*, then the vPlan hierarchy will show only items where block coverage is scored.

Apart from the above predefined views, you can set and organize views. For more details, see [Defining and Organizing Views](#) on page 43.

By default, when you load a vPlan, *All vPlan* view is already set. However, you can change the default view in the *Configuration* dialog box by:

1. Select *View* —> *Configuration*.
2. Select *Views* under the *Flows* folder in the left pane.
3. Select the *Override default view for vPlan analysis* check box.
4. Select the desired view from the *vPlan Analysis Default View* drop-down.
5. Click *OK*.

Viewing Broken, Bin Filtered, and Excluded Items

In the vPlan analysis pages, you can set a required view, and view only the excluded items, items with broken mappings, or bin filtered items.

You can select any of the following views:

- vPlan_Broken—To show all entities with broken mappings. This view adds an additional column, *Broken reason* to the table; and filters the table data to show items for which the *Broken reason* is not blank, as shown in [Figure 5-17](#) on page 263.

Figure 5-17 vPlan_Broken View

Ex	UNR	Name	Overall Average Grade	Overall Covered	Assertion Status	Broken reason
		(no filter)	(no filter)	(no filter)	(no filter)	(no filter) != ""
		new_plan27_3	7.7% *	32 / 84...	n/a	n/a
		I FIFO	0%	0 / 0 (n...)	n/a	n/a
		I.1 code	0%	0 / 0 (n...)	n/a	No metric items
		2 MEMORY	15.4% *	32 / 84...	n/a	n/a

Shows items with broken mappings

Broken reason

- vPlan_BucketsFiltered—To show entities for which bin filter is applied. This view adds an additional column, *Bin Filter*, to the table; and filters the table data to show items for which the *Bin Filter* is not blank, as shown in [Figure 5-18](#) on page 263.

Figure 5-18 vPlan_BucketsFiltered View

Ex	UNR	Name	Overall Average Grade	Overall Covered	Assertion Status	Bin Filter
		(no filter)	(no filter)	(no filter)	(no filter)	(no filter) != ""
		new_plan27_3	7.7% *	32 / 84...	n/a	n/a
		I FIFO	0%	0 / 0 (n...)	n/a	n/a
		I.1 code	0%	0 / 0 (n...)	n/a	(tfifo == 45) or
		2 MEMORY	15.4% *	32 / 84...	n/a	n/a

Shows items on which bin filter is applied

Bin filter

Note: The term bucket has been replaced with bin, and particularly bucket filter has been

renamed to bin filter.

- vPlan_Excluded—To show only the excluded items, as shown in [Figure 5-19](#) on page 264.

Figure 5-19 vPlan_Excluded View

Ex	UIN	Name	Overall Average Grade	Overall Covered	Assertion Status Grd	Is Active	Is Excluded
		(no filter)	(no filter)	(no filter)	(no filter)	false	x (no filter)
		new_plan27_3	7.7% * 32 / 84 (...)	n/a	true	false	
		I FIFO	0%	0 / 0 (n/a)	n/a	true	false
		1.2 toggle	n/a	0 / 0 (n/a)	n/a	false	true
		MEMORY	15.4% * 32 / 84 (...)	n/a	true	false	

These views are available in all vPlan analysis pages.

5.5 Computing Grade of a Section

The following sections describe how the grade computations are performed by vManager for a vPlan and the parameters that affect the computation.

- [Section Grade Parameters](#)
- [Cumulative Grade Calculation](#)
- [Goal-Relative Grade Calculation](#)
- [Mapped Elements Grade Calculation](#)
- [Assertion Status Grade Calculation](#)

5.5.1 Section Grade Parameters

The section grade is based on the following set of parameters:

- Grades for items and subsections contained in the section
- Weights for items and subsections
- Section goals

5.5.1.1 Weight

Each variable in the formula, whether it is an item or a section in the verification plan, has a non-negative weight that represents its importance in the overall coverage model.

The default weight of an item is 1. You can set the weight for an **e** cover item or cover group with the using weight option.

The default weight of a section is 1. You can set the weight for a section using the weight attribute, either in the section itself, or in a perspective that instantiates the section.

Note:

- A weight of zero means that node or item is ignored.
- When calculating the cumulative average grade or cumulative covered grade, the weight of the sections along the path to an item functions as a multiplier in determining the item's weight.

5.5.1.2 Section Goal

A view can define a goal between 1 and 100 for each section and subsection in the verification plan. This goal serves as the maximum grade required for this section in this view.

The default value for goal for a section is 100. You can set the goal for a section using the goal attribute, either in the section itself, or in a perspective that instantiates the section.

Note:

- The goal of a sub-section does not affect the grade of its parent section.
- The goal of a section is taken into account when calculating the goal-relative grade for the section.

5.5.2 Cumulative Grade Calculation

The cumulative grade is of following types:

- Cumulative Average Grade
- Cumulative Covered Grade

5.5.2.1 Cumulative Average Grade

vManager uses the following formula to calculate cumulative average grade:

$$\text{Cumulative Average Grade} = \frac{M_{\text{self}} + \sum M_{\text{cum}}}{1 + \text{Number of immediate child instances}}$$

where M_{self} is the self metric (taking into consideration goal value, if defined) and M_{cum} is the cumulative metric of each of the immediate children while taking into consideration each weight and goal values.

Note: Average rollup method is used to calculate the Cumulative Average Grade of all the items and subsections contained in the section. Average rollup method was earlier known as the *Top Elements* scheme.

5.5.2.2 Cumulative Covered Grade

vManager uses the following formula to calculate cumulative covered grade:

$$\text{Cumulative Covered Grade} = \frac{\sum_{Me_covered}}{\sum_{Me_total}} + \frac{\text{Descendents covered (i)}}{\text{Descendents total (i)}}$$

where:

- Covered coverage of item i ($\text{covered}(i)$) is calculated as:

$$\text{covered}(i) = \sum_{\text{Children}} \text{Covered_bin}(i)$$

Here, $\text{covered_bin}(i)$ is the number of covered bins of item i .

- Total coverage of item i ($\text{total}(i)$) is calculated as:

$$\text{total}(i) = \sum_{\text{Children}} \text{bin}(i)$$

Here, $\text{bin}(i)$ is the number of bins of element i .

vManager does not consider weight while calculating covered grades.

Note: The *Covered* rollup method is used to calculate the Cumulative Covered Grade of all the items and subsections contained in the section. *Covered* rollup method was earlier known as the *All Buckets* scheme.

Note: For Local grade calculations, see [Local Grade Calculation](#) on page 241.

5.5.3 Goal-Relative Grade Calculation

In the verification plan, you can define section goals that are lower than 100 percent. These goals do not affect the absolute coverage grade. Instead, vManager uses the goal to compute the goal-relative grade.

The goal-relative grade is computed per section and is the normalization of the coverage grade of that section with the specified goal, and is calculated as:

$$G_{\text{relative}} = \frac{G_{\text{vplan}}}{Goal}$$

where:

- G_{relative} is the goal-relative grade for the given section.
- G_{vplan} is the coverage grade for the given section.
- Goal is the specified goal for the given section.

Note:

- The goal-relative grade might change according to the selected view of the vPlan, as different views might define different goals.
- The default goal for each section is 100; therefore, by default, the goal-relative grade for a given section is equal to the coverage grade for the given section.
- The goal-relative grade for the given section is never greater than 100%.

5.5.4 Mapped Elements Grade Calculation

Mapped Elements Grade is the percentage of mapped elements out of planned elements accumulated. It is calculated as:

`<Mapped elements>/<Planned elements accumulated> * 100`

where:

- Mapped elements indicates the number of elements mapped to the selected item.
- Planned elements accumulated indicates the following:
 - For a metrics port:
 - 1 if there are no mapping on the metric port and the *Planned Elements* attribute is not set.
 - Value that is set for the *Planned Elements* attribute if there are no mapping on the metric port and the *Planned Elements* attribute is set.
 - Maximum between the actual number of mapping and *Planned Elements* attribute value if mapping exists for the metric port.

- ❑ For a section or subsection: It shows the rolled up result, which is the sum of planned elements of all the children within the section.

[Figure 5-20 on page 268](#) shows the *Mapped Elements Grade* attribute and few related attributes in the vPlan Hierarchy pane.

Figure 5-20 Mapped Elements Grade

Name	Mapped Elements	Planned Elements	Mapped Elements Grade	Planned Elements Accumulated
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
new_plan27_3	3 / 7 (4 n/a)	42.86%	7	
1 FIFO	0 / 2 (0 n/a)	0%	2	
1.1 code	0 / 1 (0)	0%	1	
1.2 toggle	0 / 1 (0 n/a)	0%	1	
2 MEMORY	3 / 5 (6 n/a)	60%	5	
2.1 Phase1	2 / 3 (6 n/a)	66.67%	3	
2.1.1 covergroup	2 / 2 (1 n/a)	100%	2	
2.1.2 assertion	0 / 1 (0 n/a)	0%	1	
2.2 Phase2	1 / 2 (5 n/a)	50%	2	
2.2.1 toggle	1 / 1 (1 n/a)	100%	1	
2.2.2 cg	0 / 1 (0 n/a)	0%	1	

In the above vPlan Hierarchy pane:

- *Mapped Elements* column shows the following:

Mapped elements/Planned elements (Mapped elements grade)

where Mapped elements grade is the:

Percentage(%) of Mapped elements/Planned elements accumulated

- *Planned Elements* column shows the following:

- ❑ n/a if the attribute *Planned Elements* is not set in vPlanner.

- ❑ Value of the attribute *Planned Elements* if it set in vPlanner.

- *Mapped Elements Grade* column shows the percentage of mapped elements out of planned elements accumulated.

- *Planned Elements Accumulated* column shows the following:

- ❑ For a metrics port:

- 1 if there are no mapping on the metric port and the *Planned Elements* attribute is not set.
- Value that is set for the *Planned Elements* attribute if there are no mapping on the metric port and the *Planned Elements* attribute is set.
- Maximum between the actual number of mapping and *Planned Elements* attribute value if mapping exists for the metric port.

- For a section or subsection: It shows the rolled up result, which is the sum of planned elements of all the children within the section.

For more details, see *vPlanner User Guide*.

5.5.5 Assertion Status Grade Calculation

The *Assertion Status Grade* is the total percentage of passed assertions out of all assertions. It is calculated as:

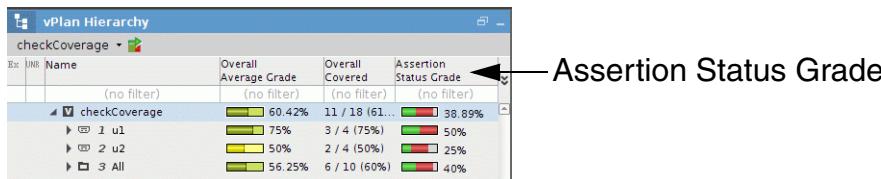
`<TotalPassedAssertions> / <AssertionTotal> * 100`

where

- `TotalPassedAssertions` is the total number of assertions that finished without failure at least once, and never failed.
- `AssertionTotal` is the total number of assertions, which is the sum of `TotalPassedAssertions`, `TotalFailedAssertions`, and `OtherAssertions`.
 - `TotalPassedAssertions` is the total number of assertions that finished without failure at least once, and never failed.
 - `TotalFailedAssertions` is the total number of assertions that finished with failure at least once.
 - `OtherAssertions` is the total number of assertions that never finished. It is calculated as `[AssertionTotal - (TotalPassedAssertions + TotalFailedAssertions)]`

[Figure 5-21](#) on page 269 shows the *Assertion Status Grade* attribute in the vPlan Hierarchy pane.

Figure 5-21 Assertion Status Grade



Note: The *Assertion Status Grade*, by default, is shown in the *All_vPlan* and *vPlan_Assertion* view. If required, you can also add the following attributes to the *vPlan Hierarchy* pane:

- Assertion Passed — It shows the number of non excluded assertions in sub-hierarchy with **Passed** status.
- Assertion Failed — It shows the number of non excluded assertions in sub-hierarchy with **Failed** status.
- Assertion Other — It shows the number of non excluded assertions in sub-hierarchy with **Other** status.
- Status — It shows the status of the assertion, which can be any of the following:
 - **Passed** if the assertion finished without failure at least once, and never failed.
 - **Failed** if the assertion finished with failure at least once.
 - **Other** if the assertion never finished.

[Figure 5-22 on page 270](#) shows the above listed attributes in the vPlan Hierarchy pane.

Figure 5-22 Assertion Status Grade

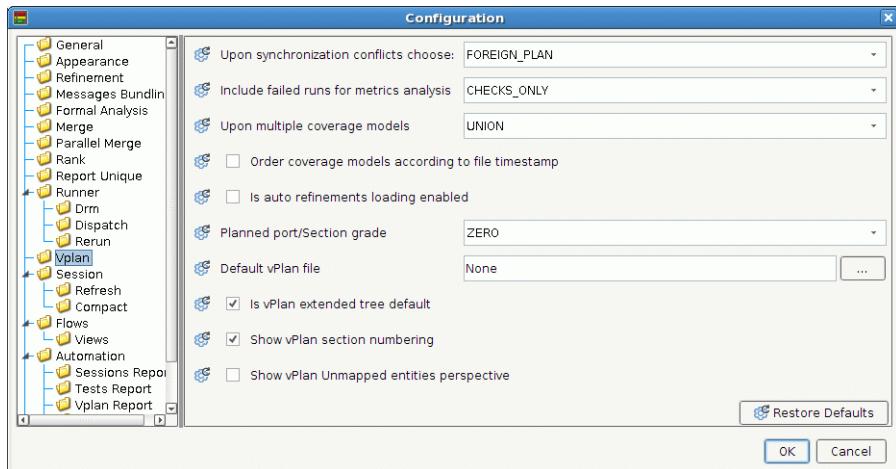
The screenshot shows a tree view of a vPlan hierarchy. The root node is 'checkCoverage'. Under 'checkCoverage', there are two nodes: 'u1' and 'u2'. 'u1' has four child nodes: 'always_pass', 'always_fail', 'sometimes_fail', and 'called_in_u1'. 'u2' has one child node: 'All'. To the right of the tree view is a table with six columns: 'Assertion Status Grade', 'Status', 'Assertion Passed', 'Assertion Failed', 'Assertion Other', and 'Count'. The table data is as follows:

Assertion Status Grade	Status	Assertion Passed	Assertion Failed	Assertion Other	Count
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	0
checkCoverage	38.89% n/a	7	9	2	0
u1	50% n/a	2	2	0	0
u1	50% n/a	2	2	0	0
always_pass	100% Passed	1	0	0	0
always_fail	0% Failed	0	1	0	0
sometimes_fail	0% Failed	0	1	0	0
called_in_u1	100% Passed	1	0	0	0
u2	25% n/a	1	2	1	0
All	40% n/a	4	5	1	0

5.6 vPlan Configuration Options

[Figure 5-23 on page 271](#) shows the vPlan options that can be configured using the **Configuration** dialog box.

Figure 5-23 Configuration (vPlan Options)



Using the *Configuration* dialog box, you can:

- Specify the vPlan to be Used in the case of Synchronization Conflicts
- Specify if Coverage from Failed Runs Must be Included While Calculating Grades
- Specify How to Merge Multiple Coverage Models
- Enable Ordering of Coverage Models According to Timestamp
- Enable or Disable Loading of vPlan Refinements File Automatically
- Metrics Port of a vPlan to Show Zero or NA for No Mapped Elements
- Specify Default vPlan File
- Enable or Disable Showing the vPlan Extended Tree
- Enable or Disable Showing Section Numbers
- Enable Showing of Unmapped Perspective in vPlan

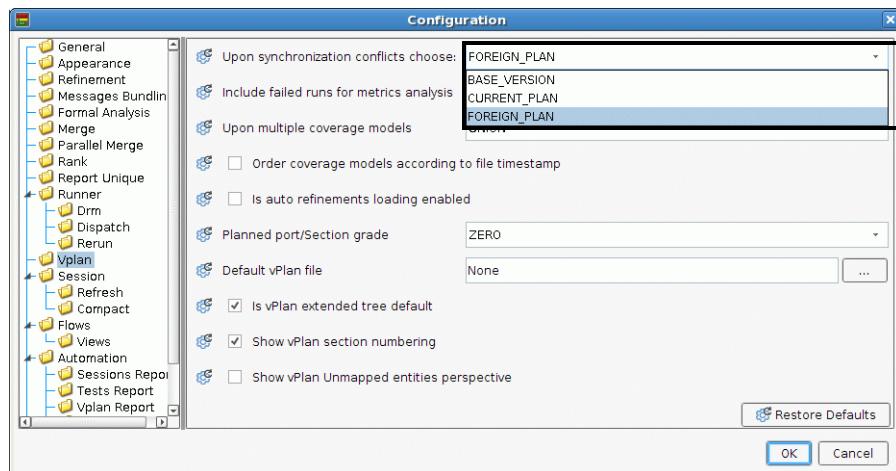
Note: If the changes made in the *Configuration* dialog box affects the coverage results, then open a new context in the *Analysis* window to view the updated results.

5.6.1 Specify the vPlan to be Used in the case of Synchronization Conflicts

The *Upon synchronization conflict choose* option allows you to specify the plan that must be used in the case of synchronization conflicts in Enterprise Planner. The value that you specify here defines the Enterprise planner behavior in case of synchronization conflicts.

[Figure 5-24 on page 272](#) shows the values that you can set for this option.

Figure 5-24 Configuration Options for Synchronization Conflicts



The following values can be set for defining the vPlanner behavior in case of synchronization conflicts:

■ **BASE_VERSION**

Select this option when you want vPlanner to use the values from the base version that is before any changes were made to the top-level plan or the embedded plan in the event of synchronization conflict.

■ **CURRENT_PLAN**

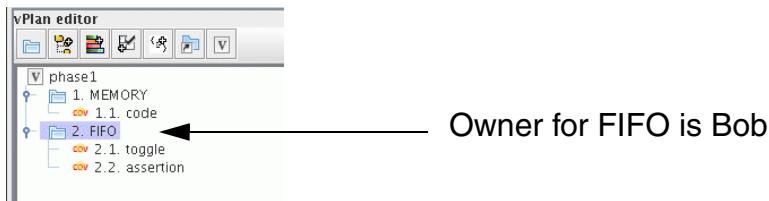
Select this option when you want vPlanner to use the values from the top-level plan in the event of synchronization conflict.

■ **FOREIGN_PLAN**

Select this option when you want vPlanner to use the values from the embedded plan in the event of synchronization conflict. This is the default behavior.

Consider an example. A plan named `phase1.vplanx` is created, as shown in [Figure 5-25 on page 273](#).

Figure 5-25 vPlan



Now, this plan is embedded in a top-level plan `sys.vplanx`.

After embedding the plan, the owner for FIFO is changed to Mike in the top-level plan.

Later, the owner in the embedded plan is changed to David.

Now, depending on the selection in the *Configuration* dialog box in vManager, the behavior of the *Sync Embedded* button of vPlanner is controlled.

If Configuration is specified as:	Result after Synchronization
BASE_VERSION	Bob
CURRENT_PLAN	Mike
FOREIGN_PLAN	David

For more details on synchronization, see the *vPlanner User Guide*.

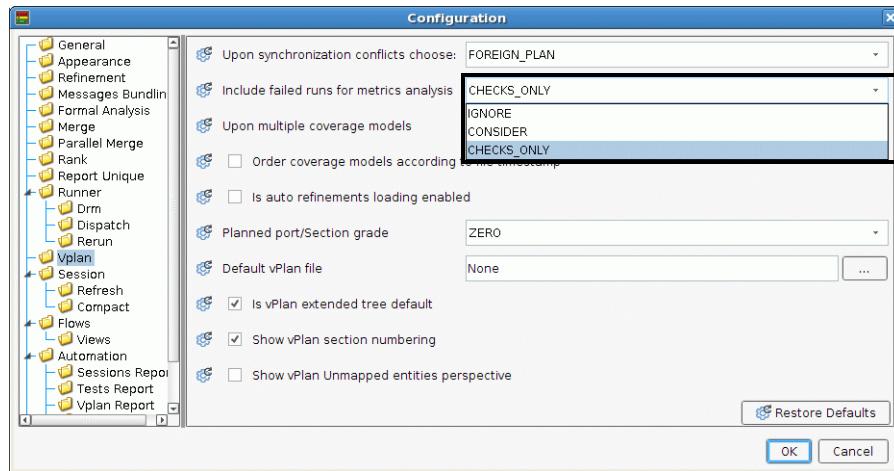
5.6.2 Specify if Coverage from Failed Runs Must be Included While Calculating Grades

The *Include failed runs for metrics analysis* drop-down lets you specify whether coverage data from failed runs must be included when calculating coverage grades. The valid options are:

- IGNORE —This causes vManager to not include failed runs when calculating grades. This is the default selection.
- CONSIDER —This causes vManager to include failed runs when calculating grades for checks and coverage.
- CHECKS_ONLY —This causes vManager to include only failed runs when calculating grades for checks.

The default value of *Include failed runs for metrics analysis* is CHECKS_ONLY. To change the default value, select the required option from the *Include failed runs for metrics analysis* drop-down, as shown in [Figure 5-26](#) on page 274.

Figure 5-26 **Include Failed Runs for Metrics Analysis**



5.6.3 Specify How to Merge Multiple Coverage Models

The *Upon multiple coverage models* drop-down specifies how to deal with multiple coverage models. The valid options are:

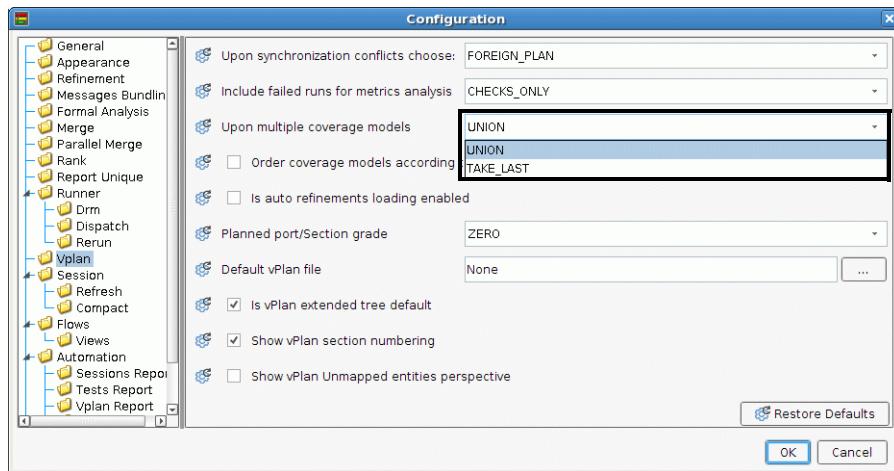
- UNION —This causes vManager to create a model that is a union of all the loaded models. vManager sorts the coverage models of the sessions according to session start time and performs a union of the models starting with the newest model. As a result, the order in which you load the sessions does not affect the union.

Note: vManager performs a union of coverage models from the same verification scope.

- TAKE_LAST —This causes vManager to take only one coverage model that is the newest according to the session start time.

The default value of *Upon multiple coverage models* is UNION. To change the behavior to use the latest model, select TAKE_LAST from the *Upon multiple coverage models* drop-down, as shown in [Figure 5-27](#) on page 275.

Figure 5-27 Upon Multiple Coverage Models



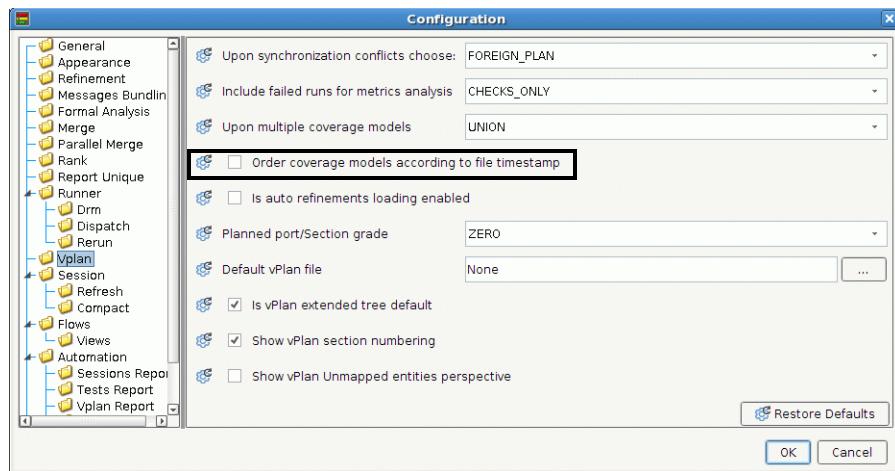
Note: This option controls how to merge coverage models after the verification plan is already loaded.

5.6.4 Enable Ordering of Coverage Models According to Timestamp

By default, vManager reads coverage models for merging according to the order in which they appear in the database.

However, if you want vManager to read coverage models for merging on the basis of the timestamp (creation/modification time on the disk), select the *Order coverage models according to file timestamp* option in the *Configuration* dialog box, as shown in [Figure 5-28 on page 276](#).

Figure 5-28 Ordering Coverage Models



Selecting this option will enforce vManager to merge coverage models on the basis of timestamp instead of the order in which they were inserted to the database.

Note: This option is relevant if the `TAKE_LAST` option was selected from the *Upon multiple coverage models* drop-down in the *Configuration* dialog box.

5.6.5 Enable or Disable Loading of vPlan Refinements File Automatically

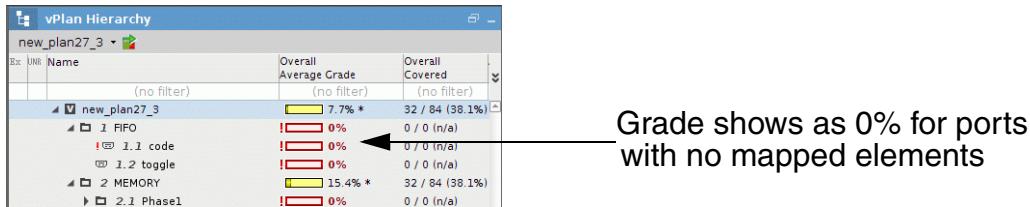
If you apply refinements to a vPlan and save the file, then by default, when you reload the vPlan again the refinements are not automatically loaded. To ensure that the saved refinements are loaded automatically when loading the vPlan, select the *Is auto refinements loading enabled* check box.

For more details, see [Automatically Loading Saved Refinements at the Time of Loading the vPlan](#) on page 303.

5.6.6 Metrics Port of a vPlan to Show Zero or NA for No Mapped Elements

By default, if a metrics port has no mapped elements, grade is taken as 0%, as shown in [Figure 5-29](#) on page 277.

Figure 5-29 Grade 0% for Unmapped Ports

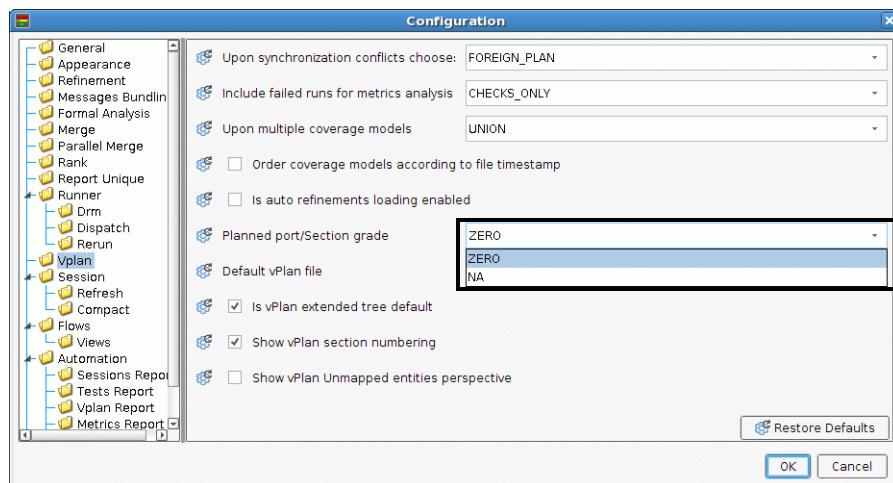


To assign a grade as n/a for items with no mapped elements:

1. Select *Configuration* from the *View* menu.
2. Click the *Vplan* folder.

The vPlan options are displayed, as shown in [Figure 5-30](#) on page 277.

Figure 5-30 Configuration—vPlan Options



3. To assign a grade as n/a for unmapped elements, select *NA* from the *Metric port grade when it has no mapped elements* drop-down.
4. Click *OK*.

This will ensure that next time when the vPlan is loaded in the same vManager session or the next vManager session the grade for vPlan metrics ports with no mapped elements show as n/a.

5.6.7 Specify Default vPlan File

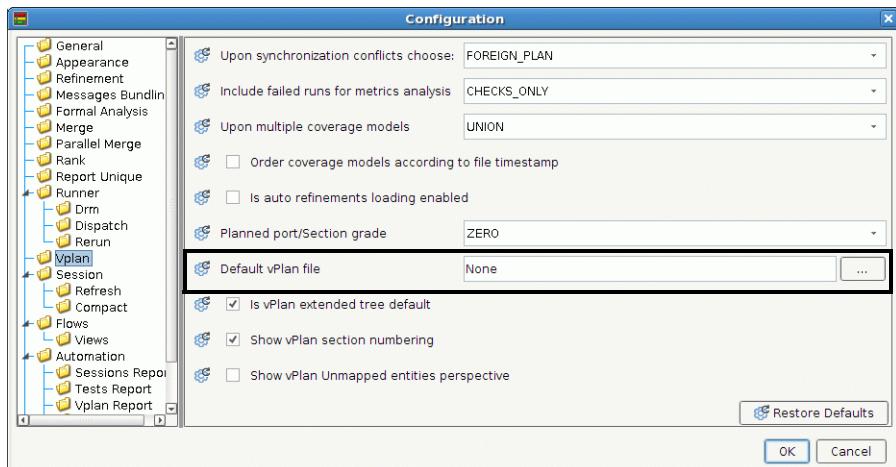
Using vManager, you can specify a default plan that should be loaded automatically when you select the Analyze vPlan option.

To specify a default vPlan:

1. Select *Configuration* from the *View* menu.
2. Click the *Vplan* folder.

The vPlan options are displayed, as shown in [Figure 5-31](#) on page 278.

Figure 5-31 Configuration—vPlan Options



3. To specify a default vplan file specify the full hierarchical path of the vPlan file. You can also select the eclipse (. . .) button and navigate through the hierarchy and select the file.
4. Click *OK*.

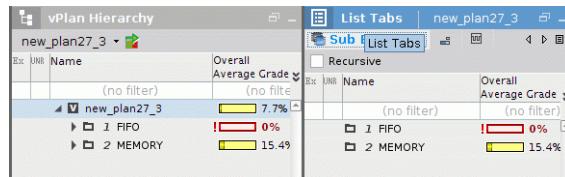
This will ensure that by default, the specified file will be loaded automatically when you select the *Analyze vPlan* option.

5.6.8 Enable or Disable Showing the vPlan Extended Tree

Using vManager, you can enable or disable showing of mapped elements in the vPlan hierarchy tree.

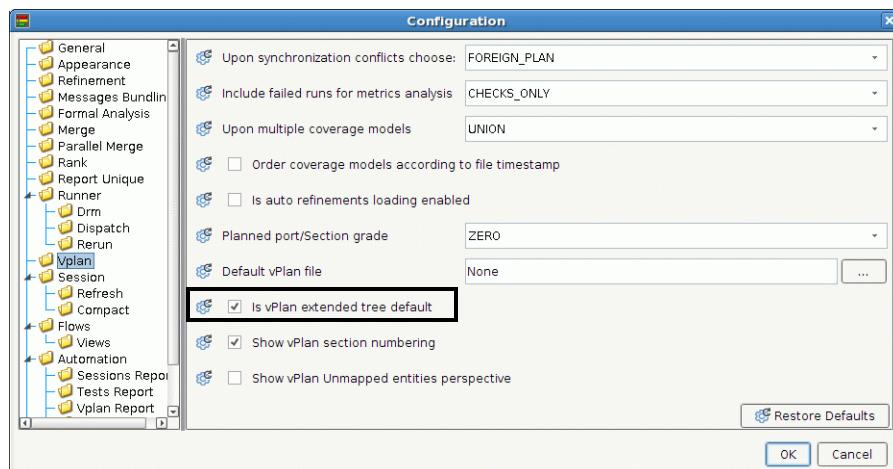
Consider [Figure 5-32](#) on page 279 which shows a vPlan loaded in vManager.

Figure 5-32 vPlan Loaded in vManager



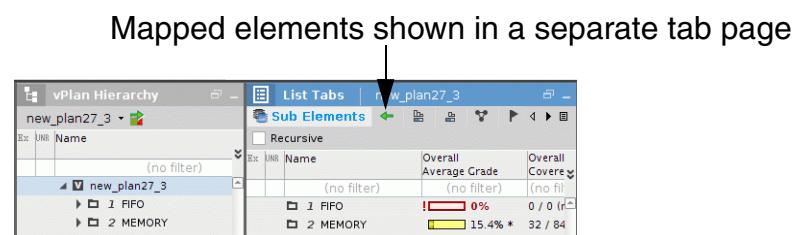
This vPlan has few mapped elements. The mapped elements are shown in the vPlan hierarchy tree and also in the *Sub Elements* table because the *Is vPlan extended tree default* option is enabled, as shown in [Figure 5-33](#) on page 279.

Figure 5-33 Configuration—vPlan Options



To stop showing the mapped elements in the vPlan hierarchy tree, clear the check box and click *OK*. [Figure 5-34](#) on page 279 shows the vPlan after the *Is vPlan extended tree default* option check box is cleared.

Figure 5-34 vPlan Loaded in vManager

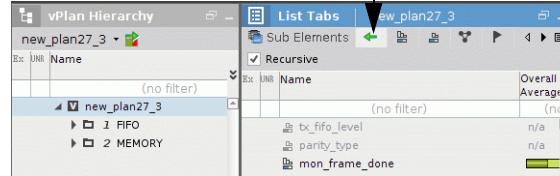


The mapped elements are no longer shown in the vPlan hierarchy tree and the *Sub Elements* table. Rather an additional tab page is added which shows the mapped elements.

When you click the *Mapped Elements* tab page, the mapped elements are displayed, as shown in [Figure 5-35](#) on page 280.

Figure 5-35 Mapped Elements Shown in Separate Tab Page

Mapped elements shown in a separate tab page

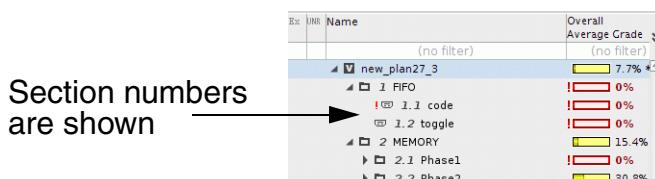


Mapped elements now show in a separate tab page.

5.6.9 Enable or Disable Showing Section Numbers

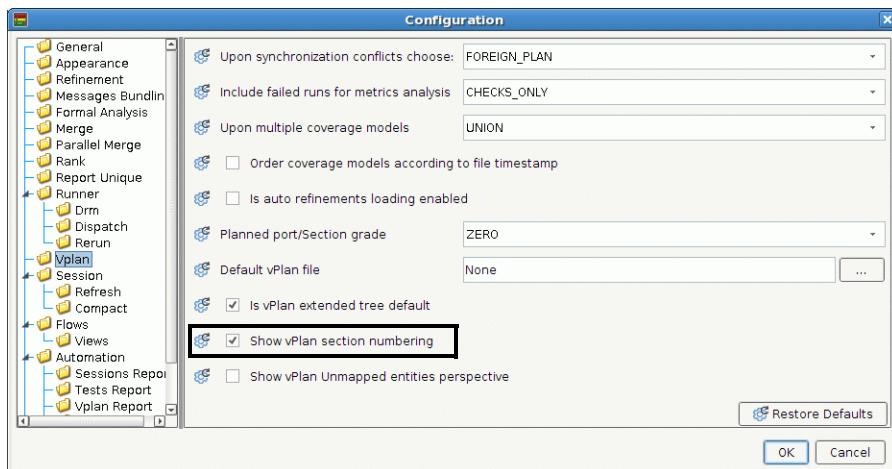
Using vManager, you can enable or disable showing of section numbers in the vPlan hierarchy tree. By default, the section numbers are shown in the vPlan hierarchy tree, as shown in [Figure 5-36](#) on page 280.

Figure 5-36 vPlan Showing Section Numbers



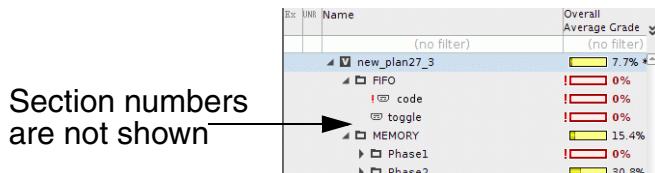
To stop showing the section numbers in the vPlan hierarchy tree, clear the *Show vPlan section numbering* check box and click *OK*, as shown in [Figure 5-37](#) on page 281.

Figure 5-37 Configuration—vPlan Options



After you clear the check box, section numbers are not shown in the vPlan hierarchy tree, as shown in [Figure 5-38](#) on page 281.

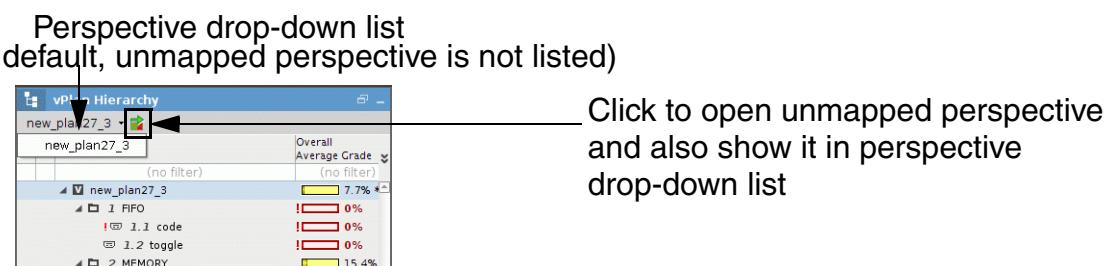
Figure 5-38 vPlan Not Showing Section Numbers



5.6.10 Enable Showing of Unmapped Perspective in vPlan

By default, Unmapped Perspective is not shown in the perspective drop-down list, as shown in [Figure 5-39](#) on page 281.

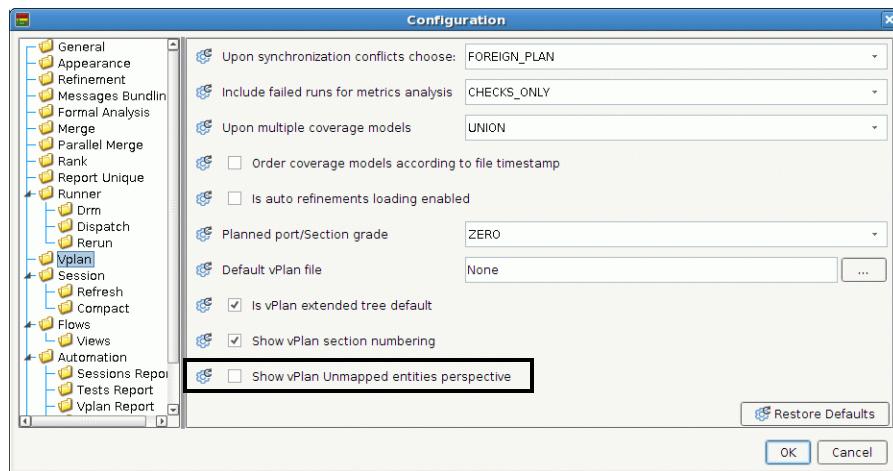
Figure 5-39 vPlan Perspective Drop-Down List



You can enable showing of Unmapped Perspective in the perspective drop-down list by doing any of the following:

- Method 1: Click the button next to the perspective drop-down list
- Method 2: Select the *Show vPlan Unmapped entities perspective* check box in the *Configuration* dialog box and click *OK*, as shown in [Figure 5-40](#) on page 282.

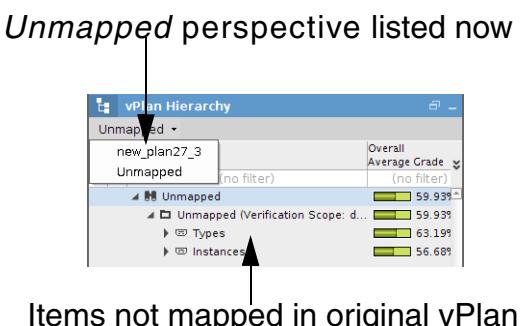
Figure 5-40 Configuration—vPlan Options



Note: If you use Method 2, then the *Unmapped Perspective* option is listed by default, only for the vPlans loaded after selecting this option.

However, if you use Method 1, (that is -- clicking the button next to the perspective drop-down list), then the *Unmapped* perspective is launched and the *Unmapped* option is also listed immediately in the perspective drop-down list, as shown in [Figure 5-41](#) on page 282.

Figure 5-41 vPlan Perspective Drop-Down List



The *Unmapped* perspective lists the entities that are defined in the verification environment but not included or mapped in the verification plan. This view is useful for identifying the items that can be mapped to the verification plan.

5.7 Looking for a Section or Port

The *Lookup* toolbar, shown in [Figure 5-42](#) on page 283 allows you to search for and select the section of a selected port or section, look for the section or port in the *vPlan* hierarchy tree, or look for the parent section of the selected section.

Figure 5-42 *Lookup*



The *Lookup* toolbar has the following options:

- [Lookup Section in Hierarchy](#)
- [Lookup Parent Section in Hierarchy](#)
- [Lookup Port in Hierarchy](#)
- [Lookup Entity in Metric Hierarchy](#)

Note: These options are also available in the *Analysis* menu.

5.7.1 *Lookup Section in Hierarchy*

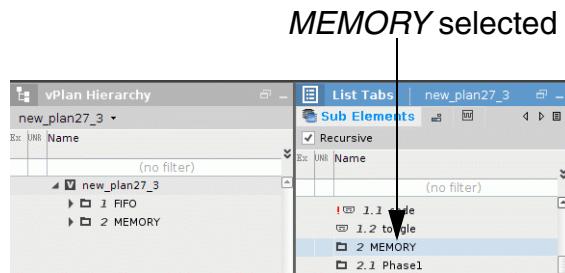
The *Lookup Section in Hierarchy* option is used to look up the selected section in the *vPlan Hierarchy* pane.

This option is available only for the selections made in the *List tabs* pane. You can select the section in the *Sub Elements* tab page of the *List tabs* pane and then select the *Lookup Section in Hierarchy* option. This will immediately take you to the selected section in the *vPlan Hierarchy* pane.

For example, to look up for section *MEMORY* in the *vPlan Hierarchy* pane:

1. Select *MEMORY* in the *Sub Elements* tab page of the *List tabs* pane, as shown in [Figure 5-43](#) on page 284.

Figure 5-43 Lookup Section in Hierarchy

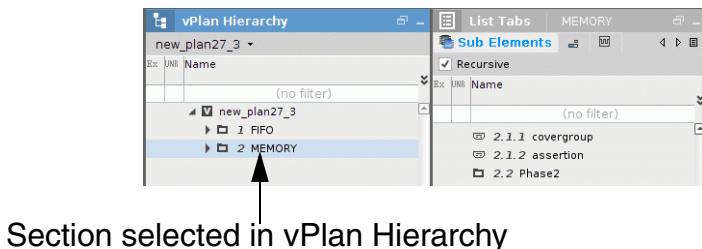


2. Select *Lookup Section in Hierarchy* from the *Analysis* menu.

Note: Alternatively, you can right-click the section in the *Sub Elements* tab page of the *List tabs* pane and select *Lookup Section in Hierarchy* from the pop-up menu or you can select *Lookup Section in Hierarchy* from the *Lookup* toolbar.

The selected instance gets highlighted in the *vPlan Hierarchy* pane, as shown in [Figure 5-44](#) on page 284.

Figure 5-44 Section Highlighted in vPlan Hierarchy Pane



5.7.2 Lookup Parent Section in Hierarchy

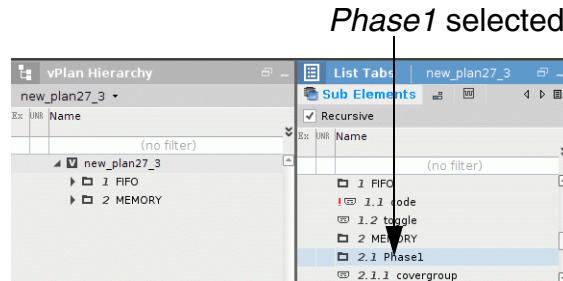
The *Lookup Parent Section in Hierarchy* option is used to look up the parent of the selected section in the *vPlan Hierarchy* pane.

This option is available only for the selections made in the *List tabs* pane. You can select a section in the *Sub Elements* tab page of the *List tabs* pane and then select the *Lookup Parent Section in Hierarchy* option. This will immediately take you to the parent of the selected section in the *vPlan Hierarchy* pane.

For example, to look up for parent of section *Phase1*:

1. Select *Phase1* in the *Sub Elements* tab page of the *List tabs* pane, as shown in [Figure 5-45](#) on page 285.

Figure 5-45 Lookup Parent Section in Hierarchy

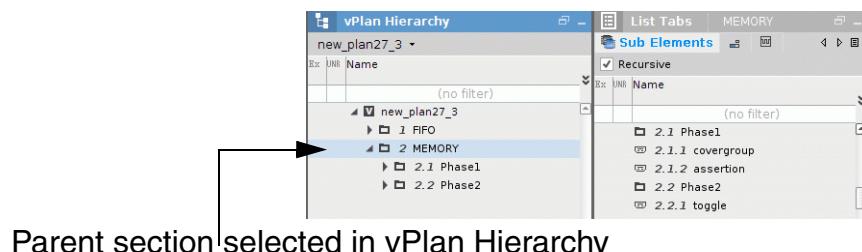


2. Select *Lookup Parent Section in Hierarchy* from the *Lookup* toolbar.

Note: Alternatively, you can right-click the instance in the *Sub Elements* tab page of the *List tabs* pane and select *Lookup Parent Section in Hierarchy* from the pop-up menu.

The selected instance gets highlighted in the *vPlan Hierarchy* pane, as shown in Figure 5-46 on page 285.

Figure 5-46 Parent Instance Highlighted in vPlan Hierarchy Pane



Parent section selected in vPlan Hierarchy

The section *MEMORY* as highlighted as it is the parent instance of *Phase 1*.

5.7.3 Lookup Port in Hierarchy

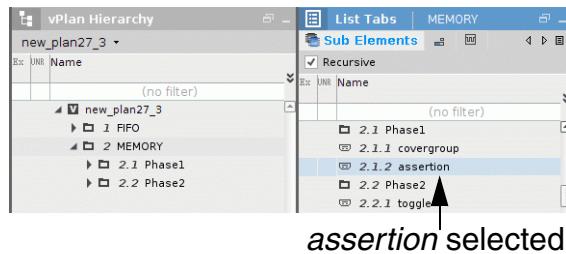
The *Lookup Port in Hierarchy* option is used to look up the selected port in the *vPlan Hierarchy* pane.

This option is available only for the selections made in the *List tabs* pane. You can select the section in the *Sub Elements* tab page of the *List tabs* pane and then select the *Lookup Port in Hierarchy* option. This will immediately take you to the selected port in the *vPlan Hierarchy* pane.

For example, to look up for port *assertion* in the *vPlan Hierarchy* pane:

1. Select *assertion* in the *Sub Elements* tab page of the *List tabs* pane, as shown in [Figure 5-47](#) on page 286.

Figure 5-47 Lookup Port in Hierarchy

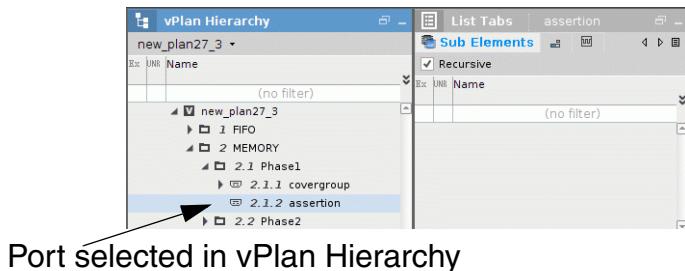


2. Select *Lookup Port in Hierarchy* from the *Lookup* toolbar.

Note: Alternatively, you can right-click the section in the *Sub Elements* tab page of the *List tabs* pane and select *Lookup Port in Hierarchy* from the pop-up menu.

The selected port gets highlighted in the *vPlan Hierarchy* pane, as shown in [Figure 5-48](#) on page 286.

Figure 5-48 Port Highlighted in vPlan Hierarchy Pane



5.7.4 Lookup Entity in Metric Hierarchy

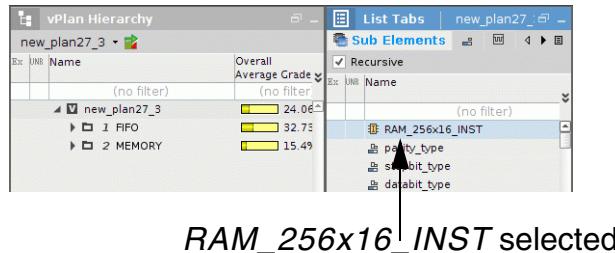
The *Lookup Entity in Metric* option is used to look up the selected entity in the metrics hierarchy (*Verification Hierarchy* pane of the *Metrics* page).

This option is available only for mapped elements. You can select a mapped element in the *Mapped Elements* tab page of the *List tabs* pane or in the *Sub Elements* tab page of the *Extended vPlan tree*. You can then select the *Lookup Entity in Metric* option in the *Lookup* toolbar. This will immediately take you to the *Metrics* page and show the selected element in the *Verification Hierarchy* pane.

For example, to look up for entity *RAM_256x16_INST* in the *Metrics* hierarchy:

1. Select *RAM_256x16_INST* in the *Mapped Elements* tab page of the *List tabs* pane, as shown in [Figure 5-49](#) on page 287.

Figure 5-49 Lookup Entity in Metric Hierarchy

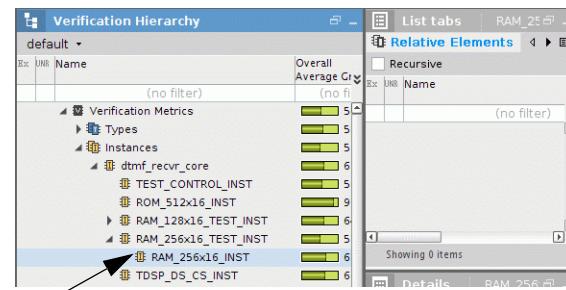


2. Select *Lookup Entity in Metric* from the *Lookup* toolbar.

Note: Alternatively, you can right-click the entity in the *Mapped Elements* tab page of the *List tabs* pane and select *Lookup Entity in Metric* from the pop-up menu.

The *Metrics* page is shown and the selected entity gets highlighted in the *Verification Hierarchy* pane, as shown in [Figure 5-50](#) on page 287.

Figure 5-50 Entity Highlighted in Verification Hierarchy Pane of Metrics page



Entity selected in Verification Hierarchy of Metrics page

5.7.4.1 Lookup in Extended Tree

If the extended tree option is enabled for both vPlan as well as metrics tree, then:

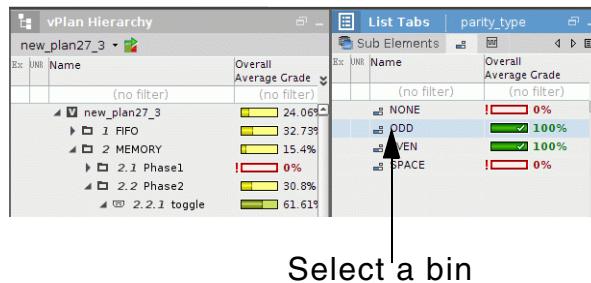
- In the case of toggles and bins, the *Lookup Entity in Metric* option takes the user to the respective item in the Metrics page and to the related metrics tree tab in the Lists tabs pane.
- In the case of FSMs/covergroups/coveritems/assertions, the *Lookup Entity in Metric* option takes the user to the respective item in the Metrics page.

However, if the extended tree configuration option is not enabled for the metrics tree, then the lookup to entity in metrics tree is not possible, and the *Lookup Entity in Metric* option shows disabled.

Consider that the extended tree option is enabled for both vPlan as well as metrics tree.

Select a bin (in this case ODD) in the vPlan page, as shown in [Figure 5-51](#) on page 288.

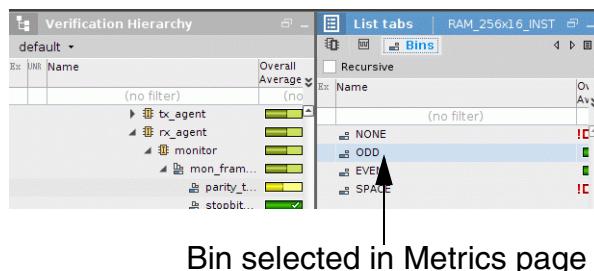
Figure 5-51 Lookup Entity in Metric Hierarchy



Now, right-click and select *Lookup Entity in Metric* option from the popup menu.

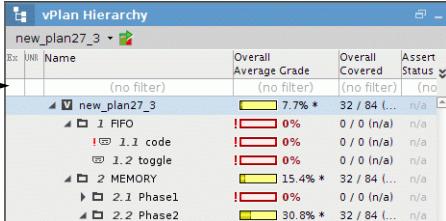
This will open the *Metrics* page and take the user to the related extended metrics tree tab, as shown in [Figure 5-52](#) on page 288.

Figure 5-52 Bin Highlighted in Metrics page



5.8 Filtering Items in vPlan Hierarchy

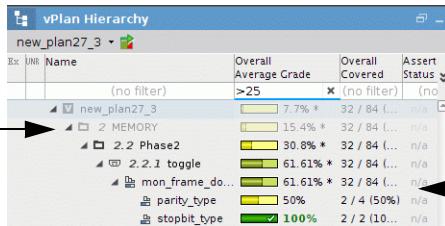
[Figure 5-53](#) on page 289 displays the quick filter bar shown at the top of the vPlan hierarchy pane. The quick filter bar helps you to filter the vPlan hierarchy based on the filter criteria.

Figure 5-53 vPlan Hierarchy Pane—Filter Bar


The screenshot shows the vPlan Hierarchy pane with a header row containing columns for 'Ex UNB Name', 'Overall Average Grade', 'Overall Covered', and 'Assert Status'. Below this is a table with several rows representing items in a hierarchy. A 'Quick filter bar' is overlaid on the top-left of the table, pointing to the first row of the table.

Ex UNB Name	Overall Average Grade	Overall Covered	Assert Status
(no filter)	(no filter)	(no filter)	(no filter)
new_plan27_3	7.7% *	32 / 84 ...	n/a
1 FIFO	0%	0 / 0 (n/a)	n/a
1.1 code	0%	0 / 0 (n/a)	n/a
1.2 toggle	0%	0 / 0 (n/a)	n/a
2 MEMORY	15.4% *	32 / 84 ...	n/a
2.1 Phase1	0%	0 / 0 (n/a)	n/a
2.2 Phase2	30.8% *	32 / 84 ...	n/a

For example, to filter the vPlan hierarchy tree to show items for which *Overall Average Grade* is more than 25, specify *>25* in the filter row and press Enter. After you press Enter, the vPlan hierarchy tree is filtered, as shown in [Figure 5-54](#) on page 289.

Figure 5-54 vPlan Hierarchy Pane—Filtered


The screenshot shows the vPlan Hierarchy pane with a header row containing columns for 'Ex UNB Name', 'Overall Average Grade', 'Overall Covered', and 'Assert Status'. Below this is a table with several rows representing items in a hierarchy. A 'Quick filter bar' is overlaid on the top-left of the table, pointing to the first row of the table. The table has a row for '>25' in the 'Overall Average Grade' column. Arrows point from the text 'Items that partially meet filter criteria' to items like '2.2 Phase2' and '2.2.1 toggle' which are partially transparent, and from the text 'Items that meet filter criteria' to items like 'mon_frame_d...', 'parity_type', and 'stopbit_type' which are solid.

Ex UNB Name	Overall Average Grade	Overall Covered	Assert Status
(no filter)	>25	(no filter)	(no filter)
new_plan27_3	7.7% *	32 / 84 ...	n/a
2 MEMORY	15.4% *	32 / 84 ...	n/a
2.2 Phase2	30.8% *	32 / 84 ...	n/a
2.2.1 toggle	61.61% *	32 / 84 ...	n/a
mon_frame_d...	50%	2 / 4 (50%)	n/a
parity_type	100%	2 / 2 (100%)	n/a
stopbit_type			

The filtered hierarchy tree shows following kinds of items:

- Items that meet the filter criteria. These items are shown as regular items.
- Items that partially meet the filter criteria. These items are shown as partially transparent. These are the items that do not meet the filter criteria completely, but appear because of their children.

The items that do not meet the filter criteria are removed from the hierarchy tree.

For more details on filtering, see [Filtering Table Data](#) on page 33.

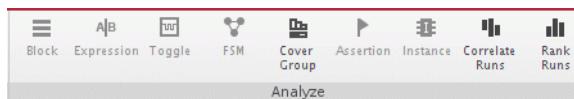
5.9 Launching Detailed Analysis Page for vPlan Entity

After loading the vPlan, you can also launch detailed analysis pages for vPlan entities of your interest. For example, if you want to do FSM analysis of a specific vPlan entity, you can select the entity in the vPlan Hierarchy pane and launch FSM analysis for the selected entity.

The *Analyze* toolbar allows you to launch detailed analysis page for detailed analysis of a selected vPlan entity.

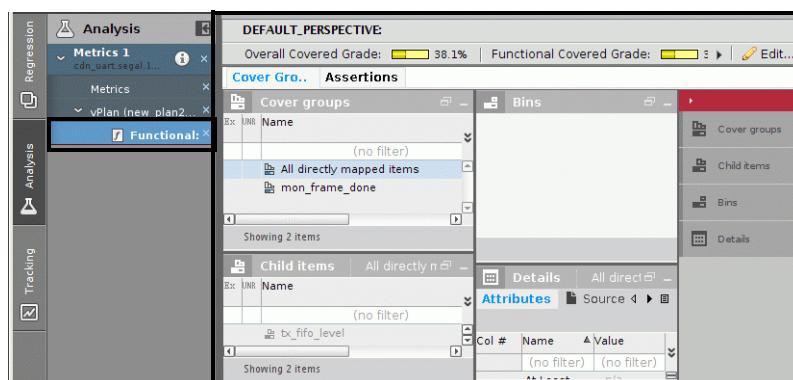
[Figure 5-55 on page 290](#) shows the options available in the *Analyze* toolbar.

Figure 5-55 Analyze Toolbar



For example, to launch Covergroup analysis, select vPlan *Cover Group* from the toolbar. This will launch a detailed analysis page for Covergroup analysis, as shown in [Figure 5-56 on page 290](#).

Figure 5-56 Detailed Analysis Page Launched



Similarly, you can launch more detailed pages, as required.

5.10 Refining vPlan Items

While analyzing a verification plan, you may want to exclude certain undesired ports or sections from analysis. The process of excluding certain items from analysis is called refinement.

The *Refinement* toolbar contains the commands that allow you to exclude sections or ports, un-exclude already excluded items, save refinements, load refinements and show already excluded items.

You can choose the following items from the *Refinement* toolbar or from the *Analysis* menu:

- Exclude—To exclude the selected item and its children
- Edit comment—To edit exclusion comment.

- Un-Exclude—To un-exclude already excluded items
- Clear dirty rules —To clear/delete the dirty rules. This option is available recursively from the type/instance/fsm parent entity. It is also available from the dirty excluded entity itself.
- Approve dirty rules — To approve the dirty rules (make the rule completely valid). This option is available recursively from the type/instance/fsm parent entity. It is also available from the dirty excluded entity itself.
- Clear all orphan rules—To clear/delete all the orphan rules of the selected item. This option is available recursively from the type/instance/fsm parent entity.
- Apply all orphan rules —To apply all the orphan rules of the selected item (by index of originally excluded entity) and clear them. This option is available recursively from the type/instance/fsm parent entity.
- All at once—To refine attributes of the selected sections or metric ports together
- Refine each—To refine attributes of the selected section or metric port one by one
- Read vPlan Refinements —To load already saved exclusion file
- Save vplan Refinements —To save the exclusions to a file for later use
- Unload Refine All/Selected entity—To revert all or selected refinements in the current vManager invocation

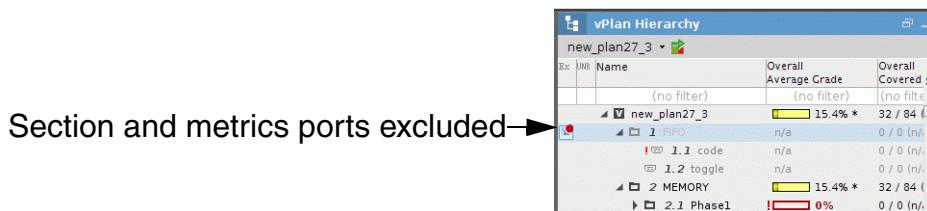
5.10.1 Excluding Sections or Metrics Ports

To exclude a specific section or metrics port from overall vPlan analysis:

1. Select the section or port in the vPlan hierarchy pane. For example, select section *FIFO* in the vPlan Hierarchy pane.
2. Select *Exclude* from the *Refinement* toolbar.

When you exclude a section, the coverage for that section and its metrics ports is removed from the overall grade calculation. In addition, the section and metrics ports are grayed out, and an icon appears along with the section name in the vPlan hierarchy tree, as shown in [Figure 5-57](#) on page 292.

Figure 5-57 Refinement—Exclude



Icon along with the section name indicates that markings are applied to that section.

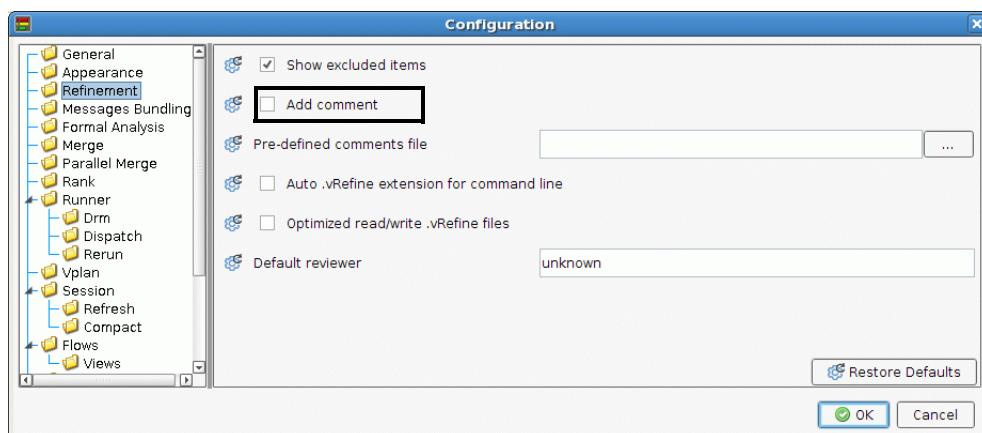
Note: Alternatively, you can exclude a section or metrics port by right-clicking that section or metrics port, and selecting *Exclude* from the pop-up menu.

5.10.2 Adding Comments at the Time of Exclusion

When excluding a section or a port, you can add comments to describe the refinement.

For this, you must set the *Add comment* option in the *Configuration* dialog box, as shown in [Figure 5-58](#) on page 292.

Figure 5-58 Configuration—Refinement Options



By default, the *Add comment* check box is not selected. This indicates that you will not be prompted to add comments at the time of exclusion. After you select the *Add comment* check box, you will be prompted to add comments at the time of exclusion. A dialog box will be displayed which will prompt you to add comments.

Note: Once you enable the *Add comment* check box, it applies to all the exclusions done in that connection of vManager.

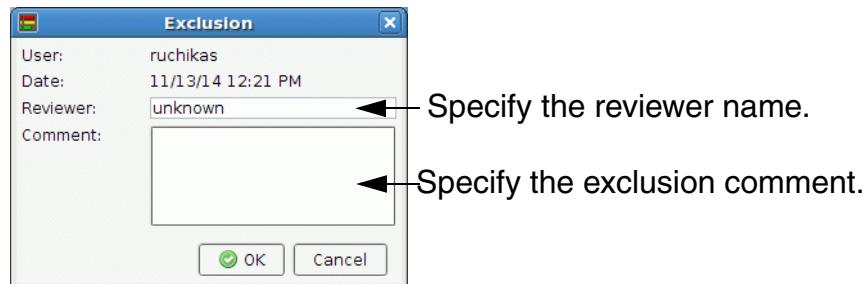
Consider an example. To mark the port `always_fail` as exclude and to add user comments at the time of exclusion, perform the following steps:

1. Select *Configuration* from the *View* menu.
2. Click the *Refinement* folder.
3. Select the *Add comment* check box and click *OK*.

Note: Once you enable the *Add comment* check box, it applies to all the exclusions done in that connection of vManager. As a result, steps 1, 2, and 3 need not be repeated for each exclusion.

4. Select the port in the vPlan Hierarchy pane. For example, select `always_fail` in the verification hierarchy pane.
5. Select *Exclude* from the *Refinement* toolbar. When you click the *Exclude* button, the *Exclusion* dialog box is displayed, as shown in [Figure 5-59](#) on page 293 for adding user comments.

Figure 5-59 Refinement—Exclusion with Comments



In the *Exclusion* dialog box, you can specify the name of the reviewer and also the comments to describe the reason for excluding the specified entity. The *Exclusion* dialog box also displays two non-editable fields, *User* and *Date*. These values are picked automatically from the system on which the tool is running.

6. By default, the reviewer is specified as `unknown`. To specify the reviewer name as `Andrew`, delete the text `unknown`, and specify `Andrew` in the *Reviewer* field. You can also set a default reviewer once in the *Configuration* dialog box and it will appear automatically in the *Exclusion* dialog box. By doing so, you will not have to enter it every time you make exclusion. For more details, see [Configuring Refinement Options](#) on page 58.

Note: The reviewer name cannot be more than 24 characters.

7. For example, to specify the comment as, *The instance seems irrelevant for analysis*, specify this text in the *Comment* field. Comment cannot be more than 500 characters.

Note: In case you have specified a comments file in the *Pre-defined comments file* of the *Configuration* dialog box, you will see additional fields, as shown in [Figure 5-60](#) on page 294.

Figure 5-60 Refinement—Exclusion from Comments File

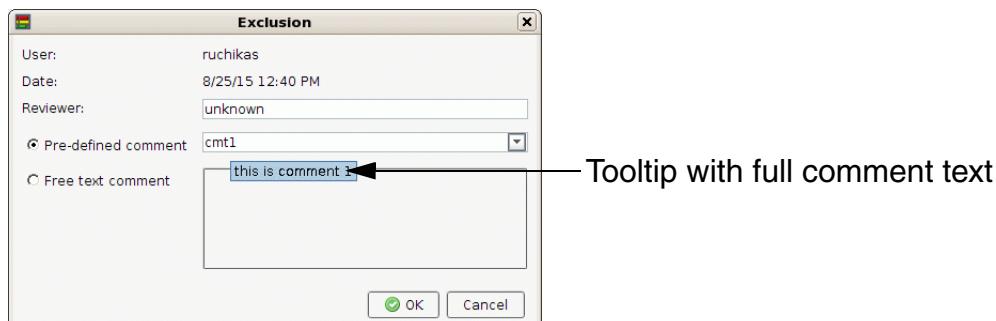


The *Pre-defined comment* drop-down lists all the comments that are defined in the comments file (that you specified in the Configuration dialog box). You can place the cursor on the comment name that appears in the drop-down list. It will show you the associated comment text as a tooltip. In this case, the underlying comments file has the following text:

```
NAME, COMMENT  
cmt1, this is comment 1  
cmt2, this is comment 2
```

When you place the cursor on `cmt1` in the *Exclusion* dialog box, the associated comment is displayed, as shown in [Figure 5-61](#) on page 294.

Figure 5-61 Refinement—Exclusion from Comments File

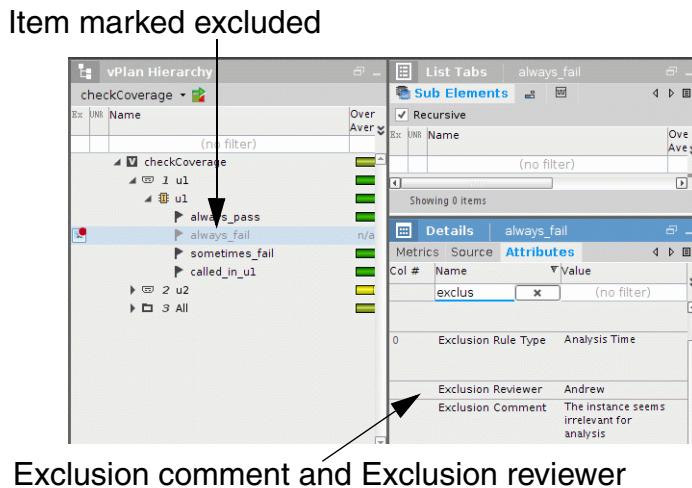


In case you do not want to add comment from the pre-defined list of comments, click the *Free text comment* radio button and specify the comment in the text field.

8. Click **OK**.

After performing the above steps, the port *always_fail* is excluded. In addition, the values specified in the *Exclusion* dialog box are displayed in the *Attributes* tab page, as shown in [Figure 5-62 on page 295](#).

Figure 5-62 Adding User Comments



You can add the required attributes to the table shown in the *vPlan Hierarchy* pane and the *List tabs* pane, as required. Once you add the attributes to required tables, you can apply filters, sort data, and also save it as a view.

Note: You can also add comments for multiple items simultaneously. When adding comments for multiple items, the comments added will apply to all of the selected items.

To turn off adding user comments at the time of exclusion, clear the *Add comment* check box in the *Configuration* dialog box.

The comments added at the time of exclusion are preserved and are available for later use if the exclusions are saved to a refinement file. For more details on saving a refinement file, see [Saving Refinements on page 301](#).

5.10.3 Editing User Comments

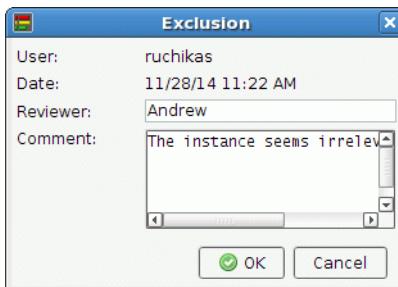
After adding user comments, you can edit the comments, as required.

To edit user comments, select the excluded item whose comments you want to edit, and then click the *Edit Comment* button on the *Refinement* toolbar.

Note: Alternatively, you can select the excluded item whose comments you want to edit, right-click and select *Edit Comment for exclusion* from the pop-up menu.

This opens the *Exclusion* dialog box, as shown in [Figure 5-63](#) on page 296.

Figure 5-63 Editing User Comments

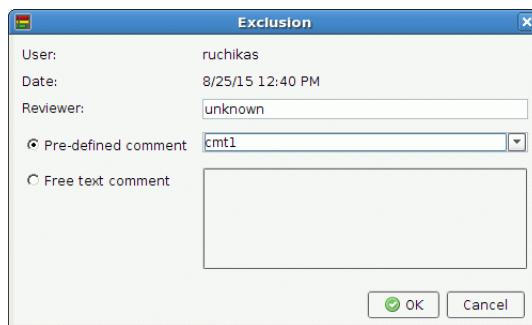


You can edit the reviewer and comment, as required, and click on *OK*.

You can also delete the reviewer and comment by removing the values in the respective fields.

Note: In case you have specified a comments file in the *Pre-defined comments file* of the *Configuration* dialog box, you will see additional fields in the *Exclusion* dialog box at the time of editing comments, as shown in [Figure 5-64](#) on page 296.

Figure 5-64 Refinement—Editing Comments



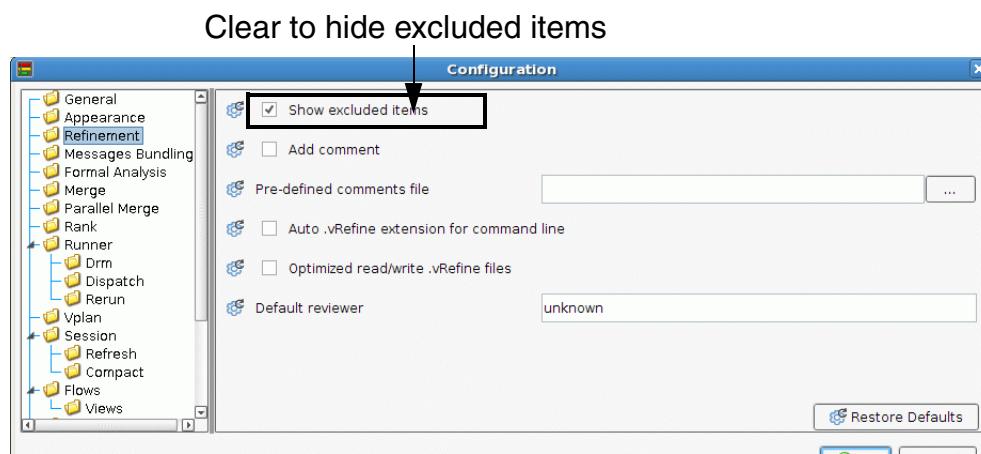
Note: If an old refinement file (from previous releases) is loaded, then at the time of editing user comments, the *User* and *Reviewer* fields show the value as *unknown*, *Date* field is set to default 01/01/1976, and the *Comment* field does not show any value.

5.10.4 Hide Excluded Items

When you exclude a section or a metrics port, then by default that section or metric port still appears in the hierarchy (though the coverage for that item is excluded from overall grade). However, you can hide the excluded sections or metrics ports and not show those items in the vPlan hierarchy.

For this, you need to clear the *Show excluded items* check box in the *Configuration* dialog box, as shown in [Figure 5-65](#) on page 297.

Figure 5-65 Configuration—Hide Excluded Items



After you clear the *Show excluded items* check box, the excluded items will not be shown in the vPlan hierarchy.

5.10.5 Un-Excluding Excluded Items

To un-exclude an already excluded item:

1. Select the excluded section or metrics port in the vPlan hierarchy pane. For example, select *FIFO* in the *vPlan Hierarchy* pane.
2. Select *Un-Exclude* from the *Refinement* toolbar.

Note: Alternatively, you can right-click the section or metrics port and select *Un-Exclude* from the pop-up menu.

After selecting *Un-Exclude*, the coverage for that instance is included in the overall grade.

Similarly, you can un-exclude specific metrics ports, as required.

Note: You can un-exclude multiple items together by selecting them in the *Sub elements* table. You cannot make multiple selections in the *vPlan Hierarchy* pane.

5.10.6 Refining vPlan Attributes

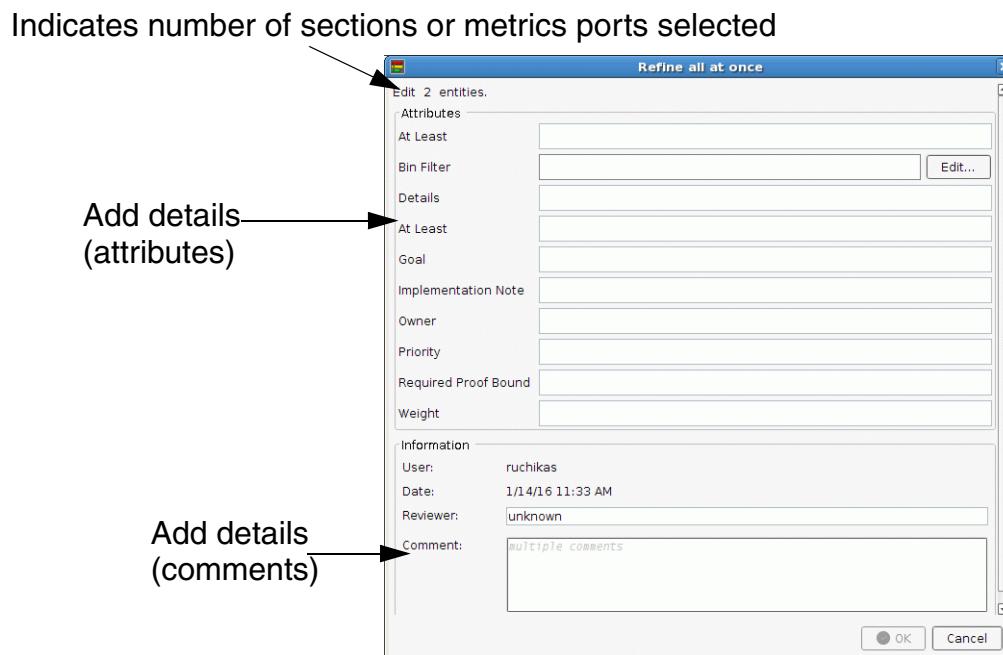
The *Refinement* toolbar has following options that allow you to modify attributes or add comments and description to the selected section or metrics port of the loaded verification plan.

- All at once—To refine attributes of the selected sections or metric ports together
- Refine each—To refine attributes of the selected sections or metric ports one by one

5.10.6.1 All at once

The *All at once* option allows you to add or modify details to all of the selected sections or metrics ports at once. [Figure 5-66](#) on page 298 shows the *Refine all at once* dialog box.

Figure 5-66 Refine all at once

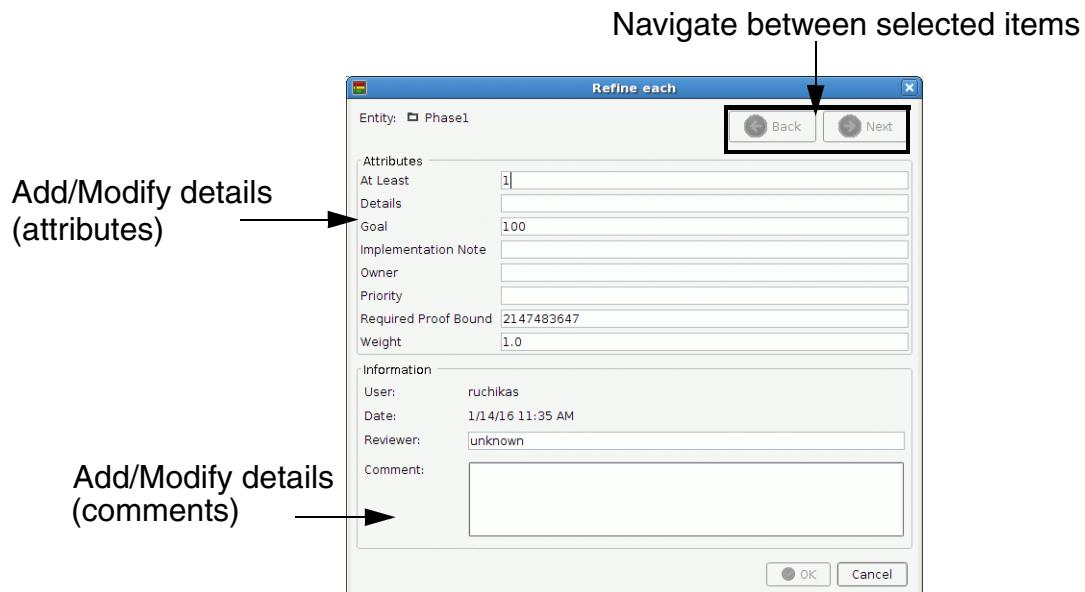


The information added using this dialog box, applies to all of the selected items.

5.10.6.2 Refine each

The *Refine each* option allows you to add or modify attributes of all of the selected sections or metrics ports one by one. [Figure 5-67](#) on page 299 shows the *Refine each* dialog box.

Figure 5-67 Refine each



Using this dialog box, you can add or modify existing values for the selected items individually. After adding or modifying details, you can navigate to the next or previous item using the *Back* or *Next* buttons.

Editing Bin Filters from vManager

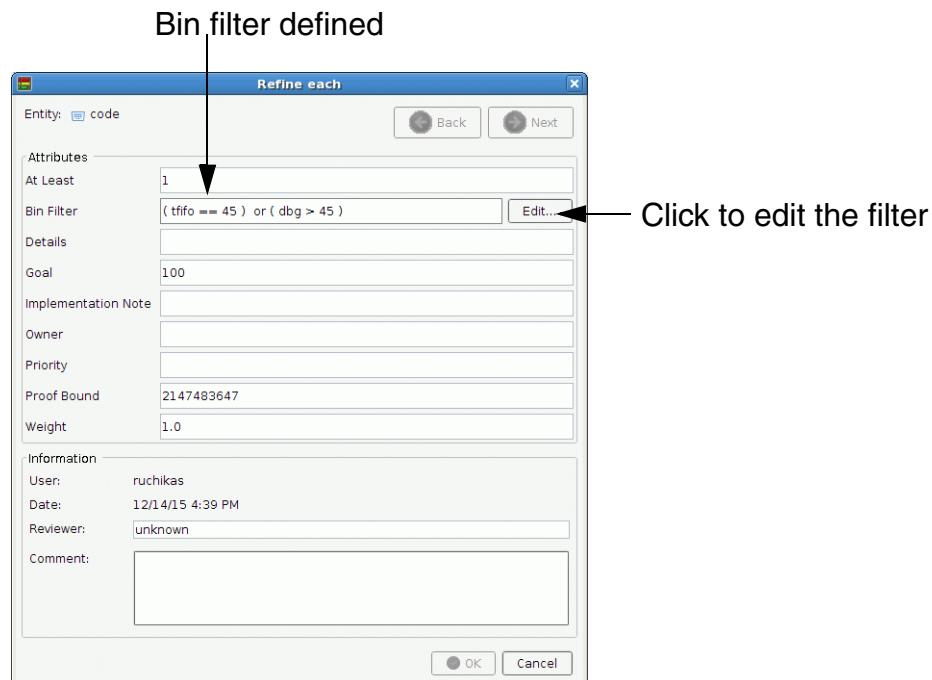
vManager allows you to edit bin filters that were defined in vPlanner.

To edit the bin filters, perform the following steps:

1. (Optional) Launch the *vPlan_BucketsFiltered* view from the Views drop-down to list the items that have bin filters defined.
2. Select the item that has bin filter defined.
3. Right-click and select *Refine each*.

Figure 5-68 on page 300 shows the *Refine each* dialog box.

Figure 5-68 Refine each



4. Click Edit button next to the *Bin Filter* field.

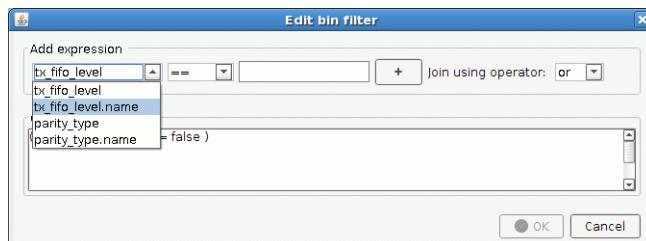
The *Edit bin filter* dialog box is displayed, as shown in [Figure 5-69](#) on page 300.

Figure 5-69 Edit bin filter



5. You can edit the filter, as required. You can modify the expression by selecting required options from the drop-down lists available in the *Add expression* section. For example, if you want to change the filter to a name-based filter, then click the drop-down arrow and select `tx_fifo_level.name`, as shown in [Figure 5-70](#) on page 301.

Figure 5-70 Edit bin filter



6. You can build/edit the query as required, and click the *OK* button.

The modification you make is taken into account immediately, and affects both the appearance of the vPlan hierarchy tree and the grade calculation.

5.10.7 Revert vPlan Refinements

To revert the vPlan refinements made in the current invocation of vManager, select *Unload Refine* drop-down from the *Refinement* toolbar. This drop-down has following options:

- *Unload Refine*—To revert all the vPlan refinements made in the current invocation of vManager. This is the default option.
- *Unload Refine on selected entities*—To revert vPlan refinement only from the selected entity.

5.10.8 Saving Refinements

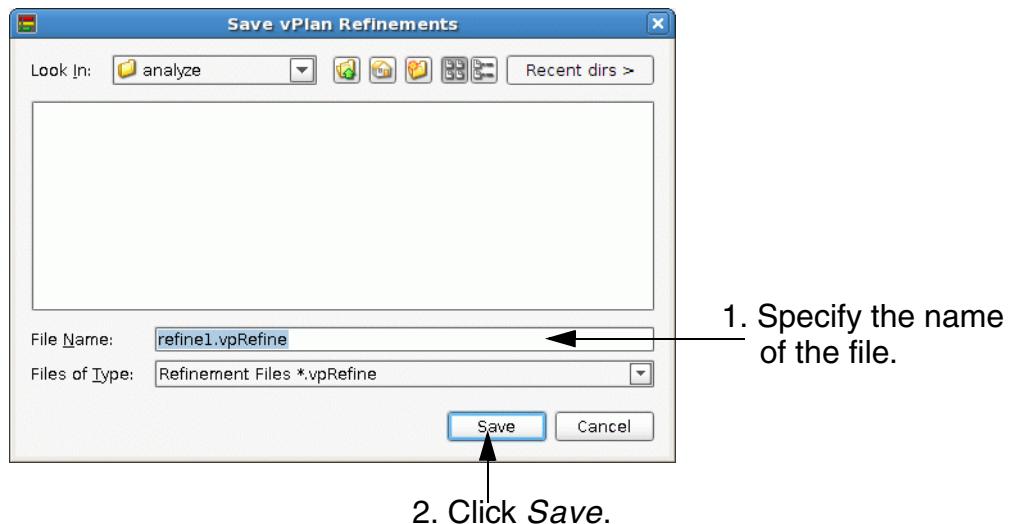
After excluding sections or ports during analysis, you can save the refinements to a file and later, in another vManager invocation, load the saved refinements. This helps in saving the effort of making exclusions again in the next vManager invocation.

To save the refinements made in one vManager invocation for reuse in another invocation:

1. Click *Refinement* in the toolbar and select *Save vplan Refinements*.

The Save vPlan Refinements dialog box appears, as shown in [Figure 5-71](#) on page 302.

Figure 5-71 Save vPlan Refinements



2. By default, the file name is shown as `refine1.vpRefine`. To specify a different name, specify the name of the vplan refinements file in the *File Name* text box and click *Save*. For example, specify the name as `my_exclude`.

Note: The extension of a vPlan refinement file is always `.vpRefine`.

In this case, when you save the file, the file will be named as `my_exclude.vpRefine`.

Note: The refinements saved in GUI mode can be loaded and applied in the batch mode of vManager and vice versa. In addition, refinement file saved with this version of the tool will not be supported in any prior version of the tool.

For details on saving vPlan refinements in CLI mode, see [save](#) on page 427.

5.10.9 Loading Saved Refinements

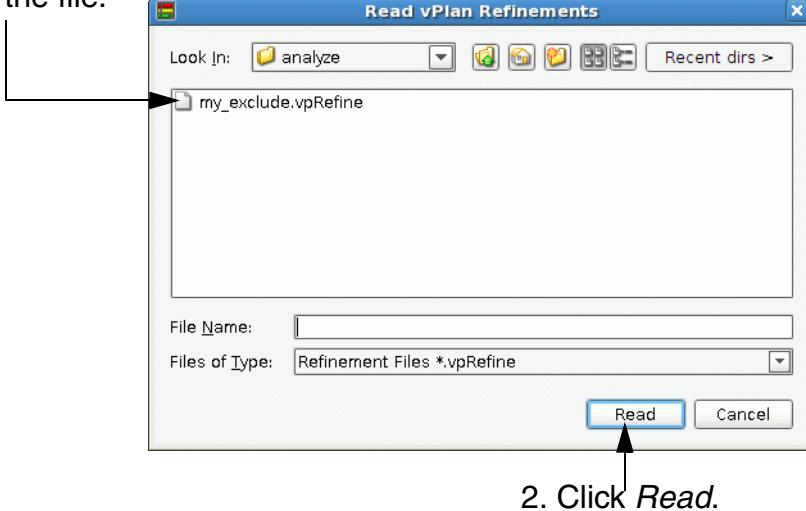
To load the already saved refinements for reuse in another vManager session:

1. Click *Refinement* in the toolbar and select *Read vPlan Refinements*.

The Read vPlan Refinements dialog box appears, as shown in [Figure 5-72](#) on page 303.

Figure 5-72 Read Refinement Rules

1. Select the file.



2. Click *Read*.

2. Select the vPlan refinements file you want to load, and click *Read*.

When you load the vPlan refinements file, the saved exclusions are applied to the loaded vPlan.

 **Important**

If a vPlan has mapped entities and the underlying source code of the mapped entities has changed, then at the time of loading the vPlan refinement file, you are prompted with a warning to indicate that some of the rules might be invalid. For more details, see [Reusing Refinements File](#) on page 304.

Note: Instead of loading the saved refinements after loading a vPlan, you can also ensure that the vPlan refinements are loaded automatically along with the vPlan when the vPlan is loaded again.

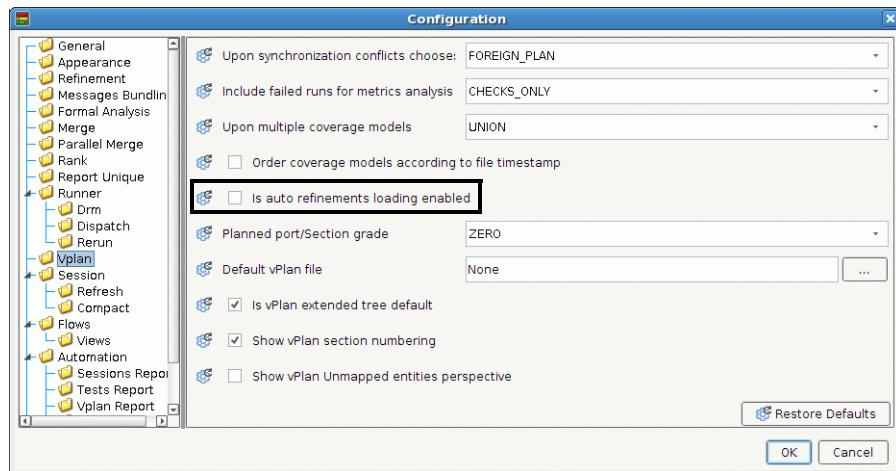
5.10.9.1 Automatically Loading Saved Refinements at the Time of Loading the vPlan

To ensure that the saved refinements are loaded automatically when loading the vPlan:

1. Select *Configuration* from the *View* menu.
2. Click the *Vplan* folder.

The vPlan options are displayed, as shown in [Figure 5-73](#) on page 304.

Figure 5-73 Configuration—vPlan Options



3. To enable automatically loading the refinements file select the *Is auto refinements loading enabled* check box.
4. Click **OK**.

This will ensure that next time when the vPlan is loaded in the same vManager invocation or the next vManager invocation the refinements file is also loaded along with it.

Note: If for a particular vPlan, there are more than one refinement files, then at the time of loading the vPlan, all of the refinement files are loaded along with it.

Note: For details on other options on this dialog box, see [vPlan Configuration Options](#) on page 270.

5.10.10 Reusing Refinements File

If a vPlan has mapped entities and the underlying source code of the mapped entities has changed, then at the time of loading the vPlan refinement file, you are prompted with a warning to indicate that some of the rules might be invalid, as shown in [Figure 5-74](#) on page 304.

Figure 5-74 Reuse vPlan



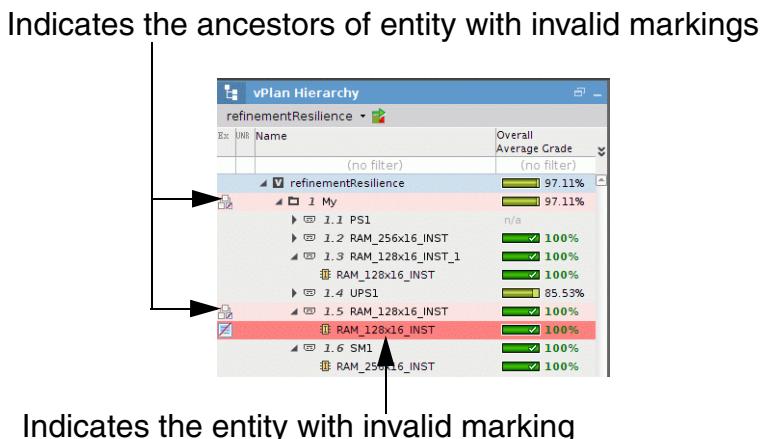
You can choose any of the following:

- Apply all of the rules: This will apply all of the rules from the refinement file. However, in this case, there is a possibility of incorrect markings.
- Apply only the valid rules: This will apply only the valid rules from the refinement file and also highlight the invalid/orphan rules so that you can review them and then clear or apply, as required.
- Cancel the operation: This will cancel the loading of refinement file.

When you select *Apply only valid rules*, then the valid rules are applied.

For the entities with invalid markings, an icon is shown in the *Ex* column, as shown in [Figure 5-75 on page 305](#).

Figure 5-75 Entities with Invalid Markings

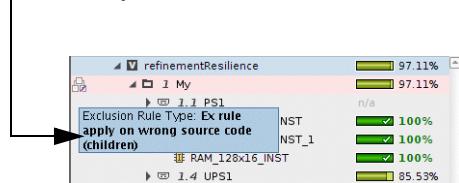


You can place the cursor on the icon shown in the *Ex* column of the entity for more details.

[Figure 5-76 on page 306](#) shows the tooltip shown when you place the cursor on the icon shown in the *Ex* column.

Figure 5-76 Tooltip for Invalid Markings

Tooltip shown for the parent with invalid marking



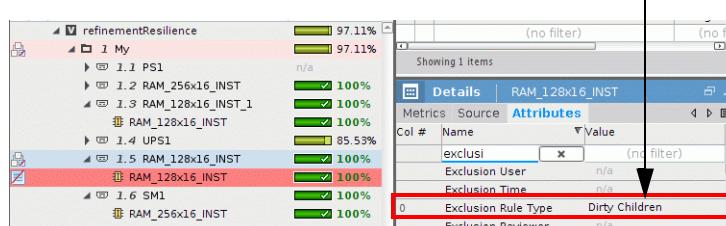
Tooltip shown for the entity with invalid marking



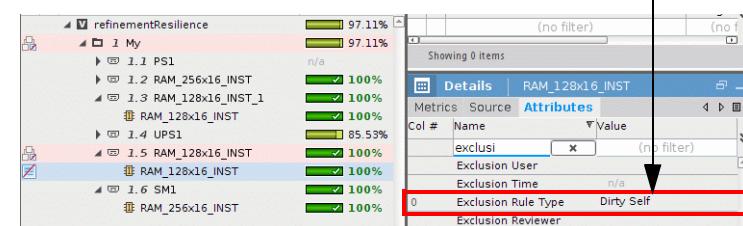
In the tooltip, text *(self)* indicates that the selected entity has invalid marking and text *(children)* indicates that some of the children might have incorrect markings. In addition, new values are added to the *Exclusion Rule Type* attribute, as shown in Figure 5-77 on page 306.

Figure 5-77 Exclusion Rule Type value for Invalid Markings

Exclusion Rule Type (parent of entity with invalid marking)



Exclusion Rule Type (entity with invalid marking)



The *Exclusion Rule Type* attribute shows value — Dirty Self for the entity with invalid marking and it shows value — Dirty children for the parent of the entity with invalid marking.

Note: The dirty rules are considered orphans.

At this stage, you have an option of:

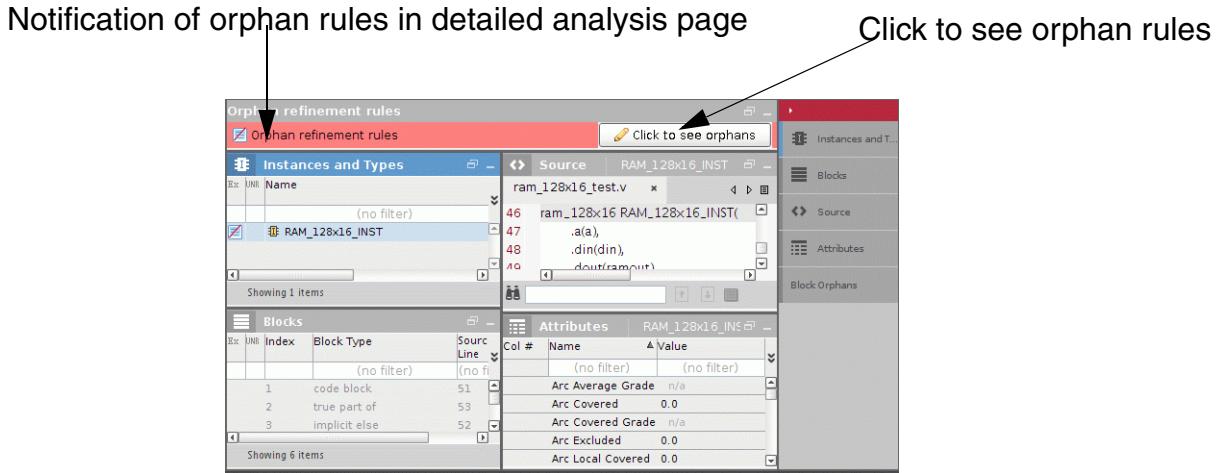
- Not reviewing each rule individually and can clear or approve all the rules in the Verification Hierarchy pane together. For this, right-click on a instance or type highlighted as red and select any of the following actions:
 - Clear all orphan rules* — To clear all the orphan rules of the selected item.
 - Apply all orphan rules* — To apply all the orphan rules of the selected item and delete them.

Note: The *Apply all orphan rules* option will apply only the rules for which the index can be found. If index is not found, then no action is taken for the orphan rule. If some of the rules are marked dirty and you want to approve or clear them, then right-click the required instance or type and select any of the following:

- Clear dirty rules—To clear/delete the dirty rules of the selected instance or type.
 - Approve dirty rules—To approve the dirty rules of the selected instance or type (make the rules completely valid).
- Reviewing each rule individually and then approving it or clearing it, as required. For this, open the detailed analysis page. For example, select *RAM_128x16_TEST_INST*, right-click and select *vPlan Block Analysis*.

[Figure 5-78](#) on page 308 shows the Block analysis page of instance *RAM_128x16_TEST_INST*.

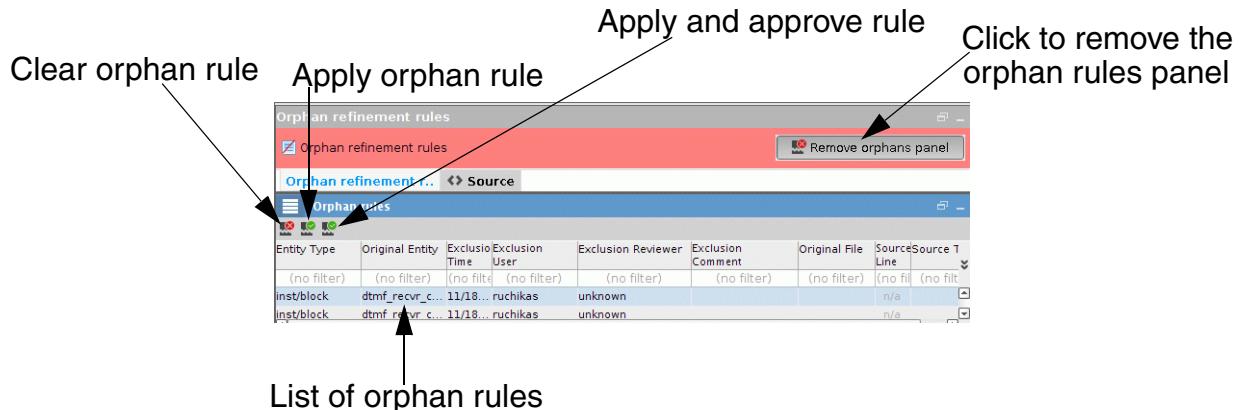
Figure 5-78 Orphan Rules Notification



1. Click the *Click to see orphans* button to review the orphan rules.

[Figure 5-79 on page 308](#) shows the details of the orphan rules.

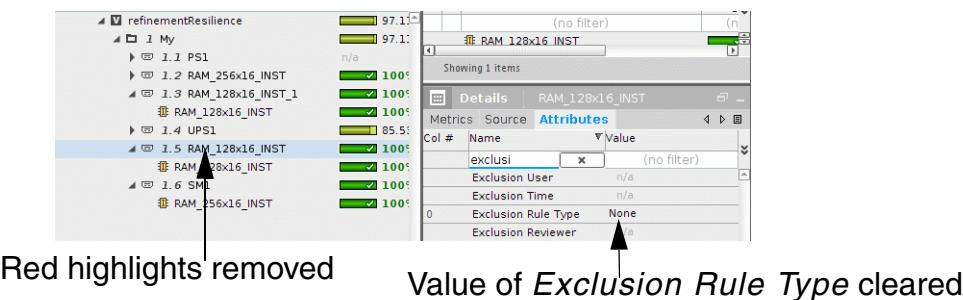
Figure 5-79 Orphan Rules Review



2. You can review each orphan rule and perform any of the following actions from the buttons available in the table or right-click the rule and select these options from the popup menu:
 - ❑ Clear — Deletes the selected orphan rule.
 - ❑ Apply—Applies the selected orphan rule and deletes it. This option will apply only the rules for which the index can be found. If index is not found, then this option is disabled.

- ❑ Apply and approve— Applies the selected rule and approves it (makes the rule completely valid). This option will apply and approve only the rules for which the index can be found. If index is not found, then this option is disabled.
3. When all orphan rules are cleared, the dirty flags (red highlights) on the related types/instances are automatically removed, as shown in [Figure 5-80](#) on page 309.

Figure 5-80 Orphans Cleared



After clearing the orphan rules, you can save the vPlan refinements file as any of the following:

- As a new refinement file: If you save the refinement file as a new refinement file, then the new refinement file will not have orphan rules and they will not be reported on loading. The original refinement file will continue to have orphan rules and if you read the original refinement file, then the orphan rules will be reported.
- Overwrite the existing refinement file: If you overwrite the existing refinement file after clearing all the orphan rules, then next time you load the refinement file, the orphan rules will not be reported.

5.11 Generating vPlan Reports

Using vManager, you can generate a vPlan report in an HTML format and then publish it on the internet. Generating an HTML report is useful for users who want to make the results available for viewing to people who might not have vManager installed with them.

The *Report* toolbar has a drop-down button named *Report* using which you can generate reports from GUI. The *Reports* option is also available in the *Analysis* menu using which you can generate reports.

The *Report* drop-down button on the *Report* toolbar has the following options for generating reports:

- Report All (Default)—To generate a vPlan report for the nodes in a vPlan.

- Report on Selected Entity—To generate a vPlan report for a selected node.
- Summary Report—To generate a summary report. For more details, see [Generating Summary Report](#) on page 51.

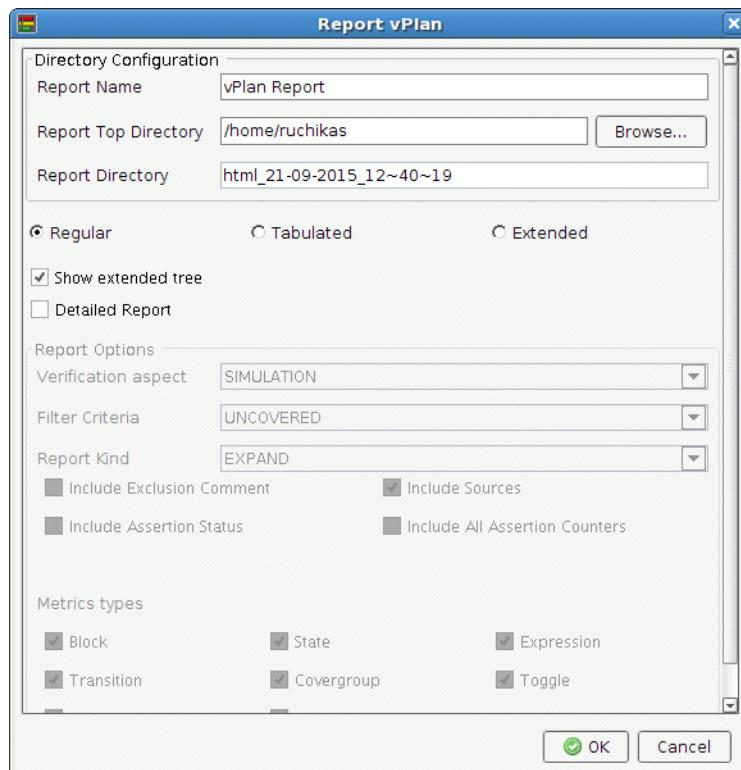
Note: The report action considers the configuration option *Show extend metrics tree* while generating the report. If the *Show extend metrics tree* option is enabled (using the *Configuration* dialog box in GUI or CLI `config` command), then the covergroups, cover items, FSMs, and assertions are also listed in the metrics hierarchy tree in the report.

For example, to generate a vPlan report for all the nodes:

1. Click the *Report* button in the *Report* toolbar.

The *Report vPlan* dialog box is displayed, as shown in [Figure 5-81](#) on page 310.

Figure 5-81 Report vPlan



2. Specify the name of the report in the *Report Name* text box. By default, the name is specified as *vPlan Report*. This name appears on the HTML report. For example, specify the name as *All_Nodes_Report*.

3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`. For example, change the name as `vp_report`.
5. Specify the type of report to be generated. It can be any of the following:
 - Regular: This is the default report. It lists nodes and sub-nodes and provides hyperlinks for navigating further. For more details, see [Example: Regular Report](#) on page 313.
 - Tabulated: This generates a bin-level tabulated report. This report contains *Metrics* or *vPlan* tree and bin details in tables side-by-side. For more details, see [Example: Bin-Level Tabulated Report](#) on page 313.
 - Extended: This option generates an extended report. In this report, the *Metrics* tree or *vPlan* tree and bins information for each of the cover items is shown in the tree. For more details, see [Example: Bin-Level Extended Report](#) on page 315.
6. Select the *Show extended tree* check box to list covergroups, cover items, FSMs, and assertions along with the types and instances in the hierarchy tree. If you want to list only the types and instances in the hierarchy tree, then clear this check box.
7. To generate a detailed report, select the *Detailed Report* check box. This enables the *Report Options* section of the *Report Metrics* dialog box.
8. In the *Verification aspect* drop-down list, specify the assertion properties to be shown in the report.
 - Simulation shows only simulation assertion properties in the report.
 - Formal shows only formal assertion properties in the report.
 - Both shows both simulation and formal assertion properties in the report.
9. You can specify the items that should be included in the report from the *Filter Criteria* drop-down list. Filter criteria can be *Covered*, *Uncovered*, *Both* (covered and uncovered), *Excludes*, or *All*. For example, select *All* to report all items.
10. You can specify the kind of report in the *Report Kind* drop-down list. Report kind can be *Expand* (prints bin information as a list) or *Aggregate* (prints the cross tuple information in a tabular format after grouping similar bins).
11. By default, the *Include Exclusion Comment* check box is not selected. This means that in the detailed report, the exclusion comments and reviewer information will not be shown. In case, you want to show this information in the report, select the *Include Exclusion Comment* check box.

12. By default, the *Include Sources* check box is selected. This means that in the detailed report, the source information will be shown in the *Source code* column. In case, you want to omit this column in the report, deselect the *Include Sources* check box.

Note: In case you have entities from the compressed files, then even if the *Include Sources* check box is selected, the *Source Code* column will be blank for entities from compressed files.

13. By default, the *Include Assertion Status* check box is not selected. This means that in the detailed report, the *Status* column, which shows the status of the assertion (Passed, Failed, Other) will not be shown in the report. In case, you want to show this information in the report, select the *Include Assertion Status* check box.

14. By default, the *Include All Assertion Counters* check box is not selected. This means that in the detailed report, the *Vacuous*, *Attempt*, and *Disabled* counters will not be shown. In case, you want to show these counters in the report, select the *Include All Assertion Counters* check box.

Note: The *Vacuous*, *Attempt*, and *Disabled* counters are reported only if the `-abvrecordvacuous` switch is used at the time of simulation run. In case the `-abvrecordvacuous` switch is not used at simulation, then the *Vacuous*, *Attempt*, and *Disabled* columns will show n/a. For more details on the `-abvrecordvacuous` switch, see the *Verilog Compilation Command-Line Options* user guide.

15. From the *Metrics types* section, select the metrics types for which you want to generate report. To omit a metrics type, clear the respective check box.

16. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 5-82](#) on page 312.

Figure 5-82 Report Done



This dialog box shows the location where the report is generated.

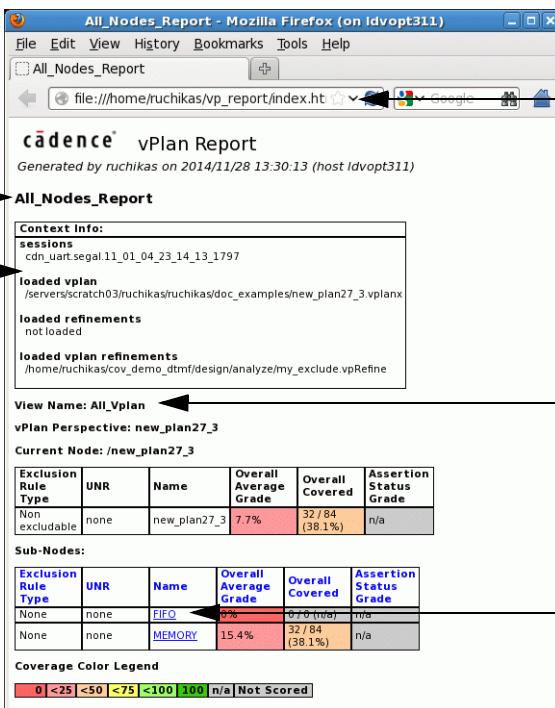
17. Click *OK*.

You plan to view the report later, you can navigate through the location shown above, and then open the `index.html` file in a Web browser.

Example: Regular Report

Figure 5-83 on page 313 shows a regular detailed report.

Figure 5-83 vPlan Report (Regular)



The screenshot shows a Mozilla Firefox window with the title "All_Nodes_Report - Mozilla Firefox (on Idvopt311)". The address bar shows the URL "file:///home/ruchikas/vp_report/index.htm". The main content area displays a "cadence vPlan Report" generated by "ruchikas" on "2014/11/28 13:30:13 (host Idvopt311)". The report includes "Context Info" such as sessions and loaded vplans. It shows a table of "Exclusion Rule Type" and "UNR" for nodes like "new_plan27_3". A "Sub-Nodes" table is also present. A "Coverage Color Legend" at the bottom maps colors to coverage ranges: 0, <25, <50, <75, <100, n/a, and Not Scored.

- Report name → All_Nodes_Report
- Context information → Context Info: sessions, loaded vplan, loaded refinements, loaded vplan refinements
- View used to generate report → View Name: All_Vplan
- Click on the node to view detailed report → Sub-Nodes table (highlighted row: None, FIFO, 0%, 0/0 (0%), n/a)

The top-level page shows information such as, who generated the report and when the report was generated. It also shows the view used to generate the report and the node selected at the time of generating the report. If a node has sub-nodes, then the sub-nodes are reported in a separate table as hyperlinks. You can navigate through the report by clicking on various hyperlinks.

You can sort the table data by clicking the column header.

The columns shown in the table are based on the columns available or view selected at the time of generating the report. For example, if a particular column is removed from the table in vManager, then the HTML report will also not include that column.

You can also generate this report from the CLI mode using the report_vplan command.

Example: Bin-Level Tabulated Report

Figure 5-84 on page 314 shows a bin-level tabulated report.

Figure 5-84 vPlan Report (Tabulated)

The screenshot shows a Firefox browser window displaying the 'vPlan Report' page. The title bar says 'vPlan Report - Mozilla Firefox (on Idvopt311)'. The address bar shows 'file:///home/ruchikas/html_24-04-2015_12~27~57/index.html'. The page content includes:

- Context information:** sessions: cdn_uart_segal.11_01_04_23_14_13_1797; loaded vPlan: /servers/scratch03/ruchikas/ruchikas/doc_examples/new_plan27_3.vplanx; loaded refinements: not loaded.
- Report name:** cadence® vPlan Bin LevelReport. Generated by ruchikas on 2015/04/24 12:28:05 (host Idvopt311).
- Report location:** vPlan Report
- vPlan Report:** A table showing the vPlan hierarchy. The columns are: Exclusion Rule Type, UNR, Name, Overall Average Grade, Overall Covered, Assertion Status Grade. The rows include:
 - Non excludable, none, new_plan27_3, 7.7%, 32 / 84 (38.1%), n/a
 - None, none, FIFO, 0%, 0 / 0 (n/a), n/a
 - None, none, MEMORY, 15.4%, 32 / 84 (38.1%), n/a
 - None, none, Phase1, 0%, 0 / 0 (n/a), n/a
 - None, none, covergroup, n/a, 0 / 0 (n/a), n/a
 - None, none, tx_fifo_level, n/a, 0 / 0 (n/a), n/a
 - None, none, parity_type, n/a, 0 / 0 (n/a), n/a
 - None, n/a, assertion, 0%, 0 / 0 (n/a), n/a
 - None, none, Phase2, 30.8%, 32 / 84 (38.1%), n/a
- Bins within selected coveritem:** A table showing bins for the selected 'assertion' item. The columns are: Exclusion Rule Type, UNR, Name, Overall Average Grade, Score. The rows include:
 - None, none, Empty, n/a, 32
 - None, none, 1 to 31, n/a, 320
 - None, none, FIFO full, n/a, 0
 - None, none, others, n/a, 0

Annotations on the screenshot:

- An arrow points from the text 'Click on the coveritem to view its bins' to the 'assertion' row in the left table.
- An arrow points from the text 'Bins within selected coveritem' to the right table.

The report shows information such as, who generated the report and when the report was generated. It also shows details such as sessions, loaded vPlan, and refinement files.

The report shows two tables. The first table shows the vPlan hierarchy. You can view the bin-level information by selecting a cover item in the vPlan hierarchy. When you select a cover item, the bins associated with the cover item are displayed in the table shown next to it.

The columns shown in the table are based on the columns available or view selected at the time of generating the report.

Both the tables in the report allows you to filter data, as required.

Note: Tabulated reports are supported in Firefox, Safari, and Internet Explorer. Chrome does not support loading data from local file system by default.

You can also generate this report from the CLI mode using the report_vplan command.

Example: Bin-Level Extended Report

Figure 5-85 on page 315 shows an extended report.

Figure 5-85 vPlan Report (Extended)

loaded refinements
not loaded

loaded vplan refinements
not loaded

View Name: All_Vplan
vPlan Perspective: new_plan27_3
[Expand all](#) [Collapse all](#)

Exclusion Rule Type	UNR	Name	Overall Grade	Average Covered	Assertion Status Grade
Non excludable	none	new_plan27_3	7.7%	32 / 84 (38.1%)	n/a
None	none	FIFO	0%	0 / 0 (n/a)	n/a
None	n/a	code	0%	0 / 0 (n/a)	n/a
None	n/a	toggle	0%	0 / 0 (n/a)	n/a
None	none	MEMORY	15.4%	32 / 84 (38.1%)	n/a
None	none	Phase1	0%	0 / 0 (n/a)	n/a
None	none	Phase2	30.8%	32 / 84 (38.1%)	n/a
None	none	toggle	61.61%	32 / 84 (38.1%)	n/a
None	none	mon_frame_done	61.61%	32 / 84 (38.1%)	n/a
None	none	parity_type	50%	2 / 4 (50%)	n/a
None	none	NONE	0%	0 / 1 (0%)	n/a
None	none	ODD	100%	1 / 1 (100%)	green
None	none	EVEN	100%	1 / 1 (100%)	green
None	none	SPACE	0%	0 / 1 (0%)	n/a
None	none	stopbit_type	100%	2 / 2 (100%)	n/a
None	none	ONE	100%	1 / 1 (100%)	green
None	none	TWO	100%	1 / 1 (100%)	green
None	none	databit_type	75%	3 / 4 (75%)	n/a
None	none	FIVE	0%	0 / 1 (0%)	n/a

Bin information shown in hierarchy tree

The report shows a single table. In this report, the *Metrics* tree and the bins information for each of the cover items is shown in the tree. You can expand or collapse the hierarchy tree using the links (*Expand all* and *Collapse all*) shown on the page.

Note: The expand/collapse capability in an extended report is supported only for trees with less than 4K items.

The columns shown in the table are based on the columns available or view selected at the time of generating the report.

Note: This report is a single HTML file and can be easily shared over mail in batch mode.

You can also generate this report from the CLI mode using the [report_vplan](#) command.

5.12 vPlan Visitor Interface

A visitor is a general design pattern used in software, described in detail in *Design Patterns* by Erich Gamma, Richard Helm, et al. The intent of a visitor is to “represent an operation to be performed on the elements of an object structure. A visitor lets you define a new operation without changing the classes of the elements on which it operates.”

The vPlan visitor interface is mainly used for creating a fully customized report or exporting the vPlan data to some format.

5.12.1 Using the vPlan visitor Interface

To use the vPlan visitor interface, you must be familiar with TCL scripting, verification plans, and vPlanner.

In vManager, the vPlan visitor interface is based on TCL scripting. This interface enables you to scan the vPlan and extract all data (attributes) of the elements in the vPlan tree, including the mapped elements. For this, you need to:

1. Implement the Required Methods in a TCL Script File
2. Run the script using CLI vscan command

Note: The vPlan visitor does not apply on user-defined attributes.

5.12.1.1 Implement the Required Methods in a TCL Script File

To scan a vPlan, implement the following methods in a TCL script file:

- `startVplanNode {attributes}`— Is called before visiting any sub-nodes or mapped elements.
- `endVplanNode {attributes}`— Is called after visiting all sub-nodes or mapped element.
- `mappedElement {attributes}`— Is called for each mapped element.

In all these methods, `attributes` is a TCL associative array used to store all the attributes of the current vPlan node. Any change done to the array will have no impact on the vPlan data maintained by vManager (the data is read only).

Note: You can get a list of all the attributes available for a node by exporting the attributes to a CSV file from the GUI. For more details on exporting data to CSV file, see [Exporting Table Data \(Sessions, Runs, Failures, Tests, vPlan, Metrics, Snapshots\) to a CSV File](#) on page 49.

Consider an example. The following code in a TCL script will print a list of vPlan nodes along with their overall grades.

```
proc startVplanNode {attributes} {
    upvar attributes attrs
    puts $attrs(NAME)
    foreach {key val} [array get attrs] {
        puts "$key -> $val"
    }
}
```

Note: The keys of the associative array are the attribute names in upper case and underscore as word separator. For example, OVERALL_GRADE. The list of available keys can be extracted using the TCL command `array names <array>`.

5.12.1.2 Run the script using CLI `vscan` command

After implementing the required methods in a TCL script, you can run the TCL script using the `vscan` command. For more details, see [vscan](#) on page 483.

5.12.2 Extending a vPlan visitor Interface

The vPlan visitor interface can be extended such that you can calculate values of user-defined attributes.

User-defined attributes are defined in a CSV file. In this CSV file, a new column `CALCULATION_SCRIPT` is added. In the `CALCULATION_SCRIPT` column, you can specify the path of a vPlan visitor script file, as shown in [Figure 5-86](#) on page 317.

Figure 5-86 CSV File With Attribute `CALCULATION_SCRIPT`

	A	B	C	D	E
1	NAME	DISPLAY_NAME	DESCRIPTION	VALID_CONTAINER_NAMES	<code>CALCULATION_SCRIPT</code>
2	new_1	n1		section;instantiate;metricport	/hm/ben/userscripts/visit1.tcl
3	new_2	n2		section;instantiate;metricport	
4					

vManager uses the specified visitor script file to calculate the value of the user-defined attribute.

In the visitor script file, you must use the predefined TCL command `setAttributeValue` to assign value to the given attribute. For example, the following visitor script file `visit1.tcl` sets the value of attribute `new_1` (shown in [Figure 5-86 on page 317](#)) as the sum of `goal` and `weight`.

```
proc endVplanNode {attributes} {
    upvar attributes attrs;
    setAttributeValue [ expr { $attrs(GOAL) + $attrs(WEIGHT) } ];
}
```

Note: The `setAttributeValue` command is supported only in `startVplanNode` and `endVplanNode` methods.

In case you are extending a vPlan visitor interface to calculate the values of user-defined grade attributes, remember to ensure the following in the CSV file:

- Such attributes should be of type DOUBLE.
- The value of the `IS_GRADE_ATTRIBUTE` is set to TRUE so that these grade attributes are shown in GUI with appropriate colors and bars.

[Figure 5-87 on page 318](#) shows an example of CSV file with relevant modifications in the case of user-defined grade attributes.

Figure 5-87 CSV File for User-defined Grade Attributes

	A	B	C	D	E	F	G
1	NAME	DISPLAY_NAME	DESCRIPTION	VALID_CONTAINER_NAMES	IS_GRADE_ATTRIBUTE	TYPE	CALCULATION_SCRIPT
2	new_1	n1		section;instantiate;metricport	FALSE	ATTRIBUTE_TYPE_STRING	/hm/ben/userscripts/visit1.tcl
3	new_2	n2		section;instantiate;metricport	TRUE	ATTRIBUTE_TYPE_DOUBLE	/hm/ben/userscripts/visit2.tcl
4							

↑ IS_GRADE_ATTRIBUTE set to TRUE

↓ TYPE set as DOUBLE

In this example, the following modifications are made to the CSV file for the user-defined attribute `new_2`:

- `IS_GRADE_ATTRIBUTE` is set to TRUE
- Type of the attribute is set to `ATTRIBUTE_TYPE_DOUBLE`

These modifications are done because `new_2` is a user-defined grade attribute.

Project Tracking

The *Tracking* center allows you to track data over time. Using the *Tracking* center you can track progress of sessions, tests, metrics, and verification plans. You can also generate charts and tracking reports.

This chapter covers the following topics:

- [Launching the Tracking Center](#) on page 319
- [Creating a Configuration](#) on page 325
- [Taking Snapshots](#) on page 331
- [Cumulative Data Tracking](#) on page 340
- [Generating and Editing Charts](#) on page 342
- [Generating Tracking Reports](#) on page 358
- [Converting Regular Chart to Triage or Stacked Chart](#) on page 360

6.1 Launching the Tracking Center

[Figure 6-1](#) on page 320 shows the Welcome screen.

Figure 6-1 Welcome Screen -- Tracking Center

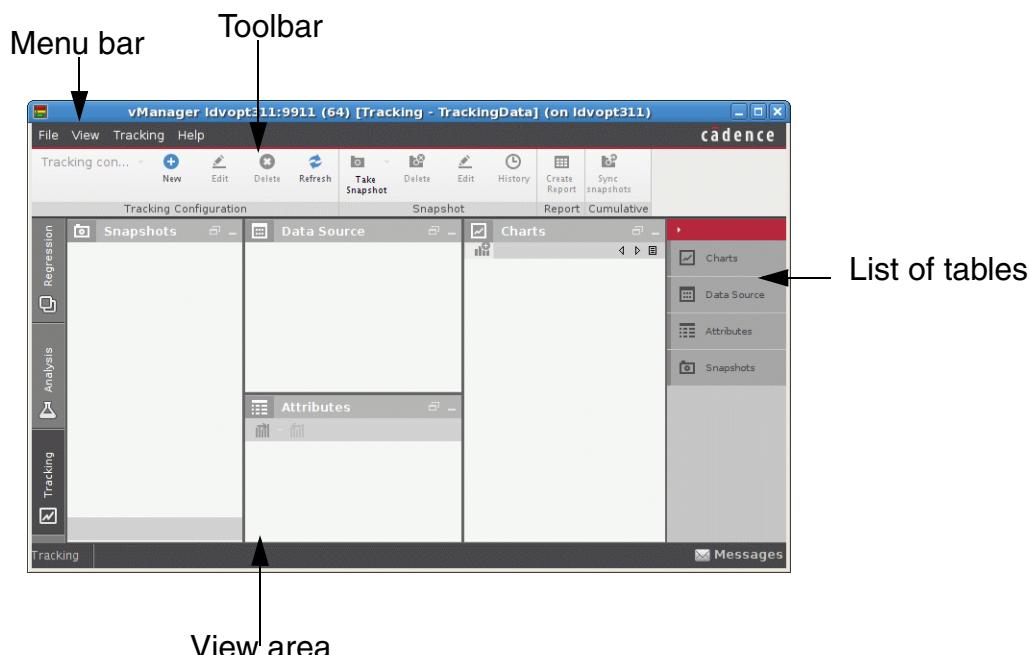


You can launch the *Tracking* center by clicking the *Tracking Center* hyperlink on the Welcome screen.

Note: You can also launch the Tracking center from the Regression and Analysis center by clicking *Tracking* in left pane of the screen.

After you click the *Tracking Center* hyperlink, the *Tracking* center is displayed, as shown in [Figure 6-2](#) on page 320.

Figure 6-2 Tracking Center



The main components of the *Tracking* center are:

- Menu Bar
- Toolbar
- View Area

6.1.1 Menu Bar

[Figure 6-3](#) on page 321 shows the menu bar available in the *Analysis* center.

Figure 6-3 Menu Bar

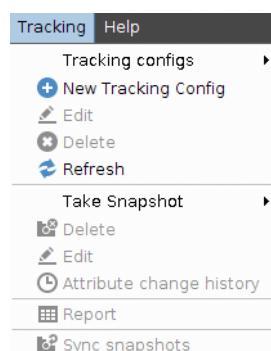


Menu options *File*, *View*, and *Help* are common to all activity centers and are discussed in the [Chapter 1, “Getting Started with vManager”](#).

The *Tracking* menu is related to the *Tracking* center and is discussed here.

[Figure 6-4](#) on page 321 displays the *Tracking* menu.

Figure 6-4 Tracking Menu



From the *Tracking* menu, you can select any of the following options:

- **Tracking configs**—To launch an already available tracking configuration.
- **New Tracking Config**—To create a new tracking configuration. For more details, see [Creating a Configuration](#) on page 325.
- **Edit**—To edit a tracking configuration. For more details, see [Editing a Configuration](#) on page 331.

- Delete —To delete a tracking configuration. For more details, see [Deleting a Configuration](#) on page 331.
- Refresh —To refresh the selected configuration. For more details see [Refreshing a Configuration](#) on page 331.
- Take Snapshot—To take snapshots at regular intervals. For more details, see [Taking Snapshots](#) on page 331.
- Delete —To delete the selected snapshot. For more details, see [Deleting Snapshots](#) on page 338.
- Edit — To modify the snapshot date and title. For more details, see [Editing Snapshots](#) on page 338.
- Attribute change history — To view changes made to attributes of the selected snapshot. For more details, see [Show Snapshot Editing History](#) on page 339.
- Report—To generate tracking reports. For more details, see [Generating Tracking Reports](#) on page 358.
- Sync snapshots—Recalculates the snapshot data and clears all dirty snapshots. For more details, see [Cumulative Data Tracking](#) on page 340.

6.1.2 Toolbar

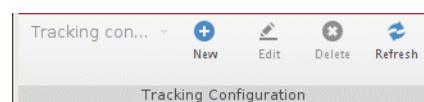
The *Tracking* center has following toolbars:

- [Tracking Configuration](#)
- [Snapshot](#)
- [Report](#)
- [Cumulative](#)

6.1.2.1 Tracking Configuration

[Figure 6-5](#) on page 322 shows the *Tracking Configuration* toolbar of *Tracking* center.

Figure 6-5 Tracking Configuration Toolbar



The *Tracking Configuration* toolbar has following options:

- Tracking configs—To launch an already available configuration.
- New —To create a new configuration. For more details, see [Creating a Configuration](#) on page 325.
- Edit —To edit a configuration. For more details, see [Editing a Configuration](#) on page 331.
- Delete —To delete a tracking configuration. For more details, see [Deleting a Configuration](#) on page 331.
- Refresh —To refresh the selected configuration. For more details see [Refreshing a Configuration](#) on page 331.

6.1.2.2 Snapshot

[Figure 6-6](#) on page 323 shows the *Snapshot* toolbar of *Tracking* center.

Figure 6-6 Snapshot Toolbar



The *Snapshot* toolbar has following options:

- Take Snapshot—To take snapshots at regular intervals. For more details, see [Taking Snapshots](#) on page 331.
- Delete —To delete the selected snapshot. For more details, see [Deleting Snapshots](#) on page 338.
- Edit — To modify the snapshot date and title. For more details, see [Editing Snapshots](#) on page 338.
- History — To view changes made to attributes of the selected snapshot. For more details, see [Show Snapshot Editing History](#) on page 339.

6.1.2.3 Report

[Figure 6-7](#) on page 324 shows the *Report* toolbar of *Tracking* center.

Figure 6-7 Report Toolbar



The *Report* toolbar has an option *Create Report*, which allows you to create Tracking report in an HTML format. For more details, see [Generating Tracking Reports](#) on page 358.

6.1.2.4 Cumulative

[Figure 6-8](#) on page 324 shows the *Cumulative* toolbar of *Tracking* center.

Figure 6-8 Cumulative Toolbar

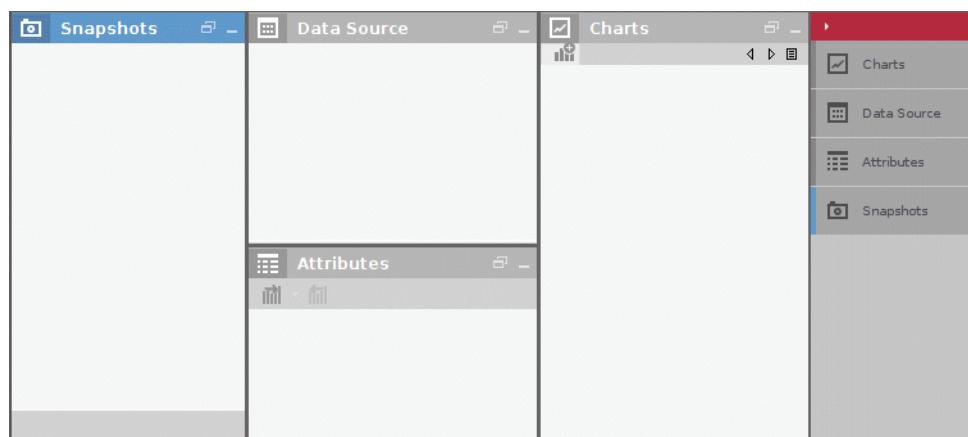


The *Cumulative* toolbar has an option *Sync snapshots*, which recalculates the snapshot data and clears all dirty snapshots. For more details, see [Cumulative Data Tracking](#) on page 340.

6.1.3 View Area

[Figure 6-9](#) on page 324 shows the view area of the *Tracking* center.

Figure 6-9 View Area



The view area has the following panes:

- Charts
- Snapshots
- Data Source
- Attributes

The above screen shows all the panes blank because no snapshot is generated yet. Once, you generate a snapshot the *Snapshots*, *Data Source*, and *Attributes* pane gets populated with relevant data. For more details, see [Taking Snapshots](#) on page 331.

The *Charts* pane get populated, when you generate charts. For more details, see [Generating and Editing Charts](#) on page 342.

6.2 Creating a Configuration

After launching the *Tracking* center, the first task is to create a tracking configuration.

Creating a tracking configuration includes, defining configuration name, owner, specifying the metrics to be sampled, vPlan to be sampled, and defining the sessions that will be read at the time of taking the snapshot.

Note: By default, VMG100 is acquired, which supports 1 tracking configuration. You can add more tracking configurations using the dedicated admin GUI. For more details, see [Incisive vManager Installation and Configuration Guide](#).

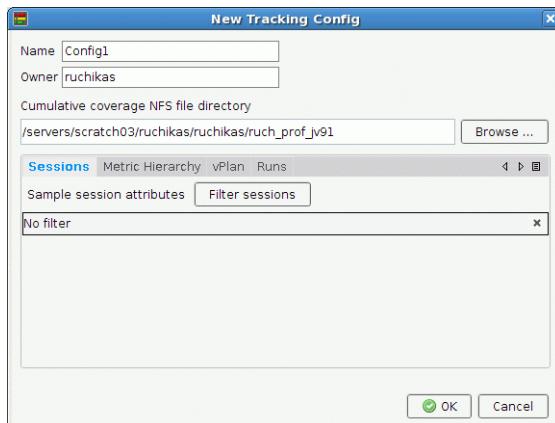
To create a new tracking configuration, perform the following steps:

1. In the *Tracking* center, click the *New* button in the *Tracking Configuration* toolbar.

Note: Alternatively, you can select *New Tracking Config* option from the *Tracking* menu.

The *New Tracking Config* dialog box appears, as shown in [Figure 6-10](#) on page 326.

Figure 6-10 New Tracking Config



2. Specify the name of the tracking configuration in the *Name* field. By default, vManager assigns a name to the tracking configuration. For example, to name the tracking configuration as *Tracking_Sessions*, specify *Tracking_Sessions* in the *Name* field.
3. The *Owner* field, by default, shows the login of the person creating the configuration. If, required, you can assign a different owner in the *Owner* field.
4. Specify the location for storing cumulative coverage data in the *Cumulative coverage NFS file directory* field. By default, this directory will be created under the vManager server profile directory, and the name will be identical to the project tracking configuration name. This directory stores cumulative coverage data.

The *New Tracking Config* dialog box has different pages: *Sessions*, *Metric Hierarchy*, *vPlan*, and *Runs*.

5. In the Sessions page, specify the sessions that must be read at the time of creating the snapshot. By default, all the loaded sessions are considered. In case you want to restrict the sessions at the time of snapshot generation, you can add a filter. You can filter the list of sessions by clicking the *Filter sessions* button. Click the *Filter sessions* button.

The *Edit Filter* dialog box appears, as shown in [Figure 6-11](#) on page 326.

Figure 6-11 Edit Filter



6. Specify the filter criteria in the *Edit Filter* dialog box. For example, to include only the completed sessions for tracking, specify the filter criteria, as shown in [Figure 6-12](#) on page 327. For more details on filtering, see [Filtering Table Data](#) on page 33.

Figure 6-12 Filter Sessions

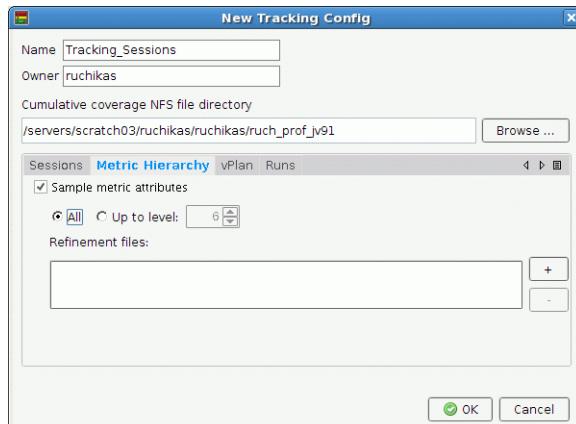


Note: You can also specify filters on more than one field.

7. Click *OK* after specifying the filtering criteria. After you click *OK*, you return to the *New Tracking Config* dialog box.
8. Click *Metric Hierarchy* tab in the *New Tracking Config* dialog box.

[Figure 6-13](#) on page 327 shows the *Metric Hierarchy* tab page.

Figure 6-13 New Tracking Config -- Metric Hierarchy Tab Page

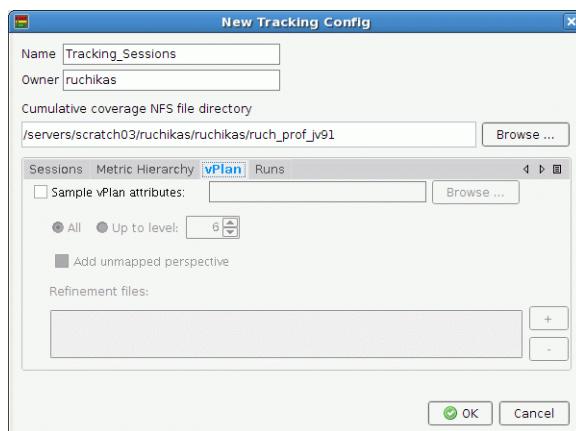


9. To sample metrics attributes at the time of generating the snapshot, select the *Sample metrics attributes* check box. By default, this check box is selected. You can stop sampling of metrics attributes by clearing this check box.
10. Next, specify the levels of hierarchy that must show in the *Metric Hierarchy* page of the tracking configuration. You can select any of the following:
 - All — To show all the levels of hierarchy. By default, this option is selected.
 - Up to level — To specify the number of levels to be shown. By default, 6 is specified in this field. You can increase or decrease the levels, as required.

11. In case you want to apply refinements to the metrics data, you can specify the refinement files in the *Refinement files* field.
12. If required, you can include the vPlan attributes at the time of generating snapshots. For this, click *vPlan* tab in the *New Tracking Config* dialog box.

[Figure 6-14 on page 328](#) shows the *vPlan* tab page.

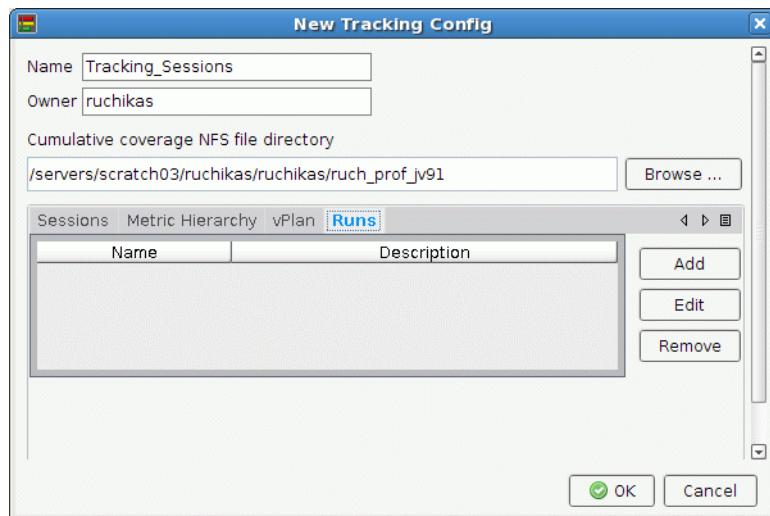
Figure 6-14 New Tracking Config -- vPlan Tab Page



13. To include the vPlan attributes at the time of generating snapshots, select the *Sample vPlan attributes* check box. This will enable other options related to vPlan attributes.
14. Specify the vPlan from which the attributes must be included in the text box next to the *Sample vPlan attributes* option. You can click the *Browse* button to navigate and select the vPlan to be used.
15. Next, specify the levels of hierarchy that must show in the *vPlan* page of the configuration. You can select any of the following:
 - All — To show all the levels of hierarchy. By default, this option is selected.
 - Up to level — To specify the number of levels to be shown. By default, 6 is specified in this field. You can increase or decrease the levels, as required.
16. In case you want to apply refinements to the vPlan data, you can specify the refinement files in the *Refinement files* field.
17. If required, you can add runs data for tracking to the snapshot. For this, click *Runs* tab in the *New Tracking Config* dialog box.

[Figure 6-15 on page 329](#) shows the *Runs* tab page.

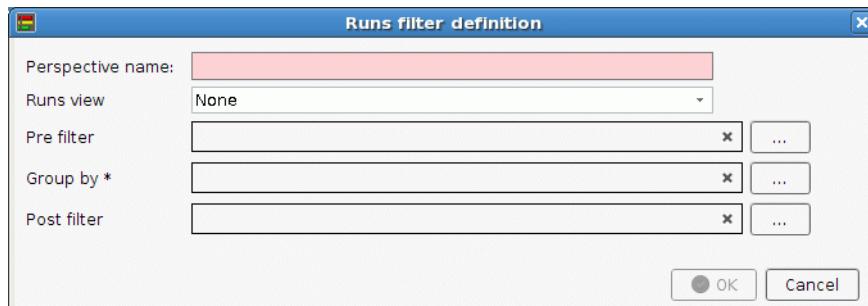
Figure 6-15 New Tracking Config -- Runs Tab Page



In the Runs tab page of the dialog box, you need to define a grouping and filtering criteria that will be executed on the runs that belong to the snapshot's sessions. The runs defined based on the filtering and grouping criteria will be added to the snapshot and then you can track this data in the Tracking center.

18. To add filter and grouping criteria for the runs, click *Add*. The Runs filter definition dialog box is displayed, as shown in [Figure 6-16](#) on page 329.

Figure 6-16 Runs Filter Definition



19. Specify a name for the filter/group definition in the *Perspective name* field.
 20. You can now do any of the following:
 - Select a pre-defined runs view from the *Runs view* drop-down list, or
 - Define a runs filter and group using the *Pre filter*, *Group by*, and *Post filter* fields.
- Note:** Clicking on the *...* button next to *Pre filter* and *Post filter* fields will open

the *Advanced filter* dialog box. You can specify the filters, as required. For more details on advanced filtering, see [Advanced Filtering](#) on page 37. Clicking on the ... button next to the *Group by* field will open the *Attribute Selector* dialog box. You can select the required attributes for grouping the runs (same as in ranking). The *Group by* field is a mandatory field. For more details, see [Ranking Runs](#) on page 184.

For example, [Figure 6-17](#) on page 330 shows a run filter definition named rs1 which groups runs based on the attribute First Failure Description.

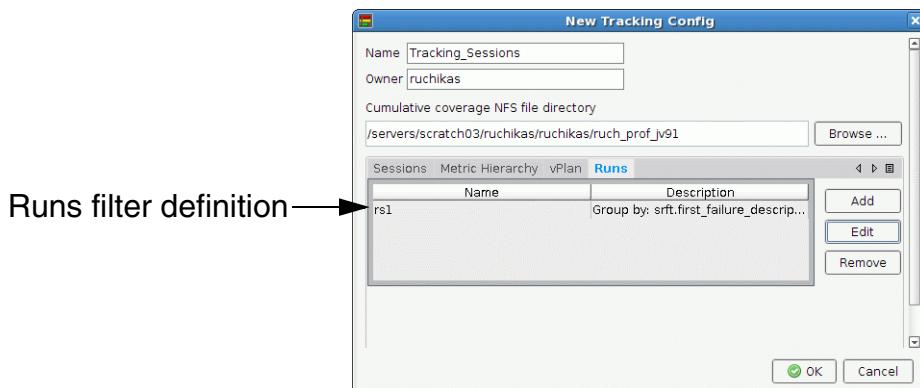
Figure 6-17 Runs Filter Definition



21. Click *OK*.

You will return to the *New Tracking Config* dialog box. The run filter definition rs1 is displayed on the *Runs* tab page, as shown in [Figure 6-18](#) on page 330.

Figure 6-18 New Tracking Config dialog box with Runs Filter Definition



22. Click *OK* to close the *New Tracking Config* dialog box.

This will create a new tracking configuration definition. You can now take snapshots, as required and track the status of your configuration by creating charts.

6.2.1 Editing a Configuration

After you define a tracking configuration, you can edit it, if required. For editing tracking configuration definition, select the tracking configuration you want to edit from the *Configs* drop-down list in the *Tracking Configuration* toolbar and then do any of the following:

- Click the *Edit* button in the *Tracking Configuration* toolbar.
- Select *Edit* from the *Tracking* menu.

This will open the *Edit* dialog box. You can now modify configuration definition, as required, and click *OK*.

6.2.2 Deleting a Configuration

To delete a tracking configuration, select the tracking configuration you want to delete from the *Configs* drop-down list in the *Tracking Configuration* toolbar and then do any of the following:

- Click the *Delete* button in the *Tracking Configuration* toolbar.
- Select *Delete* from the *Tracking* menu.

This will open a confirmation dialog box. Once you confirm deletion, the selected configuration will be deleted.

6.2.3 Refreshing a Configuration

To refresh the snapshots and charts in a tracking configuration, select the tracking configuration you want to refresh from the *Configs* drop-down list in the *Tracking Configuration* toolbar and then do any of the following:

- Click the *Refresh* button in the *Tracking Configuration* toolbar.
- Select *Refresh* from the *Tracking* menu.

This will refresh all of the panes in the view area and will show the refreshed results.

6.3 Taking Snapshots

After creating a configuration, you can take snapshots at regular intervals (every day, every week, or a specific milestone) based on your requirements.

The *Take Snapshot* drop-down button in the *Snapshot* toolbar has following options for taking snapshots:

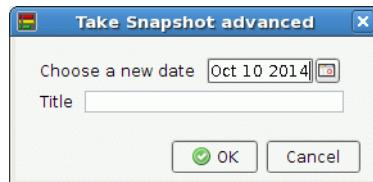
- *Take Snapshot* — To take snapshot as today (without prompting the user to specify snapshot title or date).
- *Take Snapshot advanced* — To prompt the user to specify a date and title at the time of taking snapshot.

Note: Both the above options are also available from the *Tracking* menu.

To create snapshots, click the *Take Snapshot* button in the *Snapshot* toolbar. This will create the snapshot as today, without prompting for any further information, such as date or title.

In case you select *Take Snapshot advanced* option, you are prompted to specify date and title for the snapshot, as shown in [Figure 6-19](#) on page 332.

Figure 6-19 Take Snapshot advanced



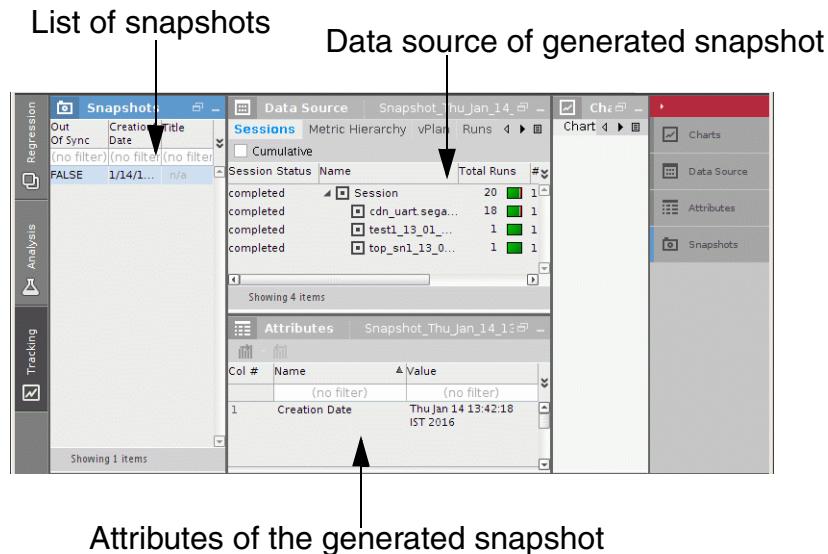
You can specify date and title for the snapshot, as required and click *OK*.

Once you click *OK*, the snapshot is created and *Snapshot*, *Data Source*, and *Attributes* panes in the view area get populated, as shown in [Figure 6-20](#) on page 333.

Note: For taking snapshots in CLI mode, see [snapshot](#) on page 470. To take snapshots from the Regression center, see [Taking Snapshots](#) on page 125.

[Figure 6-20](#) on page 333 shows the generated snapshot.

Figure 6-20 Generated Snapshot



The above screen shows the Tracking center view area. It has the following panes:

- Snapshots Pane
- Data Source Pane
- Attributes Pane

Snapshots Pane

[Figure 6-21 on page 333](#) shows the *Snapshots* pane.

Figure 6-21 Snapshots Pane



The *Snapshots* pane shows the list of snapshots.

You can add more columns or remove columns from the table using the Tools & Options double-arrow button, which when clicked shows the options that allows you to add or remove

columns in the view area, show or hide the search bar, and so on. You can detach the pane using the Toggle floating option. You can hide the pane using the Toggle auto-hide option.

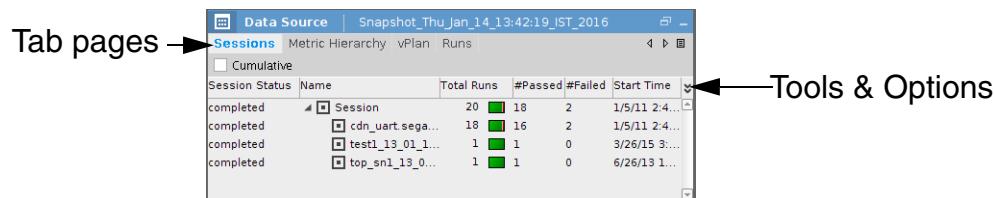
You can right-click on a snapshot and perform the following operations:

- Clear Filters —To release an already applied filter.
- Undo Sort —To unsort the table data.
- Copy Cell—To copy the data in the selected cell and paste it in any editor outside of vManager.
- Copy Row—To copy the data in the selected row and paste it in any editor outside of vManager.
- Delete —To delete the selected snapshot.
- Edit—To change the snapshot date and title.

Data Source Pane

Figure 6-22 on page 334 shows the *Data Source* pane.

Figure 6-22 Data Source Pane



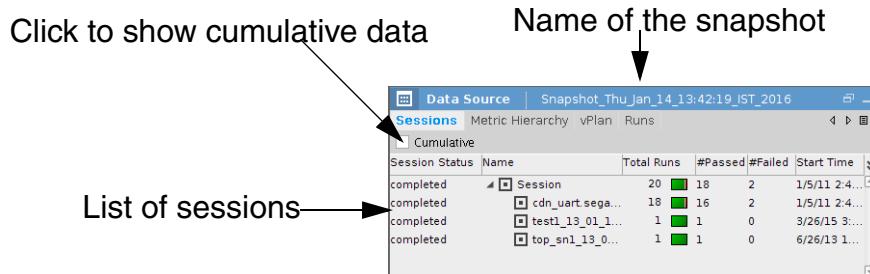
The *Data Source* pane has following tab pages:

- Sessions
- Metric Hierarchy
- vPlan
- Runs

Sessions

Figure 6-23 on page 335 shows the *Sessions* tab page.

Figure 6-23 Sessions Tab Page (Data Source Pane)



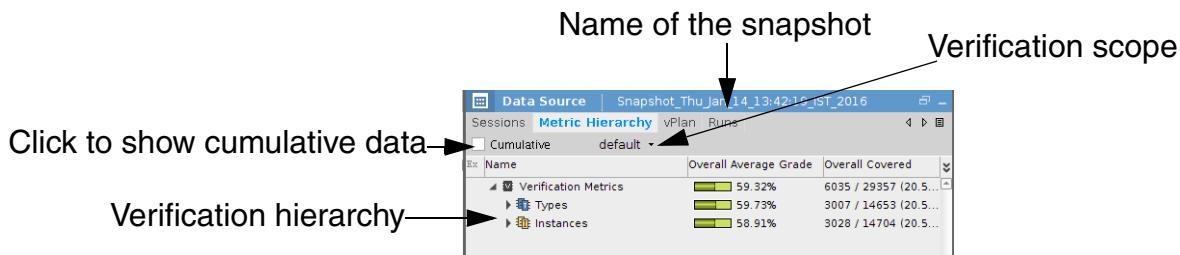
The *Sessions* tab page by default, shows the session status, name, total runs, passed runs, failed runs, and start time of sessions. The sessions listed here are based on the configuration definition that was specified in the *New Tracking Config* dialog box at the time of defining the configuration.

The *Cumulative* check box allows you to show cumulative data.

Metric Hierarchy

[Figure 6-24 on page 335](#) shows the *Metric Hierarchy* tab page.

Figure 6-24 Metric Hierarchy Tab Page (Data Source Pane)



The *Metric Hierarchy* tab page shows the *Types* node and *Instances* node. You can navigate through the various instances and types scored in the design. The levels in the hierarchy are shown based on the configuration definition that was specified in the *New Tracking Config* dialog box at the time of defining a configuration.

The *Cumulative* check box allows you to show cumulative data.

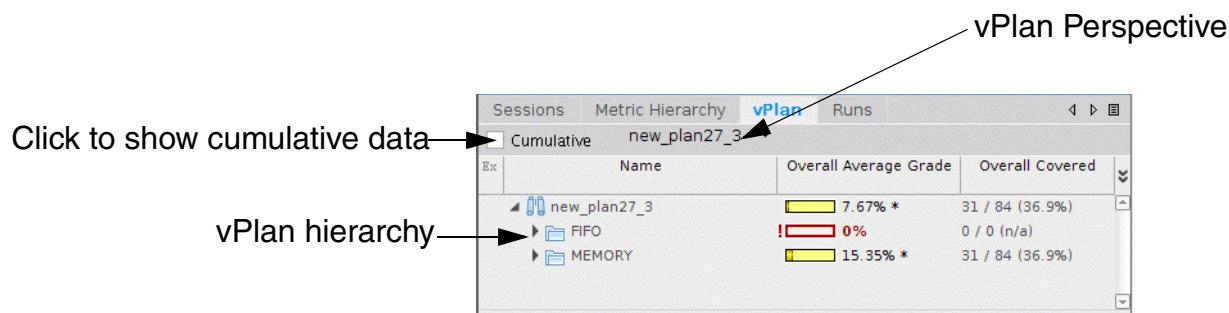
At sampling time vManager samples all the available verification scopes (default). If required, you can select an appropriate verification scope from the *Verification Scopes* drop-down.

Note: The *Metric Hierarchy* tab page will not show anything if at the time of creating a configuration, the *Sample metrics attributes* check box was not selected.

vPlan

[Figure 6-25 on page 336](#) shows the *vPlan* tab page.

Figure 6-25 vPlan Tab Page (Data Source Pane)



The *vPlan* tab page shows the *vPlan* hierarchy. You can navigate through the various *vPlan* elements. The levels in the *vPlan* hierarchy are shown based on the configuration definition that was specified in the *New Config* dialog box at the time of defining a configuration.

The *Cumulative* check box allows you to show cumulative data.

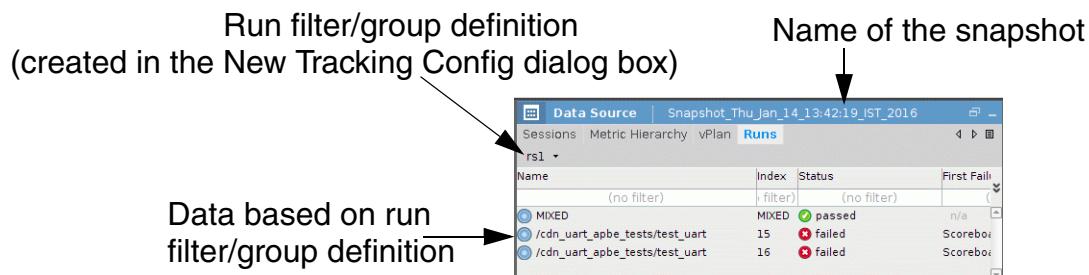
If required, you can also change the perspective from the *Perspectives* drop-down list.

Note: The *vPlan* tab page will not show anything if at the time of creating a configuration, the *Sample vplan attributes* check box was not selected.

Runs

[Figure 6-26 on page 336](#) shows the *Runs* tab page.

Figure 6-26 Runs Tab Page (Data Source Pane)



The *Runs* tab page shows data based on the run filter/group definition that was defined in the *New Tracking Config* dialog box. The list of run/filter definitions are shown as a drop-down list on the table toolbar.

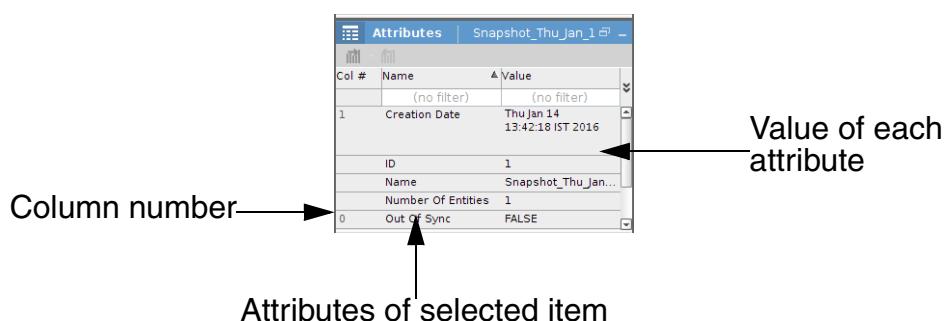
If required, you can change the run/filter definition from the drop-down list.

Note: The *Runs* tab page will not show anything if at the time of creating a configuration, the runs were not defined.

Attributes Pane

[Figure 6-27](#) on page 337 shows the *Attributes* pane.

Figure 6-27 Attributes Pane



The *Attributes* pane displays the characteristics recorded for the item selected in the *Snapshots* pane or the *Data Source* pane.

You can right-click on an attribute and select any of the following:

- Add attribute to chart—To add the selected attribute to the chart. For more details, see [Generating and Editing Charts](#) on page 342.
- Add/Edit attribute to chart advanced—To change the type of graph and color of the graph while adding attribute to the chart. For more details, see [Editing Chart Attributes](#) on page 352.
- Add/Edit attribute for entire perspective—To track all run groups automatically. For more details, see [Generating Charts \(Tracking Run Groups Automatically\)](#) on page 349.
- Remove attribute from chart—To remove the selected attribute from the chart. For more details, see [Generating and Editing Charts](#) on page 342.
- Add Columns—To add the selected attribute in the corresponding pane.
- Remove Columns—To remove the selected attribute from the corresponding pane.

Note: Alternatively, you can double-click on an attribute to add it, and then again double-click it to remove it from the pane.

- Clear Table Filter—To release an already applied filter.
- Undo sort Table—To unsort the table data.
- Copy Cell—To copy the data in the selected cell and paste it in any editor outside of vManager.
- Copy Row—To copy the data in the selected row and paste it in any editor outside of vManager.

6.3.1 Deleting Snapshots

To delete a snapshot, perform the following steps:

1. Select the snapshot in the *Snapshots* pane.
2. Do any of the following:
 - ❑ Click the *Delete* button in the *Snapshot* toolbar.
 - ❑ Select *Delete* (under Take Snapshot) from the *Tracking* menu.
 - ❑ Right-click and select *Delete* from the pop-up menu.

This will open a confirmation dialog box. Once you confirm deletion, the selected snapshot will be deleted.

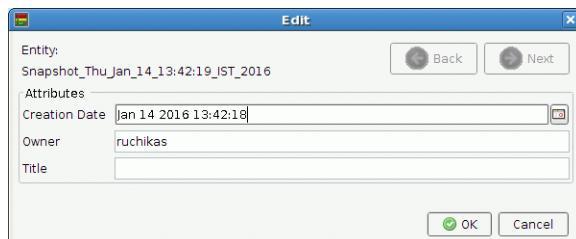
6.3.2 Editing Snapshots

To modify the date, owner, and title of the snapshot, perform the following steps:

1. Select the snapshot whose attributes you want to change.
2. Do any of the following:
 - ❑ Click the *Edit* button in the *Snapshot* toolbar.
 - ❑ Select *Edit* (under Take Snapshot) from the *Tracking* menu.
 - ❑ Right-click and select *Edit* from the pop-up menu.

The *Edit* dialog box appears, as shown in Figure 6-28 on page 339.

Figure 6-28 Change Snapshot Date



3. Change the date, as required in the *Creation Date* field.
4. If required, change the owner in the *Owner* field.
5. If required, change the title of the snapshot in the *Title* field.
6. Click *OK*.

Note: The attribute *Title* of the snapshot can be modified to give it a meaningful name. However, the attribute *Name* cannot be edited. The name of the snapshot is a unique identifier for the snapshot, which is generated automatically by the tool and has timestamp suffixed to it.

6.3.2.1 Show Snapshot Editing History

The *History* option on the Snapshot toolbar allows you to view the attribute change history of the selected snapshot. Figure 6-29 on page 339 shows the *Attribute editing history* dialog box.

Figure 6-29 Attribute Editing History

Name	Owner	Original Value	New Value	Date
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Creation Date	ruchikas	2016-01-14 13:42:18 IST	Wed Jan 13 13:42:18 IST 1/14/16 2:02 PM	Wed Jan 13 13:42:18 IST 1/14/16 2:02 PM

Annotations:

- Name of the attribute whose value was changed: Points to the 'Name' column header.
- Owner: Points to the 'Owner' column header.
- Original value of the attribute: Points to the 'Original Value' column.
- New value of the attribute: Points to the 'New Value' column.
- Date and time when attribute was edited: Points to the 'Date' column.

This dialog box maintains the history of all the changes made to different attributes of the selected snapshot.

6.4 Cumulative Data Tracking

By tracking cumulative data, you can easily follow the project's progress, and easily find the problematic points in the project. For example, you can identify the areas in the metrics model that are not covered in any of the snapshot.

Note: Cumulative data calculation is heavy and time-consuming. As a result, for each snapshot, vManager calculates the cumulative data even if the user has not clicked on the *Cumulative* check box. This enables vManager to provide cumulative data quickly at any point in time, even if the original coverage files have been deleted.

This section covers the following topics:

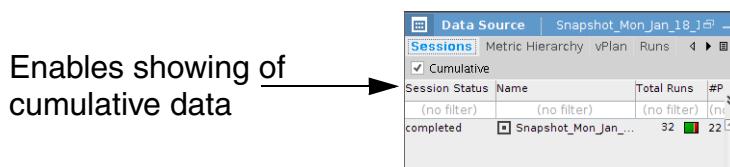
- [Viewing Cumulative Data](#)
- [Adding Cumulative Attributes to Chart](#)
- [Recalculating Cumulative Data](#)

6.4.1 Viewing Cumulative Data

The *Sessions*, *Metric Hierarchy*, and *vPlan* tab pages of the *Data Source* pane has a cumulative check box, that allows you to show the cumulative data.

[Figure 6-30](#) on page 340 shows the *Sessions* tab of *Data Source* pane with the cumulative data.

Figure 6-30 Data Source Pane



The session's cumulative data is calculated by merging the current snapshot's sessions with all of the previous snapshot's sessions.

In the *Metric Hierarchy* tab, the cumulative tree is calculated by executing merge on the current snapshot metrics tree with the previous snapshot's cumulative metrics tree.

In the vPlan tab, the cumulative tree is calculated based on the cumulative metrics tree. Therefore, the vPlan which is defined by the user in the project tracking configuration gets loaded on the cumulative metrics tree.

6.4.2 Adding Cumulative Attributes to Chart

You can add any of the attributes of the cumulative data to the chart.

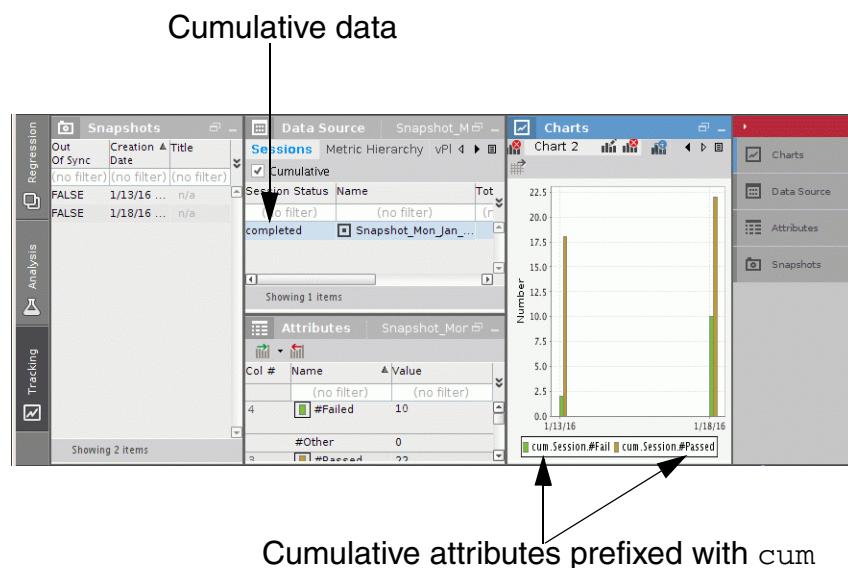
To add cumulative attribute to chart, perform the following steps:

1. In the *Attributes* pane, select the attribute to be added to the chart.
2. Right-click and select any of the following:
 - Add attribute to chart
 - Add/Edit attribute to chart advanced

Note: For more details on adding attributes to charts, see [Generating and Editing Charts](#) on page 342.

The cumulative attributes added to the chart are prefixed with text `cum`, as shown in [Figure 6-31](#) on page 341.

Figure 6-31 Cumulative Attributes



6.4.3 Recalculating Cumulative Data

The order of the snapshot depends on the snapshot's creation date attribute and is considered while calculating the cumulative data. As a result, if there is a change in the creation date of a snapshot or if a snapshot is deleted, the cumulative data of the changed snapshot and the snapshots taken after this snapshot might expire. This requires recalculation of the cumulative data.

The value of `Out Of Sync` attribute of snapshots for which the cumulative data has to be recalculated (due to a "change date" operation or a "delete snapshot" action) changes to `TRUE`, as shown in [Figure 6-32](#) on page 342.

Figure 6-32 Data Source Pane

Out Of Sync	Creation Date	Title
(no filter)	(no filter)	(no filter)
FALSE	10/28/14 4:11 PM	n/a
TRUE	10/29/14 11:43 AM	s3
TRUE	10/30/14 11:42 AM	s2

This indicates that the snapshot requires recalculation of cumulative data.

To recalculate the snapshot data, do any of the following:

- Sync snapshots in the Cumulative toolbar, or
- Sync snapshots from the Tracking menu

When you click the *Sync snapshots* button, the popup message appears indicating that the recalculation might take a few minutes. It also has a check box that you can click for not showing this popup message again.

The cumulative data is recalculated and all dirty snapshots are cleared (that is -- the value of the *Out Of Sync* field is changed to `FALSE`).

6.5 Generating and Editing Charts

After taking snapshots, you can generate charts to track the configuration status. In the charts, you can add attributes related to sessions, tests, metrics, vPlan, runs, and also the user-defined attributes. The charts are saved in the database and therefore, when a client opens the configuration, all the charts associated with the configuration are also available for analysis. In this section, following topics are covered:

- [Generating Charts \(Basic\)](#) on page 343

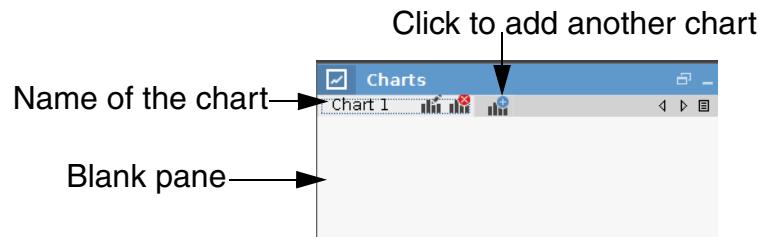
- [Generating Charts \(Advanced\) on page 346](#)
- [Generating Charts \(Tracking Run Groups Automatically\) on page 349](#)
- [Editing Chart Attributes on page 352](#)
- [Removing Attributes from Charts on page 354](#)
- [Removing Charts on page 355](#)
- [Renaming Charts on page 356](#)
- [Exporting Charts to CSV File on page 357](#)

6.5.1 Generating Charts (Basic)

To generate charts, perform the following steps:

1. When you take a snapshot, by default a blank chart *Chart 1* is created in the Charts pane, as shown in [Figure 6-33](#) on page 343.

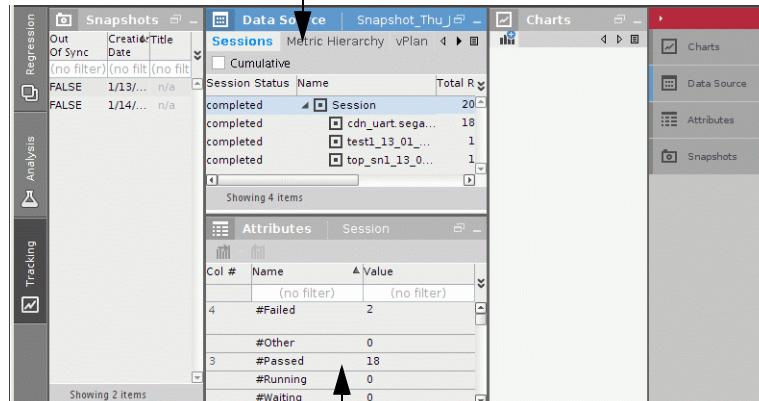
Figure 6-33 Create Charts



2. You now need to add the attributes on the chart. For this, you can navigate through the sessions, metrics hierarchy, vPlan hierarchy, and runs in the *Data Source* pane and then identify the corresponding attributes in the *Attributes* pane, as shown in [Figure 6-34](#) on page 344.

Figure 6-34 Create Charts (Navigate and identify attributes)

Navigate through sessions, metrics, vPlan and runs

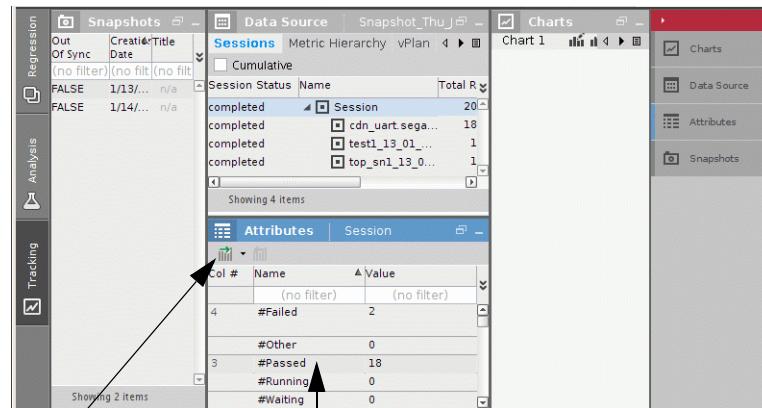


The screenshot shows the vManager interface with the 'Data Source' tab selected. In the 'Sessions' section, there are four items: completed, completed, completed, and completed. Under each 'completed' item, there is a 'Session' node with further details like 'cdn_uart_sega...' and 'test1_13_01...'. The 'Attributes' pane below shows session metrics: #Failed (2), #Other (0), #Passed (18), #Running (0), and #Waiting (0). Arrows point from the text 'Navigate through sessions, metrics, vPlan and runs' to the 'Sessions' and 'Attributes' panes.

Identify the attributes to show on the chart

3. For example, to create a chart that shows *#Passed* sessions and *#Failed* sessions, select *#Passed* in the *Attributes* pane and click the *Add attribute to chart* icon, as shown in [Figure 6-35](#) on page 344.

Figure 6-35 Create Charts (Add attributes)



The screenshot shows the same interface as Figure 6-34, but with a focus on the 'Attributes' pane. An arrow points to the 'Add attribute to chart' icon in the top right corner of the pane. Another arrow points to the '#Passed' attribute itself. The pane lists attributes: Col #, Name, and Value. The '#Passed' row has a value of 18.

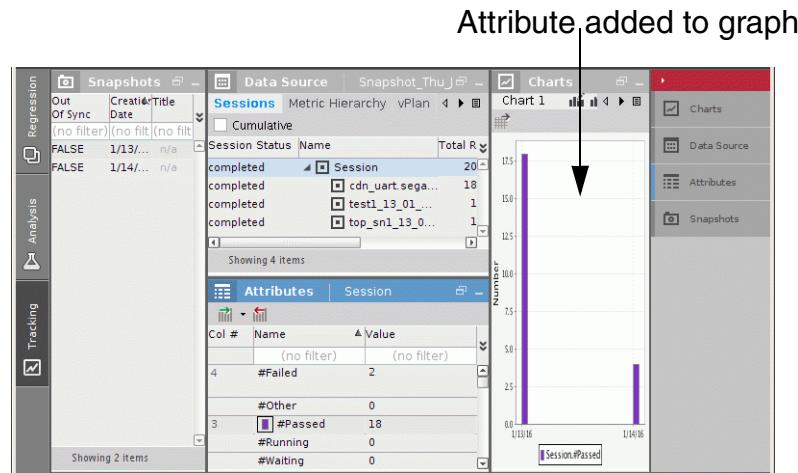
Click the *Add attribute to chart* icon

Select the attribute

Note: Alternatively, you can select the attribute, right-click, and then select *Add attribute to chart* option.

After you click the *Add attribute to chart* icon, the attribute shows as a graph in the *Charts* pane, as shown in [Figure 6-36](#) on page 345.

Figure 6-36 Create Charts (Attribute Added)

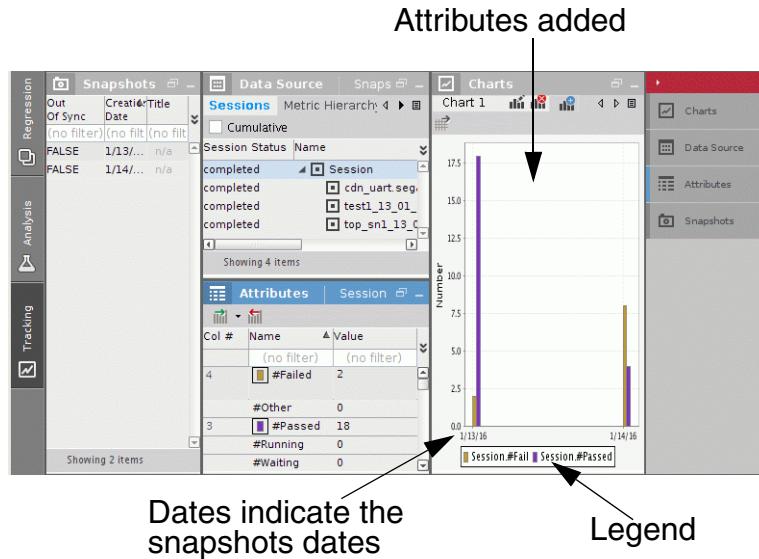


Note: By default, a bar graph is created and also the colors are automatically assigned to the chart by vManager. However, you have an option of changing the type of graph and colors according to your requirements. For this, instead of selecting the *Add attribute to chart* option in the above step, select the *Add/Edit attribute to chart advanced* option. For more details, see [Generating Charts \(Advanced\)](#) on page 346.

4. Similarly, select *#Failed* in the Attributes tab page and click *Add attribute to chart* icon.

Now, the *#Failed* attribute also shows on the graph, as shown in [Figure 6-37](#) on page 346.

Figure 6-37 Create Charts (Attribute Added)



The chart now shows both the attributes. The dates shown below each bar, indicates the dates on which snapshots are created. In this case, there are two snapshots. The legend below the dates show the colors along with the attribute that it shows.

Similarly, you can create more charts and track the progress of your configuration over time.

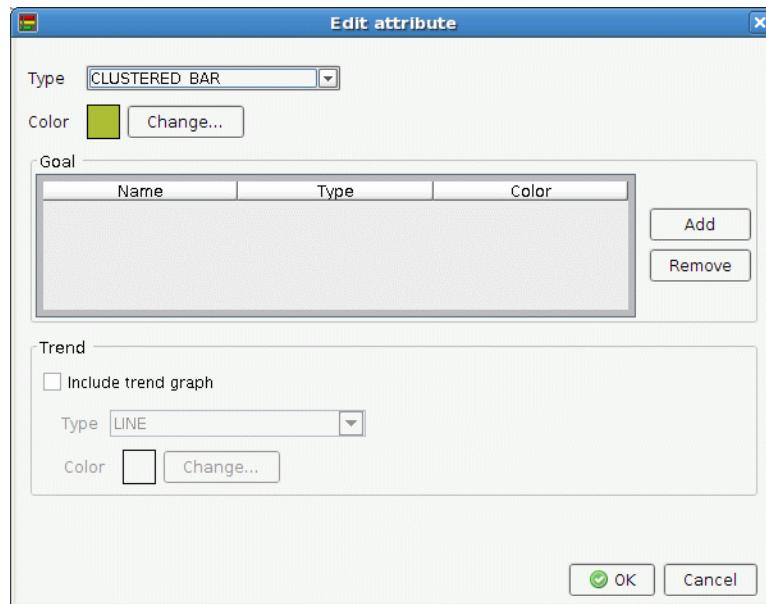
6.5.2 Generating Charts (Advanced)

While adding attributes to the chart, you have certain advanced capabilities, such as changing the chart type, changing the color of the graph, adding goal graph and trend graph for advanced analysis.

For using advanced features at the time of adding attributes to chart, perform the following steps:

1. Navigate through the sessions, metrics hierarchy, vPlan hierarchy, and runs in the *Data Source* pane. For example, select the *Metrics Hierarchy* page in the *Data Source* pane.
2. In the *Attributes* pane, identify the attribute to be added to the chart. For example, select *Overall Covered Grade*.
3. Right-click the attribute, and then select *Add/Edit attribute to chart advanced* option. The *Edit attribute* dialog box is displayed, as shown in [Figure 6-38](#) on page 347.

Figure 6-38 Edit Attribute



4. In the *Type* drop-down list, change is the type of graph to line or step, as required. By default, bar graph is created. For example, select **LINE** graph.
5. In case you want to change the color of the graph, click the *Color* button and select the required color in the *Select a color* dialog box and then click **OK**.
6. You can also add a goal graph for the selected attribute. This helps you to track the project performance against a defined goal. You can define goal in a CSV file, and then load it in the GUI. The CSV file is a simple *Date–Value* file that you can create manually. The following is the format of the CSV file for defining goal:

```
Date, <GoalName>
<Date1>,<Value1>
<Date2>,<Value2>
...
...
...
```

[Figure 6-39](#) on page 348 shows a sample CSV file for defining goal.

Figure 6-39 CSV File

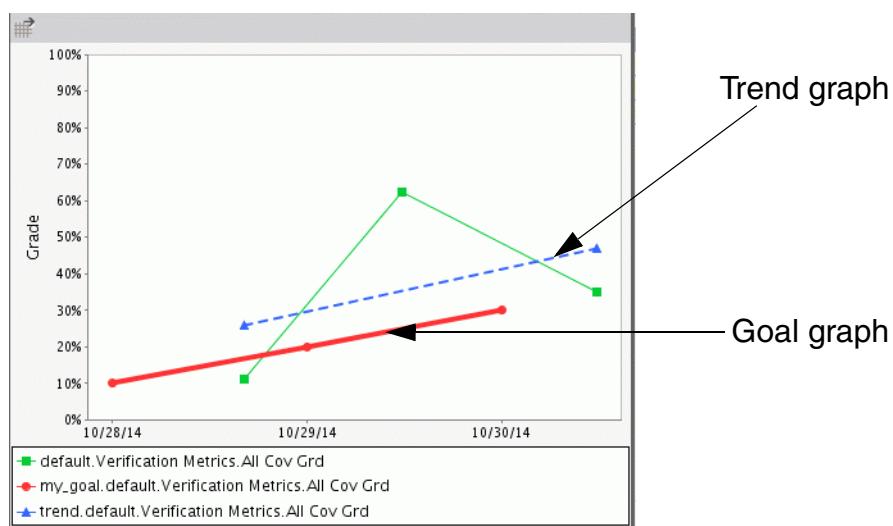
A	B
Date	my_goal
10/28/2014	10
10/29/2014	20
10/30/2014	30

After creating the CSV file, you can then add it to GUI for the required attribute.

7. Click the *Add* button in the *Edit attribute* dialog box.
 8. The *Choose Goal* dialog box appears. Select the required CSV file, and click *OK*.
 9. You then return to the *Edit attribute* dialog box. The loaded CSV file shows in the *Goal* section of the dialog box.
- Note:** To remove an added CSV file, click the *Remove* button.
10. You can now specify the creation of trend graph for the attribute. To include a trend graph, select the *Include trend graph* check box.
 11. You can then specify the type of graph for the trend graph from the *Type* drop-down list. It can be `LINE`, `THICK_LINE`, or a `DASHED_LINE`. For example, select `DASHED_LINE`.
 12. If required, change the color of the trend graph.
 13. Click *OK*.

[Figure 6-40](#) on page 348 shows the *Charts* pane with the goal and trend graph.

Figure 6-40 Chart Created (Goal and Trend Graph)



The above chart shows the goal and trend graph added to the chart.

It helps you to monitor the general direction and the actual progress of the project.

Note: In case you want to change the goal values, modify the goal values in the CSV file, then remove the already loaded goal file in the *Edit attributes* dialog box, and then add the new goal file. For more details, see [Editing Chart Attributes](#) on page 352.

6.5.3 Generating Charts (Tracking Run Groups Automatically)

While adding attributes to the chart, in the case of runs attributes, you have an additional ability to track all run group items as well as the new unknown items automatically.

This is done using the *Add/Edit attribute for entire perspective* option.

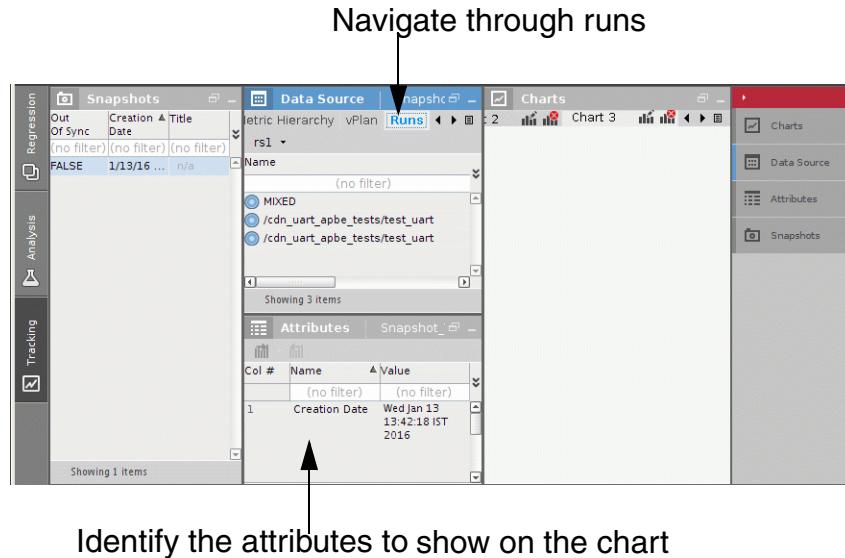
Consider the following steps to enable automatic tracking of runs attributes:

1. Create a tracking configuration.

Note: While creating the tracking configuration, ensure that the runs data is added and is grouped, as required. For more details, see [tracking runs data](#).

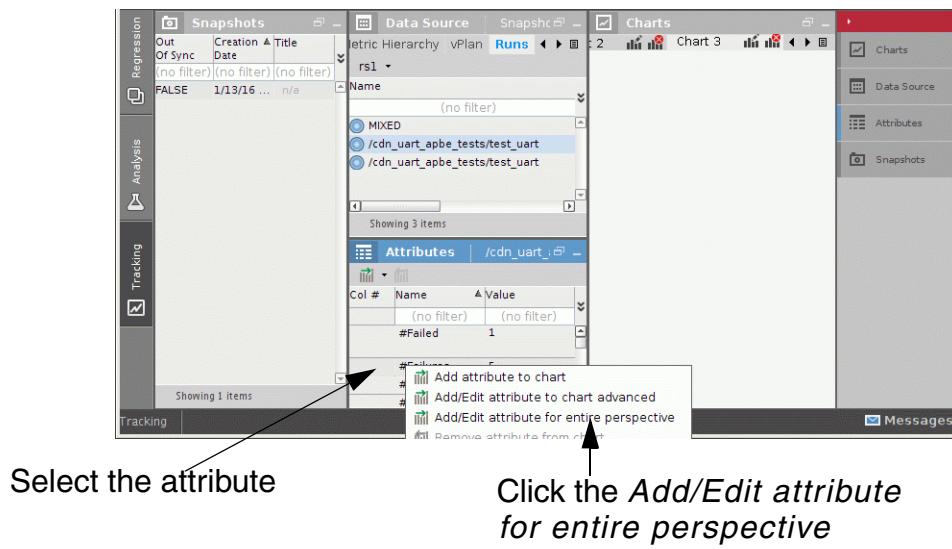
2. Generate a snapshot. For more details, see [Taking Snapshots](#) on page 331.
3. You now need to add the runs attributes to the chart. For this, click on the *Runs* page in the *Data Source* pane and then identify the corresponding attributes in the *Attributes* pane, as shown in [Figure 6-41](#) on page 350.

Figure 6-41 Create Charts (Navigate and identify attributes)



4. For example, select *#Failures* in the *Attributes* pane and click the *Add/Edit attribute for the entire perspective* from the drop-down, as shown in [Figure 6-42](#) on page 350.

Figure 6-42 Create Charts (Add attributes)



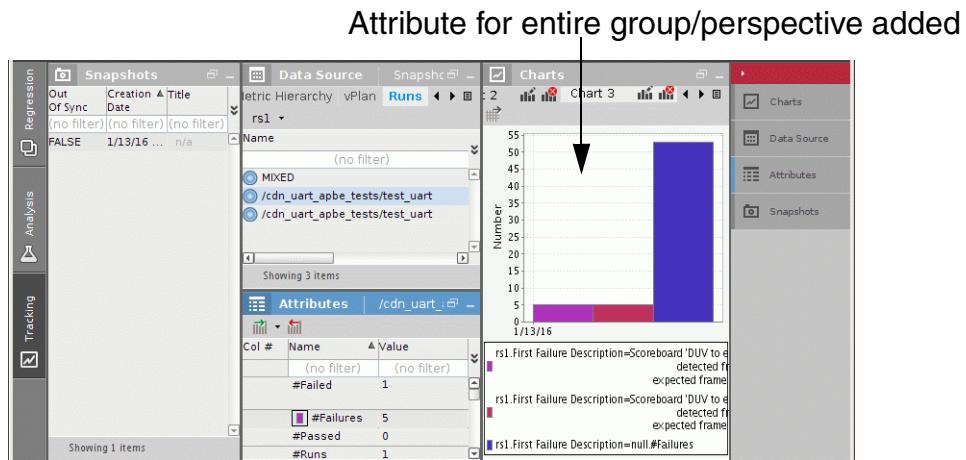
Note: Alternatively, you can do any of the following:

- In the Attributes pane, select the attribute, right-click, and then select *Add/Edit attribute for entire perspective* option.

- In the Runs page of Data Source pane, right-click the attribute column, and then select *Add/Edit attribute for entire perspective* option.

After you click the *Add/Edit attribute for entire perspective*, the attribute for the entire group/perspective gets added to the chart, as shown in [Figure 6-43](#) on page 351.

Figure 6-43 Create Charts (Attribute Added)

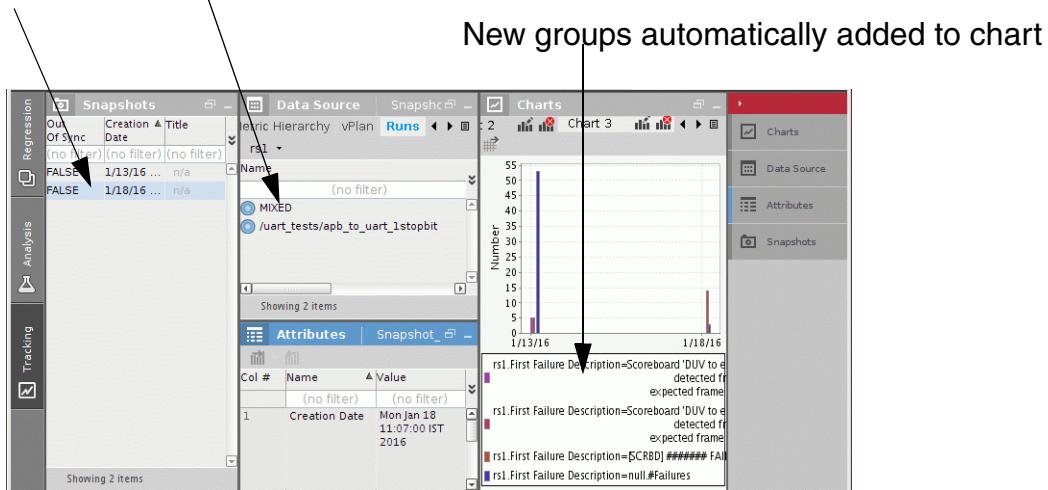


Notice that, in this case, the four groups/items are shown in the *Runs* pane. When you selected the *Add/Edit attribute for entire perspective* option, the selected attribute (#Failures) was added automatically to the chart from all the four groups. You did not had to add this attribute separately for each of the listed items in the *Runs* pane.

In addition, after you have enabled automatic charting using the *Add/Edit attribute for the entire perspective* option, then whenever you generate a new snapshot that causes generation of new run groups, then those new run groups are also added automatically to the chart, as shown in [Figure 6-44](#) on page 352.

Figure 6-44 Create Charts (New Items also Added)

New snapshot with new run groups



New groups automatically added to chart

Note: You can edit and delete the attributes on the chart, as required. The procedure for editing and removing attributes in the case of automatically added attributes is same as the ones added individually. For more details, see [Editing Chart Attributes](#) on page 352 and [Removing Attributes from Charts](#) on page 354. If you delete an attribute that was added automatically, then that attribute line will also not appear in future snapshots.

6.5.4 Editing Chart Attributes

vManager allows you to change the type of graph, color of the graph, or name of attributes added to the chart.

To edit the chart attributes, perform the following steps:

1. Select the attribute on the chart.
2. Right-click and select *Edit attributes* from the pop-up menu, as shown in [Figure 6-45](#) on page 353.

Figure 6-45 Edit Attributes

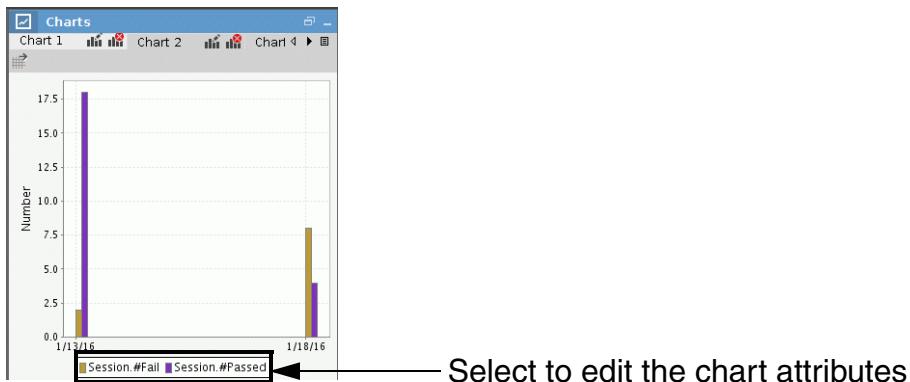
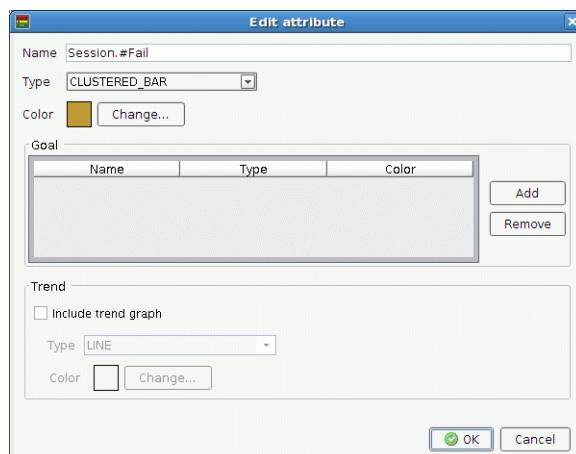


Figure 6-46 on page 353 shows the *Edit attribute* dialog box.

Figure 6-46 Edit Attributes



3. If required, change the value in the *Name* field. Sometimes, the value shown in the name field might be very long (for example, in the case of failure description) and therefore, might require modification.
4. In the *Type* drop-down list, change is the type of graph to line or step, as required. By default, bar graph is created.
5. In case you want to change the color of the graph, click the *Color* button and select the required color in the *Select a color* dialog box and then click *OK*.
6. In the *Goal* section, you can do any of the following:
 - a. Add a goal graph by clicking on *Add*.

- b. Remove an already added goal graph by selecting the goal in the list box and clicking *Remove*.

Note: In case you want to change the goal values, modify the goal values in the CSV file, then remove the already loaded goal file, and then add the new goal file.

- c. Edit the color and type of goal graph, as required.

7. In the *Trend* section, you can do any of the following:

- a. Modify the name of the trend graph in the *Name* field.
- b. Enable or disable showing of trend graph on the chart by selecting or clearing the *Include trend graph* check box.
- c. In the Type drop-down list, if required, change the type of trend graph.
- d. If required, change the color of the trend graph.

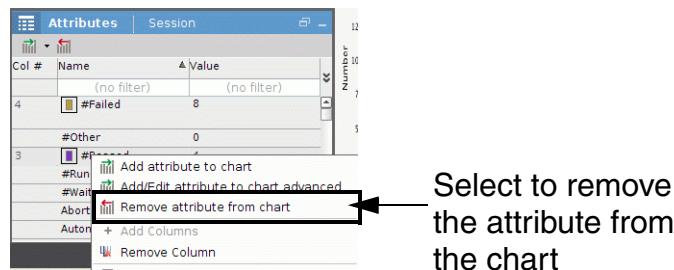
For more details on goal and trend graphs, see [Generating Charts \(Advanced\)](#) on page 346.

6.5.5 Removing Attributes from Charts

In case you want to remove an attribute from the chart, do any of the following:

- Select the attribute in the *Attributes* tab page, right-click, and select *Remove attribute from chart* from the pop-up menu, as shown in [Figure 6-47](#) on page 354.

Figure 6-47 Remove Attribute from Chart



- Select the attribute in the *Attributes* tab page and select the *Remove attribute from chart* icon, as shown in [Figure 6-48](#) on page 355.

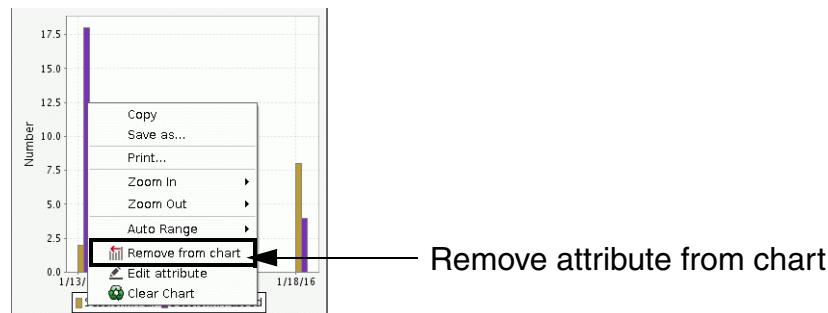
Figure 6-48 Remove Attribute from Chart

Select to remove the attribute from the chart

Col #	Name	Value
	(no filter)	(no filter)
4	#Failed	8
	#Other	0
3	#Passed	4
	#Running	0
	#Waiting	0

- Select the attribute on the chart, right-click, and select the *Remove from chart* from the pop-up menu, as shown in [Figure 6-49](#) on page 355.

Figure 6-49 Remove from Chart



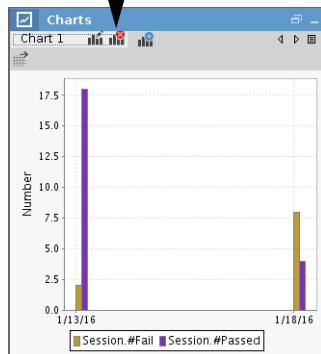
Note: To clear all of the attributes from the chart together, right-click anywhere on the chart and select *Clear Chart* from the pop-up menu.

6.5.6 Removing Charts

To remove a chart, click on the icon, as shown in [Figure 6-50](#) on page 356.

Figure 6-50 Removing Charts

Click to delete the chart



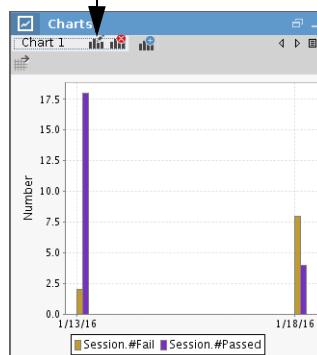
After you click the shown icon, the chart is removed from the *Tracking* pane.

6.5.7 Renaming Charts

By default, the chart is named as *Chart 1*. You can rename the chart by clicking the pencil icon, as shown in [Figure 6-51](#) on page 356.

Figure 6-51 Renaming Charts

Click to rename a chart



The *Rename chart* dialog box appears. Specify the new name of the chart and click *OK*.

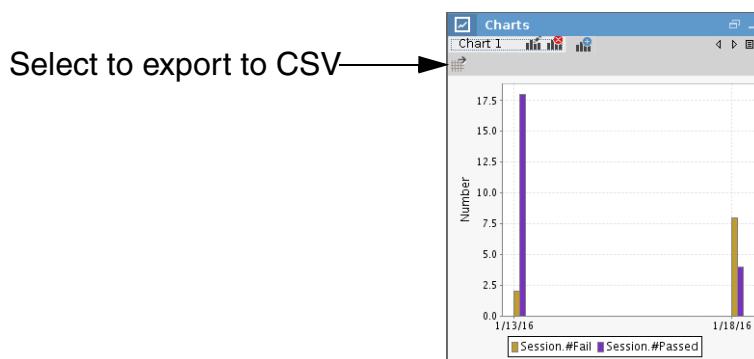
The chart will be renamed.

6.5.8 Exporting Charts to CSV File

To export a chart to a CSV file, perform the following steps:

1. Select the *Export to csv file* icon, as shown in [Figure 6-52](#) on page 357.

Figure 6-52 Export to CSV File



The *Export to csv file* dialog box appears, as shown in [Figure 6-53](#) on page 357.

Figure 6-53 Export to CSV File



2. In the *File path* field, you can specify the name and location of the file where the exported file must be saved. By default, it shows the location from which vManager was invoked and the file name is shown as <timestamp>.csv. You can change it, as required and click *OK*.

After you click *OK*, the metrics tree is exported to the CSV file at the specified location.

[Figure 6-54](#) on page 357 shows the exported CSV file.

Figure 6-54 Exported CSV File

A	B	C
1	Session.#Fail	Session.#Passed
2	10/10/2014 15:09	10
3	10/12/2014 15:54	2
4		15

Each snapshot becomes a row in the CSV file.

The first column of the CSV file is the snapshot `creation_time` attribute. The other columns included in the CSV file are based on the attributes added on the chart.

Note: You can also export data from the command-line interface of vManager. For more details, see [csv_export](#).

6.6 Generating Tracking Reports

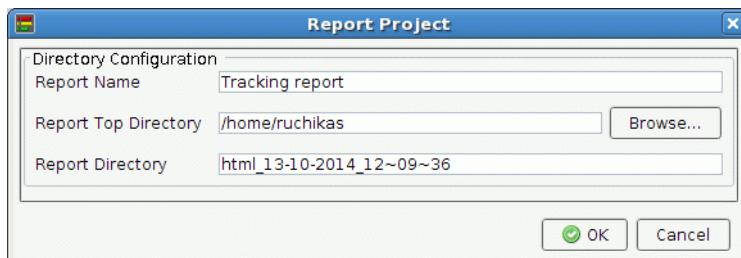
A tracking report includes charts created for the selected configuration. Using vManager, you can generate a Tracking report in an HTML format and then publish it on the internet. Generating an HTML report is useful for users who want to make the results available for viewing to people who might not have vManager installed with them.

To generate a Tracking report:

1. Click the *Create Report* button in the *Report* toolbar.

The *Report Project* dialog box is displayed, as shown in [Figure 6-55](#) on page 358.

Figure 6-55 Report Project



2. Specify the name of the report in the *Report Name* text box. By default, the name is specified as *Tracking report*. This name appears on the HTML report.
3. In the *Report Top Directory* field, specify the location where the report must be created. By default, it shows the location from which vManager was invoked.
4. Specify the name of the report directory in the *Report Directory* text box. By default, it shows the name as `html_<timestamp>`. For example, change the name as `track_report`.
5. Click *OK* to generate the report.

The *Report Done* dialog box is displayed, as shown in [Figure 6-56](#) on page 359.

Figure 6-56 Report Done



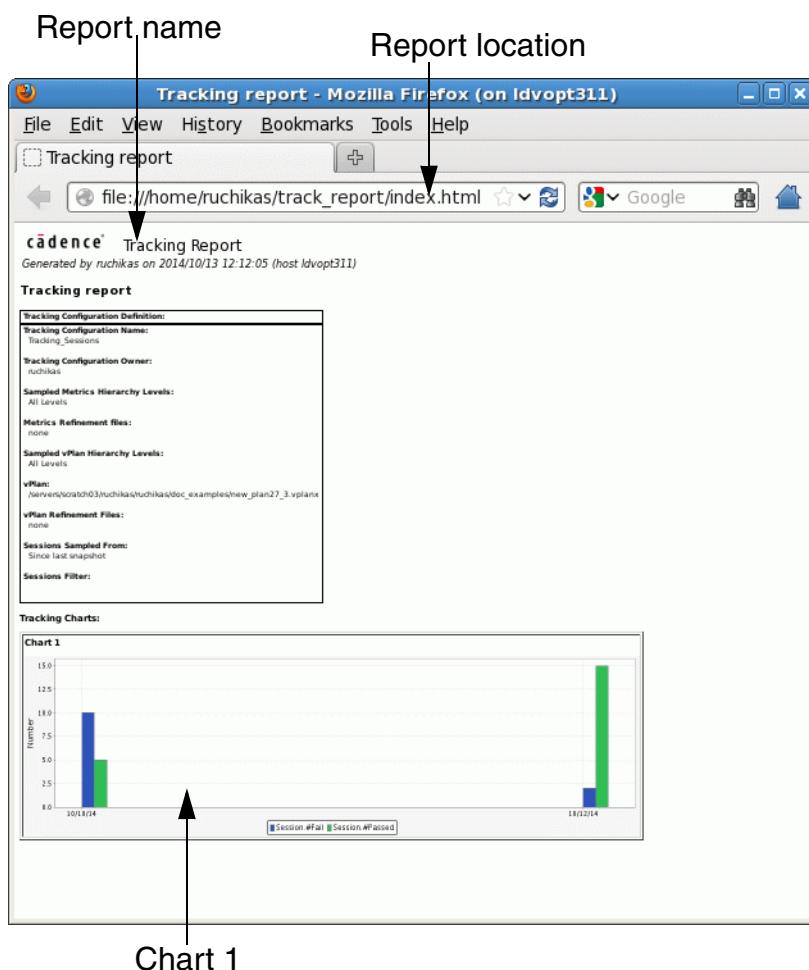
This dialog box shows the location where the report is generated.

6. Click OK.

You can navigate through the location shown above, and then open the `index.html` file in a Web browser.

[Figure 6-57 on page 359](#) shows the generated HTML report.

Figure 6-57 Tracking Report



The tracking report shows information such as, who generated the report and when the report was generated. It also shows the configuration details, such as configuration name, owner, vPlan, sessions from which data is sampled and so on. It shows all the charts created for the configuration. You can scroll down and view more charts.

Note: You can also generate reports in command-line mode of vManager using the [report_tracking](#) on page 509.

6.7 Converting Regular Chart to Triage or Stacked Chart

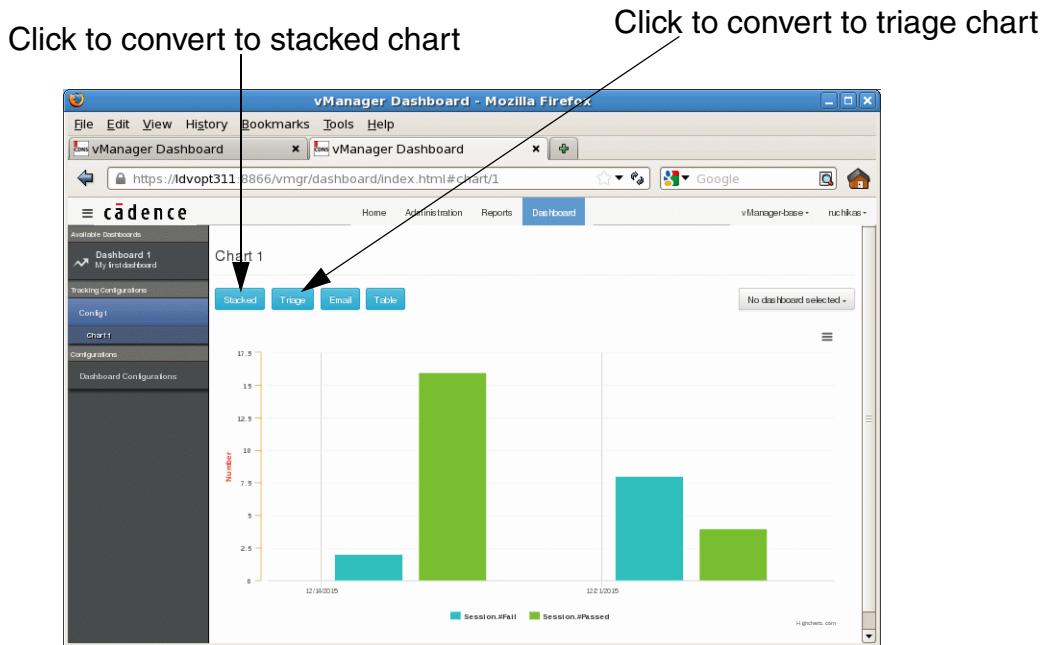
After generating a chart, you can convert it to a triage chart or a stacked chart using the Web dashboard.

Note: Not all charts can be converted to triage or stacked. Only the ones with same numeric type of attributes (for example, all attributes are numbers or percentages) can be exported to a triage or stacked chart.

The Web dashboard can be enabled using the Administration activity center. You can invoke vManager with the Administration activity center using the `-admin` option. For more details, see [-admin](#) on page 395.

After you login to the Web dashboard, the chart is available for viewing and conversion, as shown in [Figure 6-58](#) on page 361.

Figure 6-58 Web Dashboard



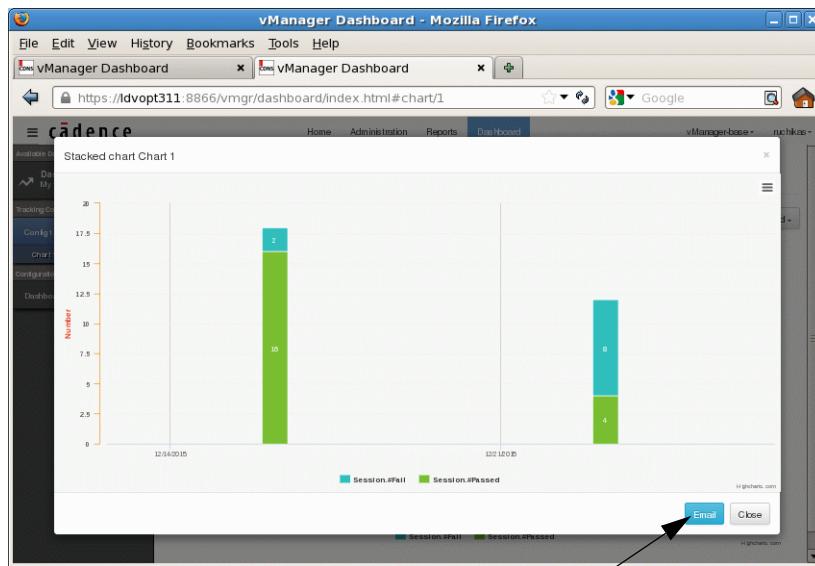
The following options are available from the Web dashboard for the charts:

- Stacked—To convert the shown regular chart to a stacked chart.
- Triage—To convert the shown regular chart to a triage chart.
- Email—To email the link of the shown regular chart to anyone, as required.
- Table—To view the snapshot table using which the chart is generated.

Convert to a Stacked Chart

To convert the shown regular chart to a stacked chart, click the *Stacked* button on the Web dashboard. [Figure 6-59](#) on page 362 shows the converted stacked chart.

Figure 6-59 Stacked Chart



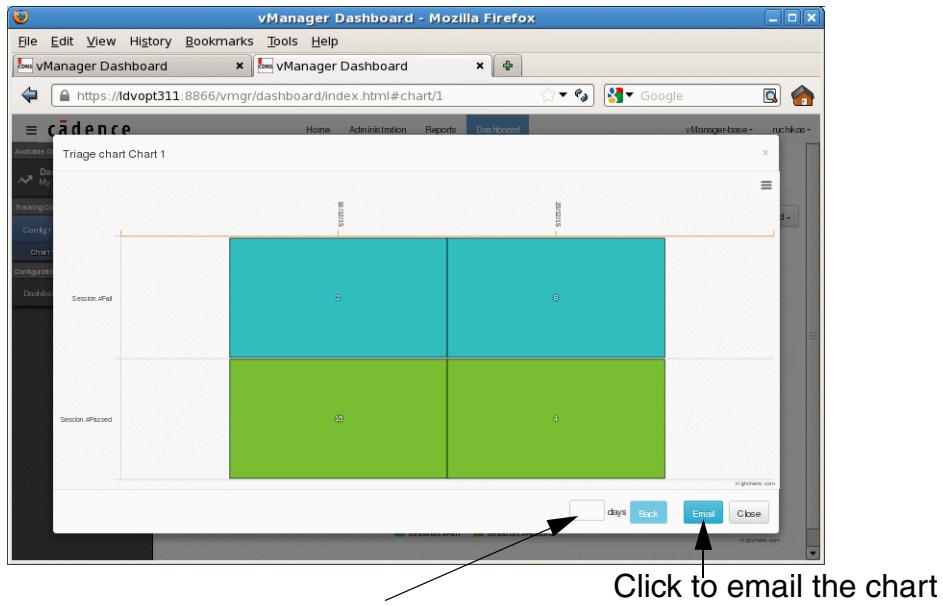
Click to email the stacked chart

- Click *Email* to send the link of this chart as an email, as required.
- Click *Close* to close the Stacked chart window and to return to the Web dashboard.

Convert to a Triage Chart

To convert the shown regular chart to a triage chart, click the *Triage* button on the Web dashboard. [Figure 6-60](#) on page 363 shows the converted triage chart.

Figure 6-60 Triage Chart



Specify days for which
the chart should be shown

Click to email the chart

- In the *days* text box, specify the number of days for which the chart must be shown and click the *Back* button. By default, chart is shown for all days.
- Click *Email* to send the link of this chart as an email, as required.
- Click *Close* to close the Triage chart window and to return to the Web dashboard.

Note: You can use the CLI command chart_export to convert and export the chart in the .png format.

Incisive vManager User Guide

List of Attributes

This appendix describes:

- [Attributes associated with Session](#)
- [Attributes associated with Test or Tests Group](#)
- [Attributes associated with Run](#)
- [Attributes associated with Failure](#)
- [Attributes associated with Formal Properties](#)

A.1 Attributes associated with Session

The following attributes show when you select a session in vManager GUI.

Attribute	Description
#Failed	Displays the number of failed runs in the session.
#Other	Displays the number of runs that did not complete as expected. For example, the runs that are stopped by a user action are mapped to this column.
#Passed	Displays the number of passed runs in the session.
#Running	Displays the number of runs that are currently executing.
#Waiting	Displays the number of runs that are waiting.
Automation file	Path of file that contains automation commands.

Attribute	Description
Bundling Policy	<p>Shows the bundling policy set for the selected session. Can be any of the following:</p> <ul style="list-style-type: none"> ■ <code>group_based</code> — Indicates that vManager will automatically bundle all calls to user scripts for each top-level group in the vsif, including all its sub-groups and tests, as a single DRM job. ■ <code>user_defined</code> — Indicates that <code>bundle_group</code> attribute was used to create custom bundles based on your knowledge of the length of the tests and the required resources. ■ <code>disabled (default)</code> —Indicates that vManager will ignore any <code>bundle_group</code> attributes in the vsif.
Comment	Shows the comments added for the selected session.
Copied Files	Shows the files to be copied into the session/group/run directory.
Default Dispatch Parameters	Specifies default options or parameters for the DRM ExecCommand.
Description	Shows the session description.
Drm	<p>Can be any of the following:</p> <ul style="list-style-type: none"> ■ <code>LSF</code> — Specifies the DPL-based integration with IBM Platform Load Sharing Facility (LSF). ■ <code>SGE</code> — Specifies the DPL-based integration with the open source batch-queuing system, Sun Grid Engine (SGE), supported by Sun Microsystems. ■ <code>LOADLEVELER</code> — Specifies the DPL-based integration with IBM's LoadLeveler. ■ <code>PARALLEL_LOCAL</code> — Specifies the DPL-based execution of runs in parallel on the local machine. ■ <code>SERIAL_LOCAL</code> — Specifies the DPL-based execution of runs serially on the local machine. ■ <code>USER_DEFINED</code> — Specifies DRM that you created using the DRM API.

Attribute	Description
Drm Xml Dir	Shows the directory that includes the DPL configuration files for different DRMs.
Duration (sec.)	Indicates the time duration of a session.
End Time	Indicates the end time of the session
Free Hosts	<p>Indicates whether to release a host machine after a DRM job completes or not.</p> <p>Default value is set to TRUE, which releases the host machine after completion of DRM job and allows fair resource sharing with other DRM users.</p> <p>If the value is set to FALSE, the host machine is not released and any additional jobs can be executed on this machine without requiring re-allocation.</p>
Hdl Files	Shows the Hdl files for a test.
Host Lock Timeout	<p>Shows the timeout for DRM job. It shows value in seconds after which no more tasks are sent to the job and the job quits after the current task ends.</p> <p>Default value is 0, which indicates no/infinite timeout.</p> <p>It shows a non-zero value only in the case of new Job runner and also if the free_hosts is set to FALSE.</p>
ID	Is the session identification number.
Linked Files	Shows the path of the files to which the symbolic link was created under the chain dir (in the session dir).
Log File	Shows the path name of the log file.
Master Submission Policy	Displays the dispatch parameters for the DRM master process.
Max Runs In Parallel	<p>Shows dispatch parameters for the DRM master process.</p> <p>Shows the keyword execute_locally or specifies additional parameters that are appended to the ExecCommand as defined in the configuration file when submitting the master process for execution.</p>
Model Dir	Shows the directory that contains the coverage model files of the session.

Incisive vManager User Guide

Attribute	Description
Name	Shows the name of the session. The name of the session is a unique automatically generated name by vManager in the following format: <session-container-name>.<user-name>.<time-stamp>
Number of Entities	Shows the number of entities in the selected group of sessions.
Original Session Dir	Displays the full pathname of the original session directory.
Original Vsif	Displays the full pathname of the original vsif file.
Output Mode	Shows either terminal or log_only. terminal — Indicates to see output interactively in a terminal window. log_only — Indicates chaining or running multiple sessions concurrently, to avoid multiple terminals opening simultaneously.
Owner	Displays the name of the owner of the session.
Parsed Vsif	Shows the name of the parsed vsif file.
Post Session Dispatch Parameters	Shows additional options or parameters for the DRM ExecCommand.
Post Session Drm	Shows the DRM for the post-session job.
Post Session Log File	Shows the location of log file generated after session run.
Post Session Script	Shows the script to run after all runs complete.
Post Session Dispatch Parameters	Specifies additional options or parameters for the DRM ExecCommand.
Pre Commands	Shows the commands to be executed before loading the top files.
Pre Session Drm	Shows the DRM for the post-session job.
Pre Session Log File	Shows the path of the log file where the output of pre_session_script (if specified) is redirected.
Pre Session Script	Shows the script to run before the start of a session.
Precious Env. Variables	Shows the environment variables that the runner should not override.

Attribute	Description
Queuing Policy	<p>Shows the queuing policy. It can be any of the following:</p> <ul style="list-style-type: none"> ■ <code>long2short</code>—Makes the longest runs first in the queue. This reduces the “long run tail” at the end of a session. ■ <code>short2long</code>—Makes the shortest runs first in the queue. This allows you to get the first test results from the regression as quickly as possible. ■ <code>round_robin</code>—Runs the tests in an arbitrary order, but creates a round-robin ordering between the groups and tests with a count greater than 1 or with multiple seeds. This mode is useful for vsifs that are arranged in logical groups, where you want to quickly see each type of test run once. ■ <code>vsif_order (default)</code>—Runs the tests in the order that they are defined in the vsif.
Run Mode	<p>Shows the run mode. It can be any of the following:</p> <ul style="list-style-type: none"> ■ <code>AS_IS</code>—To execute the run as specified in the original vsif. ■ <code>BATCH</code>—To execute the run in batch mode with timeout value as specified in the original vsif. ■ <code>BATCH_DEBUG</code>—To execute the run in batch-debug mode with timeout value as specified in the original vsif. ■ <code>INTERACTIVE</code>—To execute the run in interactive mode with timeout value as 0, which means timeout is disabled. ■ <code>INTERACTIVE_DEBUG</code>—To execute the run in interactive-debug mode with timeout value as 0, which means timeout is disabled.
Session Dir	Displays the full pathname of the session directory.

Incisive vManager User Guide

Attribute	Description
Session Status	<p>Displays the status of each session. It can be any of the following values:</p> <ul style="list-style-type: none"> ■ <code>pre_session_script</code>—Indicates that the <code>pre_session_script</code> is being run. ■ <code>pre_session_script_done</code>—Indicates that the <code>pre_session_script</code> is done. ■ <code>in_progress</code>—Indicates that the session is running. ■ <code>post_session_script</code>—Indicates that the <code>post_session_script</code> is being run. ■ <code>post_session_script_done</code>—Indicates that the <code>post_session_script</code> is done. ■ <code>completed</code>—Indicates that the <code>pre_session_script</code>, the run, and the <code>post_session_script</code> have completed without errors (No failures with severity <code>error</code> or <code>critical</code>). ■ <code>failed</code>—Indicates that the session has failed. (A failure of severity <code>error</code> or <code>critical</code> has occurred in one of the session, processes, or scripts.) ■ <code>stopped</code>—Indicates that the session has been stopped by a user action. ■ <code>suspended</code>—Indicates that the session has been suspended by a user action. ■ <code>inaccessible</code>—Indicates that the session cannot be accessed because the run count limit has exceeded the limit of runs specified for the license.
Start Session Command	Shows the start session command to start the vsif file.
Start Time	<p>Shows the date and time when the session started.</p> <p>Note: The dates are shown based on the locale set for your system.</p>
Top Dir	Shows the top-level directory for the session
Top Files	Shows the top-level files for a test.

Attribute	Description
Total Runs	<p>Displays the total number of runs in the session. It is calculated as the sum of values in the #Passed, #Failed, #Running, #Waiting, and #Others columns.</p> <p>The values shown in these columns are affected by the configuration options related to runner, rerun, and rerun filters.</p>
Type	Shows the session generation mechanism. Possible values are: runner, show_coverage, single_subrun, single_run, collected, vm_scan.
Tracking Snapshots	Shows the tracking configurations and snapshots it contributed.
User Merged Coverage Directory	<p>Shows the path of the merged coverage files (.ucm and .ucd) that were attached to a session using the <i>Edit each</i> or <i>Edit all at once</i> dialog boxes. For more details, see Associating Merged Coverage with Sessions on page 110.</p> <p>A value in this field indicates that the user has attached merged coverage with this session and if any of the operation requires a merge, vManager will use the value in this field instead of merging automatically.</p> <p>No value in this field indicates that automatic merge mechanism will be used for any operation that requires merge.</p>
Verification Scope	Shows the verification scope for a session, group, or test.
Version	Shows the session version number.

A.2 Attributes associated with Test or Tests Group

The following attributes show when you select a test or a test group in vManager GUI.

Attribute	Description
#Failed	Displays the number of failed run instances of that test. For tests groups, the value is accumulated from all leaf tests.

Attribute	Description
#Passed	Displays the number of passed run instances of that test. For tests groups, the value is accumulated from all leaf tests.
#Other	Displays the number of incomplete runs of the test. For tests groups, the value is accumulated from all leaf tests. It is calculated as: $(\#Runs - (\#Passed + #Failed))$
#Runs	Displays the total number of run instances of selected test.
Is Active	true if the run is active false if the run is inactive
Name	Displays the name of test or test group
Overall Average Grade	Displays the overall average grade of the selected item
Overall Covered	Displays the overall covered runs of the selected item
Overall Covered Grade	Displays the overall covered grade of the selected item
Overall Total	Displays the overall total runs of the selected item
Test Case Grade	Displays the test case grade of the selected item
Test Status	Shows the ratio of passed runs out of the total runs $(\#Passed / \#Runs) * 100\%$
Verification Scope	Shows the verification scope of the selected test group or test

A.3 Attributes associated with Run

The following attributes show when you select a run in vManager GUI.

Attribute	Description
#Failures	Shows the number of failures in the run
#Passed	Shows the number of times the run passed
#Failed	Shows the number of times the run failed

Attribute	Description
%Passed	<p>Shows 100% for run with status as <i>passed</i>, else 0%</p> <p>Note: This attribute is of use in the case of grouped runs. In case of grouped runs, the data would be the percentage of passed runs in per each group.</p>
%Failed	<p>Shows 100% for run with status as <i>failed</i>, else 0%</p> <p>Note: This attribute is of use in the case of grouped runs. In case of grouped runs, the data would be the percentage of failed runs in per each group.</p>
Auto scan args	<p>Shows the user arguments that are passed to scanning commands, if any. The log scanning is executed in two steps:</p> <p>Executing scanning specifications in the vsif as defined by <code>scan_script</code> attribute.</p> <p>Complementary scanning is done automatically when using Cadence tools such as Incisive Enterprise Simulator.</p> <p>While the first step is defined by the user the second step is done automatically.</p>
ALM_KEY	<p>When attaching a run to a third-party ALM system (ALM – Application Lifecycle Management), this attribute can contain the third-party entry id. For example, if a run was linked to a Jira Bug, this entry will contain the Jira ID. When working with OpsHub integration, this will be taken care automatically.</p>
Bundle Group	<p>If TRUE, the entire group, including pre-session and post-session scripts as well as the entire sub-hierarchy of groups and tests are bundled together into one DRM job. If FALSE (the default), no user-defined bundling specifications are applied.</p>
CPU Time (ms.)	<p>Displays the total CPU time for a run in seconds.</p> <p>Note: CPU time indicates the amount of time for which a CPU was used for processing instructions of a computer program or the operating system.</p>
Code Coverage	Indicates if the code coverage collection is turned on or off.
Comment	Shows the comments added for the selected run.

Incisive vManager User Guide

Attribute	Description
Compaction	Indicates if the run is compact or not. COMPACT indicates that the run is the result of compaction. FLAT indicates that the run was created during simulation.
Copied Files	Shows the files to be copied into the session/group/run directory.
Default Dispatch Parameters	Indicates the default options or parameters for the DRM ExecCommand.
Depends On	Indicates the dependency between tests.
Drm	Shows the DRM used for a session, group, or test. It can be any of the following: <ul style="list-style-type: none"> ■ LSF — Specifies the DPL-based integration with IBM Platform Load Sharing Facility (LSF). ■ SGE — Specifies the DPL-based integration with the open source batch-queuing system, Sun Grid Engine (SGE), supported by Sun Microsystems. ■ LOADLEVELER — Specifies the DPL-based integration with IBM's LoadLeveler. ■ PARALLEL_LOCAL — Specifies the DPL-based execution of runs in parallel on the local machine. ■ SERIAL_LOCAL — Specifies the DPL-based execution of runs serially on the local machine. ■ USER_DEFINED — Specifies DRM that you created using the DRM API.
Drm Job Bundle	Shows the bundle to which the selected run belongs. Note: When using Bundling Policy run are submitted in bundles (multiple runs per job).
Drm Job Id	Indicates the ID of the DRM job that contains the specified run.
Drm Job Name	Indicates the name of the DRM job that contains the specified run.

Incisive vManager User Guide

Attribute	Description
Drm Job Status	Indicate the status of the DRM job that contains the specified run. Possible values are: <ul style="list-style-type: none"> ■ waiting — The master DRM process is setting up the queue or the job is waiting for execution on the DRM queue. ■ running — The job is executing. ■ stopped — Due to a user stop action on the session, the job was terminated during execution. ■ finished — The job has completed normally. ■ other — The job has terminated in some abnormal way. Due to a user stop action on a single run or a group of runs, the run was removed from the DRM queue before execution started or was killed upon starting.
Duration (sec.)	Indicates the time duration of a run.
Dut Name	Shows the name of the device under test (DUT).
End Time	Indicates the end time of the run.
Engine	Shows the engine using which the run was collected.
Exit On	Defines the Specman exit condition. All options correspond to the Specman config run -exit_on command: command, normal_stop, error, all.
First Failure Name	Displays the name of the first failure of selected run.
First Failure Description	Displays text of the first failure of a run.
First Failure Original Description	Is a reference to the Original Description attribute of failure.
First Failure Severity	Displays the severity of the first failure. Possible values are error or critical.
First Failure Time	Displays the time of the first failure of a run. This attribute specifies the failure time, when applicable, in simulation cycles.
First Failure Tool	Displays the tool reporting the first failure of a run. Possible values include Specman, Modelsim, SystemC, and so on.
GUI Mode	Shows if the interactive mode is turned on or off.

Attribute	Description
Hdl Files	Shows the HDL files for a test.
Host	Indicates the name of the host machine.
ID	Indicates the run ID number. When you execute each run, the runner assigns a numeric run ID to each run.
Index	Displays the run identification number.
Job Status	Indicate the status of the DRM job that contains the specified run. Possible values are: <ul style="list-style-type: none"> ■ waiting — The master DRM process is setting up the queue or the job is waiting for execution on the DRM queue. ■ running — The job is executing. ■ stopped — Due to a user stop action on the session, the job was terminated during execution. ■ finished — The job has completed normally. ■ other — The job has terminated in some abnormal way. Due to a user stop action on a single run or a group of runs, the run was removed from the DRM queue before execution started or was killed upon starting.
Linked Files	Shows the path of the files to which the symbolic link was created under the chain dir (in the run dir).
Log File	Shows the pathname of the log file.
Model Dir	Shows the directory that contains the coverage models.
Name	Displays the complete hierarchical name of the run.
Number of Entities	Displays the number of entities in the selected group of runs.
Owner	Shows the owner of the session to which the selected run belongs.
Parent Run	Displays the complete path of the parent run of the selected run.
Post Commands	Shows the commands to be executed before quitting.

Attribute	Description
Post Group Script	Shows the script to be executed after a group of runs complete.
Post Run Log File	Shows path to the log file of post_run_script (if specified).
Post Run Script	Shows the script to run after the scan script completes and before the post-session script.
Post Session Log File	Shows the complete path of the log file generated after session run.
Post Simulate Script	Shows the script to run after the single run script completes and before the scan script is called.
Primary Run	<p>Indicates if the selected run is the primary run. Shows true if the run is the primary run, else shows false.</p> <p>The default value of this attribute is false. You can set the value of this attribute by selecting the <i>Primary Run</i> check box in the <i>Edit each</i> dialog box.</p> <p>If multiple runs are specified as primary run, then the run with the smallest run ID is selected as the primary run for the merge operation.</p> <p>If none of the runs are specified as primary runs, then the run with the smallest run ID is selected as the primary run for the merge operation.</p> <p>Note: In case of merging coverage from multiple sessions, the primary run will always be selected from the runs of the session with the latest timestamp, even if you have marked a run from another session as the primary run.</p>
Revision	Shows the revision information of each run.
Run Dir	Displays the full pathname of the directory containing all the results for a particular run, including the log files and the coverage file.
Run Mode	Shows the mode of the run. It can be batch, interactive, batch_debug, or interactive_debug.
Run Script	Shows the script that invokes the combined simulator/vManager executable with the SVE.
Runs Dispatch Parameters	Shows additional options or parameters for ExecCommand.

Incisive vManager User Guide

Attribute	Description
Runs #	Shows the number of runs in the selected group.
SV Seed	Shows the seed for the test to which this run belongs. It can be any of the following: random, gen_random, positive_gen_random
Scan Script	Shows the script that scans simulation output for failures.
Seed	Shows the seed that the runner passes automatically to the Specman generator.
Seed Domain	Shows the seed for seeds generation.
Session Name	Displays the name of the session.
Simulation Time	Displays the total simulation time for a run in cycles.
Simulator	Displays the name of the simulator.
Start Time	Indicates the start time of a run.
Status	Indicates the pass/fail status of the specified run. The possible values of this attribute are: <ul style="list-style-type: none"> ■ failed—an error of fatal or critical severity occurred during the run. For IFV, a run is marked as failed if the status of any assertion is Fail, Block or Abort. ■ passed —no error of fatal or critical severity occurred during the run, and the run completed normally. For IFV, a run is marked as passed if the status of all assertions have the status Pass. ■ incomplete —no error of fatal or critical severity occurred during the run, but the run did not complete normally. For example, you manually stopped the run or the session. For IFV, run is marked as incomplete, if the status of all assertions is Not_Run, Constraint, or Explored.
Sve Name	Shows the sve file to set environment variables required to control the run.
System Time (ms.)	Displays the total system time for a run in seconds. Note: System time indicates the amount of time the CPU was busy executing code in the kernel space.

Attribute	Description
Test Command	Displays the command to start a test. Can be one of test, gen, setup, none.
Test Group	Displays the full pathname of the directory containing test group.
Test Group Base Name	Displays the test group name.
Test Name	Displays the test name of a run.
Timeout (sec.)	Is a non-negative number representing the number of seconds before the run is killed.
Top Files	Shows the top-level files for a test.
Type	Shows the run type, such as simulation, formal, formal_replay, and so on.
User Time (ms.)	Displays the total user time for a run. Note: User time indicates the amount of time the CPU was busy executing code in user space.
Verbosity	Shows whether verbosity is turned on or off.
Verification Scope	Shows the verification scope of the session, group, or test.
Waveform	Shows whether waveform is turned on or off.
Waveform Command	Shows the command used to open the SimVision Waveform window.
vSif	Displays the pathname of the vsif file that created the run.
Is Rerun	Shows TRUE for runs that are rerun Shows FALSE for runs that are not rerun.

A.4 Attributes associated with Failure

The following attributes show when you select a failure in vManager GUI.

Attribute	Description
Category	Displays the category of selected failure. Possible values are: first or others.

Attribute	Description
Comment	Shows the comment added for the selected failure.
Context	Shows the struct or unit instance where the failure occurred.
Description	Shows the description of the failure.
ID	Is the failure identification number.
Module	Shows the module and line number where the failure occurred.
Name	Indicates the name of failure. Possible values include os_signal, dut_error, assertion, and so on.
Number of entities	Displays the number of entities within the selected group of failure.
Original description	Displays the original description of the failure.
Owner	Displays the owner of the session to which the selected failure belongs.
Severity	Shows the severity level of the failure. Possible values can be informative, warning, error, critical.
Time	Indicates when the failure occurred.
Tool	Shows the name of the tool that reported the failure.

A.5 Attributes associated with Formal Properties

The following attributes show when you select a formal property in vManager GUI.

Attribute	Description
CPU Time (ms.)	Displays the total CPU time for a property in seconds. Note: CPU time indicates the amount of time for which a CPU was used for processing instructions of a computer program or the operating system.
Category	Displays the category of selected property. Possible values are: first or others.
Comment	Shows the comment added for the selected property.

Attribute	Description
Constraints	Shows the constraints used in the property.
Context	Shows the struct or unit instance where the property occurred.
Counter Example Command	Shows the command used to open the JasperGold counter example waveform window.
Depth	Shows the property depth.
Description	Shows the description of the property.
Editing	Shows the type of editing done to the property. It can be none, light, or severe.
Has Counter Example	TRUE if the property has a counter example waveform, else FALSE.
Has Trace Waveform	TRUE if the property has a trace waveform, else FALSE.
Has Trigger Waveform	TRUE if the property has a trigger waveform, else FALSE.
ID	Is the property identification number.
Instance	Shows the instance to which the property belongs.
Log File	Shows the path name of the log file.
Log Line	Shows the line number of the log file on which associated log is written.
Module	Shows the module to which the property belongs.
Name	Indicates the name of the property.
Number of entities	Displays the number of properties within the selected group of failure.
Original description	Displays the original description of the failure.
Owner	Displays the owner of the session to which the selected property belongs.
Real Time	Displays the actual time of a property in seconds.
Severity	Shows the severity level of the property. Possible values can be informative, warning, error, critical.
Source File	Path of the source file.
Source Line	Line number on which the property is defined.

Incisive vManager User Guide

Attribute	Description
Status	Shows the status of the property.
Time	Indicates when the failure occurred.
Tool	Shows the name of the tool that reported the failure.
Trace Depth	Shows the property trace depth.
Trace Status	Shows the trace status of the property.
Trace Waveform Command	Shows the command used to open the JasperGold trace waveform window.
Trigger Depth	Shows the property trigger depth.
Trigger Status	Shows the trigger status of the property.
Trigger Waveform Command	Shows the command used to open the JasperGold trigger waveform window.
Type	Can be cover, check, or constraint.

Invocation and Command-Line Interface

This chapter covers the following topics:

- [Launching vManager](#) on page 383
- [Batch-Mode Commands](#) on page 404

B.1 Launching vManager

The `vmanager` command is used to launch vManager.

The command without any options, by default, launches vManager in client-server (CS) mode.

If you want that by default vManager is launched in Incisive Enterprise Manager (IEM) mode, set the environment variable `VMANAGER_INVOCATION_MODE` to `IEM` using the following command:

```
setenv VMANAGER_INVOCATION_MODE IEM
```

After the above environment variable is set to `IEM`, by default, vManager will be launched in Incisive Enterprise Manager (IEM) mode.

If required, you can switch back to CS mode by setting the environment variable `VMANAGER_INVOCATION_MODE` to `CS`.

Note: If you want to launch vManager in IEM mode for this instance only (not always), you can use the following command:

```
vmanager -iem
```

If both command-line option (`-iem`) as well as the environment variable is used, then the command-line option takes precedence.

B.1.1 Valid Modes for Launching vManager

You can launch vManager in any of the following mutually exclusive modes:

- GUI mode (Opens the Graphical User Interface)

To launch vManager in GUI mode, use the `vmanager -gui` command. This is the default mode. If `-init <tcl_file>` option is used with the `vmanager -gui` command, the `<tcl_file>` will be executed before launching vManager in GUI mode.

- Command-line Interactive mode (Opens the command-line interface)

To launch vManager in command-line interactive mode, use the `vmanager -batch` command. If `-init <tcl_file>` option is used with the `vmanager -batch` command, the `<tcl_file>` will be executed before launching vManager in command-line interactive mode.

- Batch mode (Allows execution of a batch script or commands and exits)

To launch vManager in batch mode, execute a batch script and exit, use the `vmanager -exec <command_file>` command. If `-init <tcl_file>` option is used with the `vmanager -exec <command_file>` command, then the `<tcl_file>` will be executed before executing the `<command_file>`.

To launch vManager in batch mode, execute the specified TCL command(s), and exit, use the `vmanager -execcmd <command>` command. You can specify more than one TCL command on the command line by separating them with a semicolon (;). For example, to launch vManager, run the `help` and `cdnshelp` commands and exit, use the following command:

```
vmanager -execcmd help;cdnshelp
```

Command Syntax

The syntax for the `vmanager` command is:

```
vmanager [-batch | -gui | -exec <command_file> | -execcmd <command>] [options]
```

where

- `-batch` launches vManager in Command-line Interactive mode (Opens the command-line interface).
- `-gui` launches vManager in GUI mode (Opens the Graphical User Interface).
- `-exec <command_file>` launches vManager in batch mode, executes the specified batch script, and exits.
- `-execcmd <command>` launches vManager in batch mode, executes the specified TCL commands, and exits.

Note: If none of the options (`-batch`, `-exec`, `-execcmd`, or `-gui`) is specified, `-gui` is assumed.

- [options] can be any of the following:

- [Application-specific Options](#)
 - [Generic Options](#)

Application-specific Options

- Following are the application-specific options of vManager:

<u>-help</u>	<u>-status</u>	<u>-version</u>
<u>-memlimit <limit></u>	<u>-nocopyright</u>	<u>-launch <vsif></u>
<u>-init <file></u>	<u>-logfile <file></u>	<u>-appendlog</u>
<u>-keyfile <file></u>	<u>-appendkey</u>	<u>-verbose</u>
<u>-quiet</u>	<u>-nostdout</u>	<u>-licqueue</u>
<u>-sourcemap <file></u>	<u>-load_vplan <vplan_file></u>	<u>-load_session <session_name></u>
<u>-load_refinement <refinement_file></u>	<u>-load_vplan_refinement <refinement_file></u>	<u>-planner</u>
<u>-metrics_only</u>	<u>-server <server_url></u>	<u>-analyze_all_runs</u>
<u>-analyze_failed_runs</u>	<u>-analyze_failures</u>	<u>-analyze_incomplete_runs</u>
<u>-analyze_metrics</u>	<u>-analyze_passed_runs</u>	<u>-analyze_test</u>
<u>-analyze_vplan</u>	<u>-display <arg></u>	<u>-enable_tracking</u>
<u>-regr or -regrcenter</u>	<u>initcmd <command></u>	<u>-local [<dir>]</u>
<u>-migrate_local</u>	<u>-migrate_local_backup_dir [<dir>]</u>	<u>-standalone [<dir>]</u>
<u>-context <context_name></u>	<u>-admin</u>	<u>-jp</u>
<u>-sessions_view <view></u>	<u>-filter <filter></u>	<u>-analysis_view <view></u>
<u>-analysis_filter <filter></u>	<u>-open_planner <vPlan_file></u>	<u>-planner_view <view></u>
<u>-planner_filter <filter></u>	<u>-open_tracking <tracking_config></u>	

-help

Displays a list of the `vmanager` command options with a brief description about each option.

-status

Prints run-time statistics (memory usage and CPU usage details) when you exit out of vManager.

-version

Prints the version number of vManager.

-memlimit <limit>

Sets the heap size of Java process. This will not change the C++ heap and will not impact coverage merging or reporting performance.

Examples:

```
vmanager -memlimit 4000M  
vmanager -memlimit 4G
```

-nocopyright

Suppresses printing of copyright banner.

-logfile <file>

Uses *<file>* as the log file instead of the default `vmanager.log`.

-appendlog

Appends log information from multiple runs of vManager to one log file. Use this option if you are going to run vManager multiple times, and you want all log information appended to one log file. If you do not use this option, the log file is overwritten each time you run vManager.

-keyfile <file>

Uses *<file>* as the key file instead of the default `vmanager.key`.

`vmanager.key` is the key file that is generated automatically by vManager when you launch vManager in batch mode. This file stores all interactive commands that you have issued, including misspelled commands and the `exit` command. By default, vManager names the

key file as `vmanager.key`. With the `-keyfile <file>` option, you can cause vManager to name the key file as `<file>` instead of the default `vmanager.key`.

Key files are useful when you want to reproduce a vManager session. To reproduce an interactive session, specify the name of the key file with the `-init` option. The commands in the key file are executed at the beginning of the vManager session. When vManager has processed all commands in the file, or if you interrupt processing with `CTRL-C`, input reverts back to the terminal.

You can edit the key file, as required.

-appendkey

Appends command input from multiple runs of vManager to one key file. Use this option if you are going to run vManager multiple times and you want all command input appended to one key file. If you do not use this option, the key file is overwritten each time you run vManager.

By default, vManager generates a key file called `vmanager.key` to capture command input. Use the `-keyfile` option to rename the key file.

-init <file>

Executes the vManager commands in the specified `<file>` at the beginning of the vManager session.

initcmd <command>

Executes the specified TCL command(s) at the startup of vManager.

You can specify more than one TCL command on the command line by separating them with a semicolon (`;`). In addition, the commands to be executed must be enclosed within double quotes (`" "`) or single quotes (`' '`).

For example, the following command launches vManager in command-line interactive mode, and then immediately executes the `help` command and opens the *vManager User Guide*.

```
vmanager -batch -initcmd "help; cdnshelp -user_guide"
```

-launch <vsif>

Starts vManager and launches the specified vsif file immediately after starting vManager.

For example, the following command will start vManager and immediately launch the `cdn_uart_apb_regress.vsif` file.

```
vmanager -launch cdn_uart_apb_regress.vsif
```

-nostdout

Disables printing of informative summary messages to the standard output device. By default, vManager prints all messages to the standard output device as well as the `vmanager.log` file.

-quiet

Suppresses the printing of summary messages in the `vmanager.log` file. Use of this option enhances the readability of the log file because it suppresses the output of verbose informational messages.

Note: It does not suppress tool warning, error messages, tool banner, and the command-line arguments that were used to launch the tool.

-verbose

Prints informative summary messages during commands execution to the `vmanager.log` file.

-licqueue

Queues the request for license, if not currently available, and runs the tool when the license becomes available.

-sourcemap <file>

Reads and applies source mappings specified in `<file>`. The `<file>` must include pair of paths, one per line. This option is useful when you want to map the source to new paths at the time of invoking vManager. To use this option, specify all the mappings in a separate file and then pass it to `vmanager` with the `-sourcemap` option.

For more details on mapping source paths, see the [sourcemap](#) command.

-load_vplan <vplan_file>

Launches vManager and loads the specified vPlan file (<vplan_file>).

For example, to load vPlan file named `newplan27_3.vplanx` at the time of launching vManager, use:

```
vmanager -load_vplan /home/bob/ivm/ep/newplan27_3.vplanx
```

-load_session <session_name>

Launches vManager and loads the specified sessions (<session_name>).

You can specify more than one session by separating the session names with a comma (,).

You can also specify the name of the view instead of the session name. By specifying a view name, the view is loaded and launched at the time of launching vManager.

For example, to load a session named `cdn_uart.segal.11_01_04_23_14_13_1797` at the time of launching vManager, use:

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797
```

To load sessions named `cdn_uart.segal.11_01_04_23_14_13_1795` and `cdn_uart.bob.12_06_13_14_59_26_6338` at the time of launching vManager, use:

```
vmanager -load_session  
cdn_uart.segal.11_01_04_23_14_13_1795,cdn_uart.bob.12_06_13_14_59_26_6338
```

To load sessions from view named `my_failed_sessions`, use:

```
vmanager -load_session my_failed_sessions
```

To load sessions from views named `my_passed_sessions` and `my_cdn_sessions`, use:

```
vmanager -load_session my_passed_sessions,my_cdn_sessions
```

Note: You can also specify the `-filter` switch with the `-load_session` switch, as stated below:

```
-load_session "<view> -filter <filter>"
```

where:

```
<filter> = <attribute>:<operator><value>
```

For example:

```
vmanager -batch -load_session "All_Sessions -filter name:~comp"  
-analyze_failed_runs
```

-load_refinement <refinement_file>

Launches vManager and loads the specified refinement file (<refinement_file>).

You cannot specify multiple refinement files on the command line.

Note: Session must be loaded before loading a refinement file.

To launch vManager in batch mode, and to load a refinement file named `my_exclude.vRefine` at the time of launching vManager, use:

```
vmanager -batch -load_session cdn_uart.segal.11_01_04_23_14_13_1797  
-load_refinement /home/bob/ivm/ep/analyze/my_exclude.vRefine
```

-load_vplan_refinement <refinement_file>

Launches vManager and loads the specified vPlan refinement file (<refinement_file>).

You cannot specify multiple vPlan refinement files on the command line.

Note: Session and verification plan must be loaded before loading a vPlan refinement file.

To launch vManager in batch mode, and to load a vPlan refinement file named `my_vp_exclude.vpRefine` at the time of launching vManager, use:

```
vmanager -batch -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -load_vplan  
/home/bob/ivm/ep/newplan27_3.vplanx -load_vplan_refinement  
/home/bob/ivm/ep/my_vp_exclude.vpRefine
```

-planner

Launches vPlanner GUI.

-metrics_only

Launches the IMC GUI for metrics analysis.

-server <server_url>

Connects to the vManager server using the specified URL.

-analyze_all_runs

Immediately opens the *All Runs* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *All Runs* view.

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -analyze_all_runs
```

-analyze_failed_runs

Immediately opens the *Failed Runs* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *Failed Runs* view.

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -analyze_failed_runs
```

-analyze_failures

Immediately opens the *First Failures* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *First Failures* view.

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -analyze_failures
```

-analyze_incomplete_runs

Immediately opens the *Incomplete Runs* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *Incomplete Runs* view.

-analyze_metrics

Immediately opens the *Metrics* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *Metrics* view.

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -analyze_metrics
```

-analyze_passed_runs

Immediately opens the *Passed Runs* view at the time of launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *Passed Runs* view.

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -analyze_passed_runs
```

-analyze_test

Immediately opens the *Test_Hierarchy* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified session, and opens the *Test_Hierarchy* view.

```
vmanager -load_session cdn_uart.segal.11_01_04_23_14_13_1797 -analyze_test
```

-analyze_vplan

Immediately opens the *vPlan* view after launching vManager.

For example, the following command-line launches vManager in GUI mode, loads the specified vplan, and opens the *vPlan* view.

```
vmanager -load_vplan /servers/scratch03/bob/doc_examples/my_plan.vplanx  
-analyze_vplan
```

-display <arg>

Specifies the screen on which the window must be displayed.

-enable_tracking

Enables the vManager *Tracking Center*. By default, the *Tracking Center* is disabled. For more details, see [Project Tracking](#) on page 319.

-regr or -regrcenter

Launches vManager in *Regression Center Launcher* client mode. In this mode, only the session management and metrics analysis functionality is available.

The following CLI commands are not supported in this mode:

- report_vplan
- report_tests
- convert
- csv_export -vplan

- rerun
- edit (supported only for modifying session's attributes and not for run or failure attributes)
- vscan
- snapshot

Note: These commands will show in help but will result in an error when executed in the *Regression Center Launcher* client mode. When in *Regression Center Launcher* client mode, you can upgrade to the *Full* client mode (which includes all the commands and functionality) using the `upgrade` CLI command. For more details, see [upgrade](#) on page 409.

The following invocation options are also not available in the *Regression Center Launcher* client mode:

- `-load_vplan`
- `-load_vplan_refinement`
- `-enable_tracking`

Note: An error is reported if you use the above-mentioned options along with the `-regr` or `-regrcenter` options at the time of launching vManager.

-local [<dir>]

Indicates that you want to launch vManager in local mode. In local mode, vManager client runs without any server connection setup; however, there are a few limitations when running vManager in local mode.

Note: Local mode means one server and one client together. Only one server or one client (batch or GUI) can be invoked at a given time from a given directory. Therefore, if you have invoked a GUI client, then you cannot invoke a batch client or any other GUI client from the same directory and vice versa.

When running vManager in local mode, you can specify the local mode storage directory using the optional `<dir>` option. If the local mode storage directory is not specified, then by default, the `<current_working_directory>/vmgr_db` is assumed as the local mode storage directory.

For example, to run vManager in local mode and to specify the local mode storage directory as `/servers/scratch03/mike/`, use the following command:

```
vmanger -local /servers/scratch03/mike/
```

-migrate_local

Performs migration when vManager is launched in local mode and migration is required.

-migrate_local_backup_dir [<dir>]

Specifies the backup directory to be used to save the current vManager local mode data before migration.

-standalone [<dir>]

The `-standalone` option works with the `-planner` option.

When launching vPlanner in standalone mode, you can specify the storage directory using the optional `<dir>` option. If the storage directory is not specified, then by default, the `$USER_HOME/.mdv` will be used.

For example, to launch vPlanner in local mode and to specify the local mode storage directory as `/servers/scratch03/sess1/`, use the following command:

```
vmanager -planner -standalone /servers/scratch03/sess1/
```

-context <context_name>

Launches vManager with the specified context (`<context_name>`).

The `<context_name>` is an XML file that contains context data, which would have been exported earlier using vManager GUI.

Using this option, you can invoke vManager with the specified context launched.

For example, to invoke vManager with `my_context.xml` launched, use the following command:

```
vmanager -context /servers/scratch03/my_context.xml
```

In this case, `my_context.xml` would have already been exported using vManager GUI.

It saves a lot of time as you do not need to open all the pages again and you can start analysis immediately from the point where you left it. You can share the context with other users on the same server. For more details on exporting context data, see [Export Active Context](#) on page 17.

-admin

Launches vManager with *Administration* activity center.

When you use this option, a new center named *Administration* is available, as shown in [Figure B-1](#) on page 395.

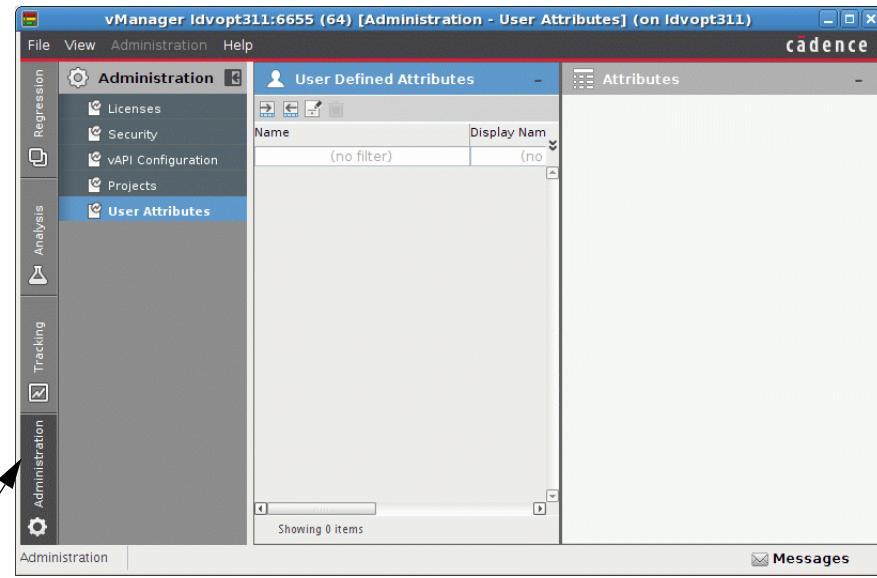
Figure B-1 vManager Administration Center



Click to launch *Administration* center

When you click the *Administration* Center link, the *Administration* center interface is displayed, as shown in [Figure B-2](#) on page 396.

Figure B-2 vManager Administration Center

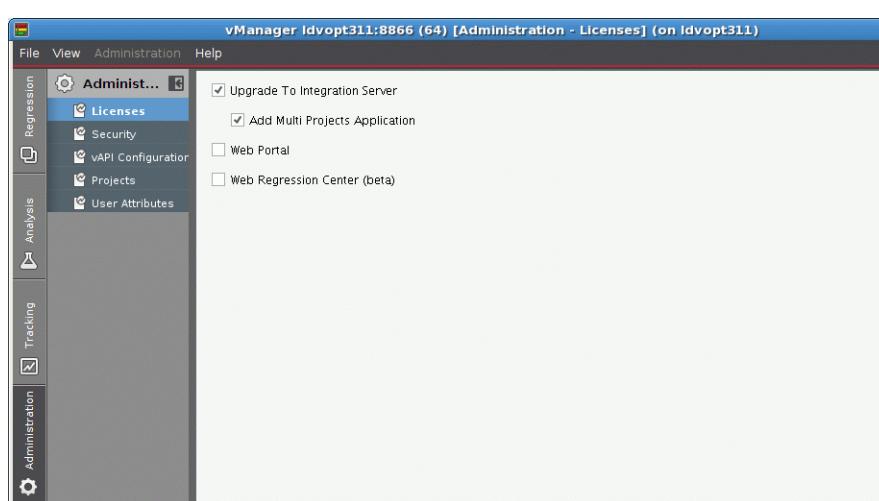


The *Administration* center is similar to the *Profile Editor* interface. It has following sections:

- Licenses — To manage licenses.

[Figure B-3 on page 396 shows the *Licenses* page of *Administration Center*.](#)

Figure B-3 vManager Administration Center (Licenses Page)

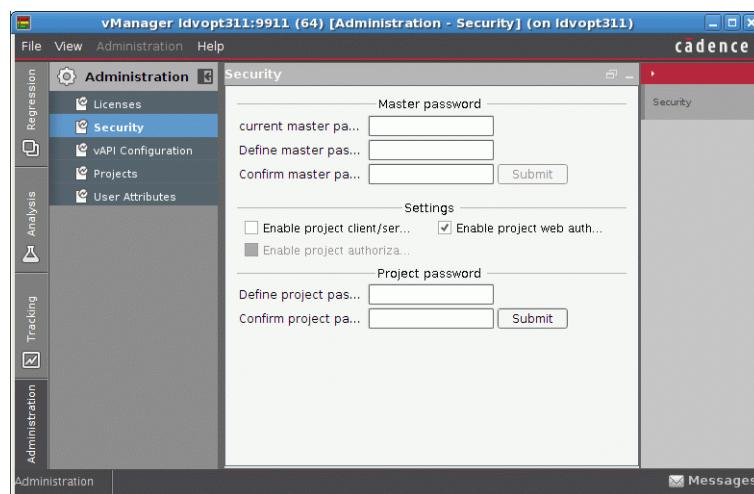


You can do any of the following:

- ❑ Upgrade to Integration Server — This will upgrade your license to VMG200, which allows unlimited runs and 1 tracking configuration.
 - Add Multi Projects Application — This option is enabled only after you upgrade to integration server. This option allows you to add unlimited tracking configurations. This license code is VMGA01.
- ❑ Web Portal — This will add the *WEB_PORTAL* license and allow you to launch the Web portal and perform administration tasks from there. For more details, see the *vManager Web User Guide*.
- ❑ Web Regression Center (Beta) — Contact Cadence representative for more details.
- Security — To manage security settings.

Figure B-4 on page 397 shows the *Security* page of *Administration Center*.

Figure B-4 vManager Administration Center (Security Page)



The *Security* page has following options:

- ❑ Current master password—Allows you to specify the current master password. It is mandatory to provide the current master password at the time of changing the master password. This prevents users who are not aware of the existing password from changing it.
- ❑ Define master password—To define a master password.
- ❑ Confirm master password—To re-type master password and confirm it.

Note: The master password can also be defined and changed using the *vmgrconf* utility. For more details, see [Incisive vManager Installation and Configuration](#).

Guide.

- ❑ Enable client/server authentication—To enable/disable password authentication at the time of launching vManager. By default, this option is not selected. Once you select this check box, the user is prompted to enter the master password at the time of launching vManager.

Note: To skip the prompt for user name and password at the time of launching the tool, you can set the environment variables VMGR_USERNAME and VMGR_PASSWORD. This will allow the users to run in batch mode (may be using a batch script tcl file) without the interactive prompt. Otherwise, the tool will prompt for the user name and password.

- ❑ Enable web authentication—To enable/disable Web authentication of vManager.

Note: vManager exposes web capabilities for the following:

- Debug capabilities and health care information for technical debug via web-client
- Rest API
- Web Dashboard (Beta)

It is a best practice for those channels to be secured by default to prevent unnecessary or un-authorized access. This option is therefore selected by default.

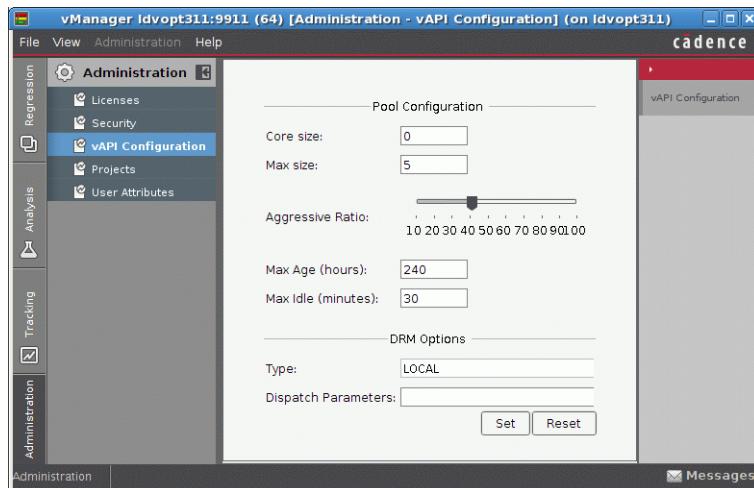
- ❑ Define project password—To define a project password.
- ❑ Confirm project password—To re-type project password and confirm it.

Note: Only one project password can be applied on a project. You can change or reset the project password after you log in to the specified project. In case you forget the project password, you can use the master password to login and then reset the project password in vManager GUI.

- vAPI Configuration— To configure the proxy manager of vAPI.

Figure B-5 on page 399 shows the *vAPI Configuration* page of *Administration Center*.

Figure B-5 vManager Administration Center (vAPI Configuration Page)



The *vAPI Configuration* page lets you specify the following:

- ❑ Pool configuration settings
 - Core size — Lets you specify the vAPI minimum pool size.
 - Max size — Lets you specify the vAPI maximum pool size. It defines the maximum number of processes that can run on the proxy manager.
 - Aggressive Ratio — Lets you define how fast the system will take the pool size from the *Core size* to the *Max size*.
 - Max Age (Hours) — Lets you specify the maximum age of a vAPI processes after which it will shut down.
 - Max Idle (minutes) — Lets you specify the time after which the non-working vAPI processes will shut down.
- ❑ DRM options
 - Type — Lets you select the DRM type to be used for running the vAPI process.
 - Dispatch Parameters — Lets you specify the dispatch parameters for the selected DRM type.

Note: After specifying the appropriate values in the above fields, click the *Set* button. The *Reset* button is used to reset the values to the last set values.

- Projects— To add additional projects. For more details, see [Incisive vManager Installation and Configuration Guide](#).

- User attributes — To import and export user-defined attributes. For more details, see [Incisive vManager Installation and Configuration Guide](#).

-jp

Enables Japanese keyboard input support and sets the default input language as Japanese.

For example, if you use the vmanager -jp command, then SCIM (Smart Common Input Method) is used as the default input engine and Japanese is set as the default language.

After this command, you can provide input to vManager in Japanese language from the keyboard.

-sessions_view <view>

Launches vManager with the specified sessions view (<view>).

Using this option, you can invoke vManager with the specified view loaded at startup.

For example, to invoke vManager with my_sessions loaded, use the following command:

```
vmanager -sessions_view my_sessions
```

-filter <filter>

Applies changes on the main table of the view.

Usage example:

```
vmanager -cs -load_session <session_name> -sessions_view <session> -filter <filter>
```

Note: The format for specifying the filter is:

```
[attribute_name] : [operator] [value]
```

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example: -filter name:~comp

-analysis_view <view>

Launches vManager with the Analysis center and sets the specified analysis view (<view>).

This option should be used with any of the -analyze_* options, such as -analyze_all_runs, -analyze_failures, and so on.

For example to launch vManager with analysis view my_view_recent_runs, use:

```
vmanager -load_session <session> -analyze_all_runs -analysis_view  
my_view_recent_runs
```

-analysis_filter <filter>

Applies changes on the main table of the analysis view.

Usage example:

```
vmanager -cs -load_session <session_name> -analyze_runs -analysis_view <view name>  
-analysis_filter <filter>
```

Note: The format for specifying the filter is:

[attribute_name] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example: -filter name:~comp

-open_planner <vPlan_file>

Launches vManager with the Planner center with the specified vPlan_file; or empty, if none is specified.

For example, to invoke vManager with the Planner center with my_vplan.vplanx file loaded, use the following command:

```
vmanager -open_planner my_vplan.vplanx
```

-planner_view <view>

Launches vManager with the Planner center and sets the specified planner view (<view>).

For example, to invoke vManager with the Planner center with my_vplan.vplanx file and mk_view view loaded, use the following command:

```
vmanager -open_planner my_vplan.vplanx -planner_view mk_view
```

-planner_filter <filter>

Applies changes on the main table of the Planner center view.

Usage example:

```
vmanager -open_planner <vPlan_file> -planner_view <view> -filter <filter>
```

Note: The format for specifying the filter is:

[attribute_name] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example: -filter name:~comp

-open_tracking <tracking_config>

Launches vManager with the Project Tracking center with the specified tracking configuration.

For example, to invoke vManager with the Project Tracking center with `my_trconfig` configuration, use the following command:

```
vmanager -open_tracking my_trconfig
```

Generic Options

Following are the generic options of vManager:

<u>-32</u>	<u>-64</u>	<u>-32only</u>	<u>-64only</u>
<u>-3264</u>	<u>-6432</u>	<u>-quiet3264</u>	<u>-debug3264</u>
<u>-v3264</u>	<u>-help3264</u>		

Note: The above options are listed when you run the command `vmanager -help3264`.

-32

Launches the 32-bit version of the vManager executable. If 32-bit version is not available, then a warning is printed and an attempt is made to launch the 64-bit version of the tool.

-64

Launches the 64-bit version of the vManager executable. If 64-bit version is not available, then a warning is printed and an attempt is made to launch the 32-bit version of the tool.

Note: vManager by default is launched in 64-bit version. However, you can force the tool to run in 32-bit mode. For more details, see [Launching vManager in 32-bit Mode](#) on page 404.

-32only

Launches the 32-bit version of the vManager executable. If 32-bit version is not available, then an error is printed. No attempt is made to launch the 64-bit version of the tool.

-64only

Launches the 64-bit version of the vManager executable. If 64-bit version is not available, then an error is printed. No attempt is made to launch the 32-bit version of the tool.

-3264

Launches the 32-bit version of the vManager executable. If 32-bit version is not available, then an informational message is printed and an attempt is made to launch the 64-bit version of the tool.

-6432

Launches the 64-bit version of the vManager executable. If 64-bit version is not available, then an informational message is printed and an attempt is made to launch the 32-bit version of the tool.

-quiet3264

Suppresses printing of warning, error, informational messages of the -32, -32only, -3264, -64, -64only, and -6432 options; and launches vManager.

-debug3264

Prints the environment information (CDS_AUTO_64BIT, PATH, and LD_LIBRARY_PATH) updated by the cdnwrapper and then launches vManager.

-v3264

Prints the cdnWrapper version information.

-help3264

Displays a list of generic command options of vManager with a brief description about each option.

Launching vManager in 32-bit Mode

You can launch vManager in 32-bit mode in any of the following ways:

- Using the -32 related command-line options as any of the following:

```
vmanager -32  
vmanager -32only
```

- Setting the following environment variable:

- setenv VMANGER_AUTO_32BIT ALL

The advantage of setting the environment variable is that you do not have to include the -32 or -32only option on the command line every time when you have to launch the tool.

Note: If you set this environment variable and also include the 64-bit related options on the command line, then tool is launched in 64-bit mode.



If you set both CDS_AUTO_64BIT ALL and VMANGER_AUTO_32_BIT ALL, then 64-bit version of the tool will be launched because CDS_AUTO_64BIT is stronger than the VMANGER_AUTO_32_BIT.

Note: To confirm which version is currently set up, use the ncbits command. This command returns either 32 or 64 depending on the currently set up version.

```
% ncbits  
64
```

B.2 Batch-Mode Commands

B.2.1 General Commands

This section describes the following general commands (related to both vManager and IMC):

- help

- exit
- history
- alias
- unalias
- sourcemap
- cpu
- memory
- cdnshelp
- upgrade

B.2.1.1 help

To display help for a command, use:

```
help [command]
```

This command displays description of the specified command. If no command is specified, a list of all vManager commands is displayed.

B.2.1.2 exit

To exit from vManager, use:

```
exit
```

B.2.1.3 history

To display a list of previously executed commands in vManager, use:

```
history
```

B.2.1.4 alias

Lets you set an alias for a command.

Syntax

To set an alias for a command, use:

```
alias <alias_name> <command_name> [options]
```

where,

- <alias_name> is the name that you want to set as an alias for the command. If only the alias_name is specified, then the current definition of the specified alias is printed.
- <command_name> [options] specifies the command for which the alias is being set. You can also alias a command along with its options.

Note: In the absence of any options to the alias command, a list of all the aliases is displayed.

Examples

To set an alias, pref for the preferences command, use:

```
alias pref preferences
```

To set an alias rd for the command report -summary -metrics block, use:

```
alias rd report -summary -metrics block
```

To view a list of all aliases set, use:

```
alias
```

To view definition for a specific alias rd, use:

```
alias rd
```

B.2.1.5 unalias

Lets you remove an existing alias.

Syntax

```
unalias <alias_name>
```

where, <alias_name> is the name of the alias set earlier using the alias command.

Examples

To remove an alias named pref, use:

```
unalias pref
```

B.2.1.6 sourcemap

vManager by default, looks for the design source files in the directory from which vManager is launched. If the design instrumentation location is different from the one from which vManager is launched, map the source paths so that vManager can locate the design source files. The `sourcemap` command is used to map the source to new paths, remove existing mapped sources, and list the currently mapped paths. This command is useful if files are moved, or are referenced differently on different systems.

Syntax

```
sourcemap  
[-add <old> <new> |  
-remove <old> <new> |  
-clear |  
-print |  
-read <file>]
```

where

- `-add <old> <new>` maps the `<old>` path to the `<new>` path.
 - `<old>` indicates the path of the source files at the time of metric data generation. The `<old>` path must start at the root level.
 - `<new>` indicates the new location of source files. The `<new>` path must start at the root level.
- `-print` lists currently mapped paths. This option is same as the command without any options.
- `-clear` removes all existing mapped paths.
- `-remove <old> <new>` allows removing of specific mapped path.
- `-read <file>` reads and applies mapping specified in `<file>`. This option is useful when you want to apply multiple mappings using a single command. To use this option, specify all the mappings in a separate file and then pass it to the `sourcemap` command using the `-read` option.

Examples

To map the directory `/home1` to the directory `/mnt/emily/home1`, use:

```
sourcemap -add /home1 /mnt/emily/home1
```

For example, the following paths are mapped:

```
/home1 /mnt/sally/home1
```

```
/home2 /mnt/sally/home2  
/home3 /mnt/emily/home3
```

To remove the path /home2 /mnt/sally/home2, use:

```
sourcemap -remove /home2 /mnt/sally/home2
```

For example, specify the following mappings in a separate file maps.txt:

```
/home1 /mnt/sally/home1  
/home2 /mnt/sally/home2  
/home3 /mnt/emily/home3
```

To apply all these mappings using a single command, use:

```
sourcemap -read maps.txt
```

B.2.1.7 cpu

To display CPU performance information, use:

```
cpu
```

B.2.1.8 memory

To display memory performance information, use:

```
memory
```

B.2.1.9 cdnshelp

To start cdnshelp application and to show the specified guide, use:

```
cdnshelp [-user_guide | -quick_ref | -kpns | -whats_new]
```

where

- -user_guide opens the *vMananger User* guide.
- -quick_ref opens the *vMananger Quick Reference* guide.
- -kpns opens the *vMananger KPNS* (Know Problems and Solutions) guide.
- -whats_new opens the *vMananger What's new* guide.

Note: If none of the options is specified with the cdnshelp command, by default, the *vMananger User* guide is opened.

B.2.1.10 upgrade

The `upgrade` command is used to convert/upgrade the *Regression Center Launcher* client to a *Full* client.

If you have launched vManager in *Regression Center Launcher* client mode (using the `-regr` or `-regrcenter` option) and now want to upgrade to the *Full* client mode, use the following command:

```
upgrade
```

Note: Not all CLI commands are available in *Regression Center Launcher* client mode. After you upgrade to *Full* client mode, all the CLI commands are available for use.



The *Regression Center Launcher* client uses the same license as IMC, which is the `Affirma_sim_analysis_env` license. However, the *Full* client uses the `vmanager_client` license. When you use the `upgrade` command, the tool checks out for the `vmanager_client` license.

- ❑ If the license is not available, an error is reported and the tool continues to run in *Regression Center Launcher* client mode.
- ❑ If the license is available, then the `Affirma_sim_analysis_env` license is released and all the functionality of *Full* client is available.

B.2.2 Metrics-related Commands

This section describes the metrics-related commands (applies to both vManager and IMC):

- [report](#)
- [exclude](#)
- [unexclude](#)
- [save](#)
- [convert_icf](#)

B.2.2.1 report

The `report` command allows you to generate reports for data analysis in interactive command-line mode. Using the `report` command, you can generate a list, summary, detailed, or HTML reports.

Syntax

```
report [-detail | -summary | -list]
[<format>]
[-metrics <metrics_type>]
[-out <name>]
[-overwrite]
[-covered | -uncovered | -both | -all | -excludes | -unr]
[-append on | off]
[-showempty on | off]
[-source on | off]
[-cumulative on | off]
[-local on | off]
[-grading <average> | <covered> | <both>]
[-cross <expand> | <aggregate>]
[-kind <expand> |
  <compact> [-cubeExpand on|off] |
  <abstract> |
  <aggregate> [[-atleast <num> | -upto <num>] [-cubesize <min>]
                [-cubestars <min>] [-excludeNA] [-illegalIgnore]]]
[-assertionStatus]
[-allAssertionCounters]
[-type |-inst]
<list>

metrics_type ::= [overall] [all] [code] [fsm] [functional] [block] [expression]
[toggle] [state] [transition] [arc] [covergroup] [assertion]
format ::= [-text | -html]
```

where

- `-summary | -detail | -list` specifies the type of report that must be generated. If not specified, `-summary` is assumed.
- `-metrics <metrics_type>` specifies the metrics types for which the report must be generated. The `<metrics_type>` can be any, or a combination of the following:
 - `code` for printing a detailed block, expression, and toggle coverage report
 - `fsm` for printing a detailed state, transition, and arc coverage report
 - `functional` for printing a detailed assertion and covergroup coverage report
 - `block` for printing a detailed block coverage report
 - `expression` for printing a detailed expression coverage report
 - `toggle` for printing a detailed toggle coverage report
 - `state` for printing a detailed state coverage report
 - `transition` for printing a detailed transition coverage report
 - `arc` for printing a detailed arc coverage report
 - `assertion` for printing a detailed assertion coverage report

- ❑ `covergroup` for printing a detailed covergroup coverage report
- ❑ `overall` for printing a detailed report for all metrics
- ❑ `all` for printing a detailed report for all metrics

Note: You can specify more than one metrics type by separating the metric types with a colon (:). If `<metrics_type>` is not specified, a detailed report for all metrics is generated.

- `-append on|off` specifies if the output file must be appended or not. By default, the output file is overwritten. To append the output file instead of overwriting it, use the `-append on` option. This option is used only with the `-out` option and is valid only for `text (-text)` reports.
- `-inst | -type` specifies if report should be generated based on instance or type. If not specified, `-inst` is assumed.
- `<list>` specifies the names of types or instances for which the report will be generated. The default value is `* . . .`, which matches all the top-level instances and their children. It can include wildcard characters `*` and `?`. Use `. . .` to specify all descendants of an instance. Instance names can be specified as hierarchical names, for example, `top.inst1.inst2`. When specifying instance names that include characters like `[]` or `()`, use a backslash `\` to escape each such character. For example, to specify instance name `sys.ahb_evc.active_masters[1]monitor`, use:

```
report -list -inst sys.ahb_evc.active_masters\[1\]monitor
```

Note: The path can be prefixed with "*Verification\ Metrics*", "*Verification\ Metrics/Instances*", "*Verification\ Metrics/Types*", "*Instances*", or "*Types*". Prefix is not mandatory. However, if the *Instances* or *Types* prefix is specified, the performance of the command can be improved. For example, `report a.b.c` can be specified as:

```
report Verification\ Metrics/Instances/a.b.c
```

or

```
report Instances/a.b.c
```

Note: Wild cards (`*`, `?`, and `...`) are not supported in the prefix *Verification\ Metrics/Types* or *Verification\ Metrics/Instances*.

- `-all | -covered | -uncovered | -both | -excludes | -unr` specifies whether to generate reports for all, covered, uncovered, both covered and uncovered, excluded, or unreachable items.

Note: If none of the options (`-all`, `-covered`, `-uncovered`, `-both`, `-unr`, or `-excludes`) is specified, `-uncovered` is assumed. However, in the case of covergroup detailed report in an HTML format, if none of these options are specified, `-both` is assumed.

- `-cumulative on|off` enables or disables printing of cumulative grades in the ASCII summary report. By default, the summary report prints only the local grades and not the cumulative grades in the report. To report the cumulative grades along with the local grades, set the `-cumulative` option to `on`. When you use the `-cumulative on` option, cumulative average grade as well as the cumulative covered grade is printed for the specified metrics type. This option is invalid and is ignored in the case of type-based reports. In addition, this option is valid only for ASCII summary reports (`-summary`).
- `-local on|off` disables or enables reporting of local grades when cumulative grades are printed. By default, local grades are printed along with cumulative grades. To disable reporting of local grades, set the `-local` option to `off`. This option is used only with the `-cumulative` option and is valid only for the summary report (`-summary`).
- `-showempty on|off` enables or disables reporting of items with no self coverage data (0/0). By default, such items are not printed. To enable reporting of items with no self coverage, set the `-showempty` option to `on`. This option is used only with the `-summary` option.

Note: When the `-showempty` option is set to `off`, the entities with no self coverage are not reported unless they have a descendant that has coverage. Empty entities are reported if they have non-empty descendants.

- `format` specifies the format of the report. It can be any of the following:
 - `-text` to generate report in a text format
 - `-html` to generate summary or detail report in an HTML format.
- Note:** If not specified, `-text` is assumed.
- `-out <name>` specifies the location to which the report must be redirected. By default, reports are written to standard output unless redirected to a file.
 - If the `format` is specified as `-text`, the text reports are redirected to `<name>` instead of standard output. In this case, `<name>` is the name of the file to which the report is saved.
 - If the `format` is specified as `-html`, the HTML reports are redirected to `<name>` instead of default HTML output directory `html_<tstamp>`. If `<name>` includes just the name of the output directory, then the command creates the output directory named `<name>` in the current working directory, and stores the HTML report files along with the top-level summary page in `<name>`. For example, if `<name>` is specified as:

```
report -detail -html -all -type * -out day1
```

then directory named `day1` is created in the current working directory to store HTML report files.

Note: If <name> already exists, the output directory is not overwritten. You must use the `-overwrite` option to overwrite an existing output directory.

If <name> includes the complete path, then the command creates a directory by the name mentioned at the lowest level in the path, and stores the HTML report files at that location. For example, if <name> is specified as:

```
report -detail -html -all -type * -out data/myreports/day1
```

then directory named day1 is created under data/myreports to store HTML report files. In this case, the path (data/myreports) must exist. The directory day1 must not exist, as it will be created and not overwritten unless the `-overwrite` option is used.

Note: If the `-out` option is not specified with the `-html` option, by default, a directory named `html_<timestmp>` is created in the current working directory, and coverage HTML reports along with the top-level summary page (`index.html`) are stored in the `html_<timestmp>` directory.

- `-overwrite` enables overwriting of the existing HTML output directory. By default, HTML output directory is not overwritten and an error is reported.

Note: The `-overwrite` option is ignored in the case of text reports.

- `-source off|on` enables or disables reporting of the source text and line numbers from the source code. By default, this information is reported. To disable printing of source text and line numbers, set the `-source` option to `off`. This option is used only with the `-detail` option.

Note: Currently, the batch mode cannot read the compressed files, and therefore, the *Source Code* column of the report of entities from the compressed files will be blank (even if the `-source` option is turned on).

- `-grading <average> | <covered> | <both>` enables printing of cumulative average grade, cumulative covered grade, or both (cumulative average as well as cumulative covered) grades in the ASCII summary report or HTML report. By default, both cumulative covered as well as cumulative average grades are printed in the report.
 - `-grading average` enables printing of only the cumulative average grade.
 - `-grading covered` enables printing of only the cumulative covered grade.
 - `-grading both` enables printing of both cumulative average grade as well as the cumulative covered grade. This is the default.

Note: In the absence of the `-grading` option, both is assumed. In addition, this option can be used only with the `-cumulative` option.

- `-cross <expand> | <aggregate>` specifies the format in which the automatically generated cross bins must be printed in the HTML report. The format can be specified as any of the following:
 - `expand`—to print the cross tuple information in a list format. This is the default behavior.
 - `aggregate`—to print the cross tuple information in a tabular format after grouping similar bins.
- Note:** The `-cross` option is supported only with HTML reports.
- `-kind` specifies the format in which the covergroup bin information must be printed. The format can be specified as any of the following:
 - `expand`—to print the bin information as a list without grouping similar bins. This is the default behavior. This option is supported only with HTML reports.
 - `compact`—to print the bin information after grouping similar bins (cubes). For each cube, a separate detailed page is created which includes information about the bins included in that cube. This option is supported only with HTML reports.
 - `-cubeExpand on|off` enables or disables generation of separate detailed page for each cube. By default, the value of this option is set to `on`, which enables generation of detailed information page for each cube. To disable generation of detailed information page, set the `-cubeExpand` option to `off`. Setting this option as `off` is useful when the information on the detailed page is not of much relevance and also in situations where you want to save the disk space.
 - `abstract`—to disable printing of bin information in the ASCII detailed covergroup reports. By default, bin information is printed in the detailed ASCII report. This option is supported only with ASCII detailed reports.
 - `aggregate`—to print the cross tuple information in a tabular format after grouping similar bins. While generating aggregated results, you can specify following options:
 - `-atleast <num> | -upto <num>` — to set the minimum and maximum number of samples required for a bin to be considered covered.
 - `-cubesize <min>` — to set the minimum size of reported cubes, (measured in % of physical space size).
 - `-cubestars <min>` — to set the minimum number of stars (*) in the reported cubes. Stars are used in aggregation results automatically when it makes the results more readable.

- -excludeNA — If specified, NA bins are not included in reported holes/covered cubes.
 - -illegalIgnore — to include aggregation also for illegal and ignore space.
- -assertionStatus enables reporting of assertion status information in the report.
- In the case of summary report, an additional column, Assertion Status is printed in the report, which shows the assertion status information for each instance or type as:
$$\text{Percentage\%} \text{ (Passed_assertions / Failed_assertions / Other_assertions)}$$
where
 - Percentage% is the total percentage of passed assertions out of all assertions, which is calculated as $\text{<Passed_assertions>} / \text{<Total_assertions>} * 100$
 - Passed_assertions is the total number of assertions that finished without failure at least once, and never failed.
 - Failed_assertions is the total number of assertions that finished with failure at least once.
 - Other_assertions is the total number of assertions that never finished.

Note: In the case of cumulative coverage, column is named as Assertion* Status.

- In the case of detailed report, an additional column, Status is printed in the report, which shows the status of each assertion in the specified instance or type. The status can be any of the following:
 - Passed if the assertion finished without failure at least once, and never failed.
 - Failed if the assertion finished with failure at least once.
 - Other if the assertion never finished.

Note: If -allAssertionCounters option is also used with the -assertionStatus option, then the value Other in the Status field is further classified as any of the following:

- Vacuous if the assertion was vacuous pass at least once but never finished.
- Disabled if the assertion transitioned to the disabled state at least once but never finished.
- Attempted if the assertion was attempted at least once but never finished.

- Not Attempted if the assertion was not attempted all (that is all the counters are equal to zero).

Note: The `-assertionStatus` option is applicable only for assertion coverage reports.

- `-allAssertionCounters` enables reporting of Vacuous, Attempt, and Disabled counters in addition to the default Finished and Failed counters in the detailed report.

Note: The `-allAssertionCounters` option is applicable only if the `-abvrecordvacuous` switch is used at the time of simulation run. In case the `-abvrecordvacuous` switch is not used at simulation, then the Vacuous, Attempt, and Disabled columns will show n/a. For more details on the `-abvrecordvacuous` switch, see the *Verilog Compilation Command-Line Options* user guide.

Examples -- List Reports

To generate a list of instances in the loaded run, use:

```
report -list -inst
```

or

```
report -list
```

To list the types in the loaded run, use:

```
report -list -type
```

To list only the instances in which covergroups are scored, use:

```
report -list -metrics covergroup
```

Examples -- Summary Reports

To generate an instance-based block summary report with cumulative and local grades, use:

```
report -summary -metrics block -cumulative on
```

To print only block cumulative average grade in the above report, use:

```
report -summary -metrics block -cumulative on -local off -grading average
```

To generate an type-based functional coverage summary report, use:

```
report -summary -metrics functional -type
```

Examples -- Detailed Reports

To print a block coverage report for the instance DIGIT_REG_INST, use:

```
report -detail -metrics block -both dtmf_recv_core.DIGIT_REG_INST
```

To print a detail report, displaying expression coverage data for the instance

dtmf_recv_core.TDSP_CORE_INST.DECODE_INST, use:

```
report -detail -metrics expression -both  
dtmf_recv_core.TDSP_CORE_INST.DECODE_INST
```

To print a detail report displaying toggle coverage data for the instance ARB_INST, use:

```
report -detail -metrics toggle -both dtmf_recv_core.AR_B_INST
```

To report detailed state coverage data for the module spi, as shown below, use:

```
report -detail -metrics state -both -type spi
```

To report detailed arc coverage data for the module spi, use:

```
report -detail -metrics arc -both -type spi
```

To generate a detailed report for assertion coverage items for instance DMA_INST in module dtmf_recv_core, use:

```
report -detail -metrics assertion -both dtmf_recv_core.DMA_INST
```

To display a detail report for covergroup coverage for type tdsp_core, use:

```
report -detail -metrics covergroup -both -type tdsp_core
```

Examples -- HTML Reports

To generate HTML report of all metrics types for all of the instances in the design, use:

```
report -detail -html -all -out detail_rep -overwrite -inst *...
```

To generate HTML report in which similar cross bins are grouped, use the -cross aggregate option, use:

```
report -detail -all -html -out detail_rep2 -cross aggregate -inst *...
```

To generate HTML report in which similar bins are grouped together to identify patterns, use:

```
report -detail -html -kind compact -out html_compact
```

B.2.2.2 exclude

The exclude command enables you to exclude metrics items or any coverage item of specified type or instance.

Syntax

```
exclude
{[-vplan_node <node>] | [-hierarchy <hierarchy>]}
{-inst <inst_name> | -type <type_name>}
[-metrics <metrics_type>] | <coverage_object_specification>
[-smart]
[-unr]
[-comment <comment>]
[-reviewer <reviewer_name>]
```

where

```
<hierarchy> ::= 
[<perspective_name>] / [<section_name>]* / [<port_name>] / [<instance>|<type>] /
[<entity_name>]

<entity_name> ::= <covergroup_name> | [<covergroup_name>].<coverpoint_name> |
<fsm_name> | <assertion_name>

<node> ::= [<perspective_name>] / [<section_name>]* / [<port_name>]

<metrics_type> ::= [all] [block] [expression] [toggle] [fsm] [assertion]
[covergroup] [state] [transition] [arc] [code] [functional]

<coverage_object_specification> ::= -assertion <assertion_name> |
-covergroup <covergroup_name> |
-coveritem <covergroup_name>.<coverpoint_name> |
-coverbin <covergroup_name>.<coverpoint_name>.<coverbin_name> |
-fsm <fsm_name> |
-state <fsm_name>.<state_name> |
-transition <fsm_name>.<from_state>/|.<to_state> |
-arc <fsm_name>.<from_state>.<to_state>.<arc_index>
-toggle <signal_full_name> [[<bit_specification>] [.rise | .fall]] |
-block <index_specification> |
-expression <index_specification>

<index_specification> ::= <index> | <range>
<range> ::= <index>-<index>
<bit_specification> ::= [<index>] | [<range>]
```

In the above syntax,

- `-hierarchy <hierarchy>` specifies the hierarchy in vPlan tree to which exclusion must apply. The `<hierarchy>` cannot include the wildcard characters *, . . . , and ?.

Note: You can use *Unmapped* perspective in the path. If the *Unmapped* perspective does not exist, then it is created automatically. For example,

```
exclude -hierarchy Unmapped/FIFO/code -inst
RAM_256x16_TEST_INST/RAM_256x16_INST -metrics block
```

- `-vplan_node <node>` specifies the path to the port in vPlan tree to which exclusion must apply. The `<node>` can include the wildcard characters *, . . . , and ?.

Note: You can use *Unmapped* perspective in the path. For example,

```
exclude -vplan_node Unmapped/FIFO/code -inst
RAM_256x16_TEST_INST/RAM_256x16_INST -metrics block
```

If the *Unmapped* perspective does not exist, then it is created automatically. However, if the wildcard characters are used instead of the perspective name, then the *Unmapped* perspective is not created automatically. For example, following command uses wildcard character and therefore, *Unmapped* perspective will not be created:

```
exclude -vplan_node Unmapped/FIFO/code -inst  
RAM_256x16_TEST_INST/RAM_256x16_INST -metrics block
```

- `-inst <instance>` specifies the instance to which exclusion must apply. The `<instance>` can include the wildcard characters `*`, `...`, and `?`.

Note: Special characters like `[]` in the instance names must be escaped with `\`. Else, you can use `{}` to avoid the use of `\`. For example, if the instance name is `xbus0.masters[0]`, then specify it as:

```
-inst xbus0.masters\[0\]
```

or

```
-inst {xbus0.masters[0]}
```

- `-type <type>` specifies the type to which exclusion must apply. The `<type>` can include wildcard characters `*` and `?`.

Note: When you exclude a type, all instances within the type get excluded. However, when you exclude all instances of a type, the type is not excluded automatically.

- `-metrics <metrics_type>` specifies the type of metrics to be excluded. This option should be used when you want to exclude coverage of all the items of the specified type or instance. The `<metrics_type>` can be any of the following:

`all` for all metric types

`code` for block, expression, and toggle coverage

`fsm` for state, transition, and arc coverage

`functional` for both assertion and covergroup coverage

`block` for block coverage

`expression` for expression coverage

`toggle` for toggle coverage

`state` for state coverage

`transition` for transition coverage

`arc` for arc coverage

`assertion` for PSL/SVA-based assertion coverage

`covergroup` for SystemVerilog-based covergroup coverage

Note: You can specify more than one metric type by separating the metric types with a colon (`:`). For example, to exclude block and expression, use `block:expression`.

- `<coverage_object_specification>` specifies the coverage objects to be excluded. This option must be used when you want to exclude individual coverage items

of the specified instance or type. The `<coverage_object_specification>` can be any of the following:

- ❑ `-assertion <assertion_name>` to specify the assertions that must be excluded. It can include wildcard characters * and ?.
- ❑ `-covergroup <covergroup_name>` to specify the covergroups that must be excluded. It can include wildcard characters * and ?.
- ❑ `-coveritem <covergroup_name>. <coverpoint_name>` to specify the coverpoints that must be excluded. It can include wildcard characters * and ?.
- ❑ `-coverbin <covergroup_name>. <coverpoint_name>. <coverbin_name>` to specify the coverpoint bins that must be excluded. It can include wildcard characters * and ?.
- ❑ `-fsm <fsm_name>` to specify the state machines that must be excluded. It can include wildcard characters * and ?.
- ❑ `-state <fsm_name>. <state_name>` to specify the states that must be excluded. It can include wildcard characters * and ?.
- ❑ `-transition <fsm_name>. <from_state>/ | . <to_state>` to specify the transitions that must be excluded. It can include wildcard characters * and ?.
- ❑ `-arc <fsm_name>. <from_state>/ | . <to_state>. <arc_index>` to specify the arcs that must be excluded. The `<arc_index>` cannot include wildcard character ?, but it can include wildcard character *.

Note: The `<arc_index>` specifies the arcs to be excluded. The `<arc_index>` starts from number 0. You can specify the `<arc_index>` as:

- `<index>` to specify the arc number for exclusion. You can specify multiple arcs by separating them by a space character.
- `<range>` to specify a range of arcs for exclusion. The right index of the range must be greater than the left index.
- ❑ `-toggle <signal_full_name> [[<bit_specification>]] [.rise | .fall]]` to specify the signals that must be excluded. The `<signal_full_name>` can include wildcard characters * and ?. The `<bit_specification>` cannot include wildcard character ?, but it can include wildcard character *. Optional keywords `rise` and `fall` indicate if rise transition must be excluded or the fall transition must be excluded.

Note: The `<bit_specification>` should include positive integer values. The `<bit_specification>` specifies the signal bits for exclusion. Signal bits can be specified as any of the following:

- Single bits using <index>
 - Range of bits using <range>
 - -block <index_specification> to specify the blocks that must be excluded. The <index_specification> cannot include wildcard character ?, but it can include wildcard character *.
 - -expression <index_specification> to specify the expressions that must be excluded. The <index_specification> cannot include wildcard character ?, but it can include wildcard character *.
- Note:** The <index_specification> should include positive integer values. The <index_specification> specifies the report items to be excluded. It can be specified as:
- <index> to specify the block number or expression number for exclusion. You can specify multiple blocks/expressions by separating them by a space character.
 - <range> to specify a range of blocks or expressions for exclusion. The right index of the range must be greater than the left index.
- -smart makes the exclusion action smart, that is, any entity connected to the excluded entity will be excluded implicitly.
 - -unr is used to exclude only the items that were marked UNR by IEV. For details on the UNR flow and how items are marked UNR by IEV, see the *IEV User Guide*.
 - -comment <comment> specifies the exclusion comment for the specified instance, type, or coverage item. The exclusion comment <comment> must be enclosed within quotes ". For example, to add the comment as "This block cannot be covered" for block 8 of instance dtmf_recv_core.SPI_INST, use the following command:

```
exclude -inst dtmf_recv_core.SPI_INST -block 8 -comment "This block cannot be covered"
```

Note: Comment cannot be more than 500 characters.

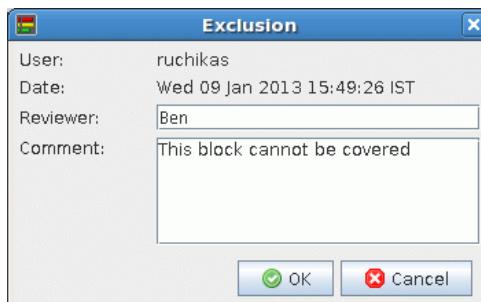
- -reviewer <reviewer_name> specifies the exclusion reviewer of the specified instance, type, or coverage item. For example, to specify the exclusion reviewer as Ben for block 8 of instance dtmf_recv_core.SPI_INST, use the following command:

```
exclude -inst dtmf_recv_core.SPI_INST -block 8 -comment "This block cannot be covered" -reviewer Ben
```

Note: If no reviewer is specified at the time of exclusion, the default value unknown is considered as the reviewer. In addition, the reviewer name cannot be more than 24 characters.

Note: The exclusion comments and reviewer details added and saved in vManager command-line interactive mode are available in the vManager GUI and vice versa. For example, if the above exclusions are saved in a refinement file, you can load them in GUI and also edit them. [Figure B-6](#) on page 422 shows the *Exclusion* dialog box in vManager GUI, which shows the details added from vManager command-line.

Figure B-6 Editing Exclusion Comments in GUI



Examples -- All Items

To exclude all coverage objects in instance `dtmpf_recv_core.SPI_INST`, use:

```
exclude -inst dtmpf_recv_core.SPI_INST
```

To exclude all coverage objects in instance hierarchy rooted at `test.DUT.I1` including `test.DUT.I1`, use:

```
exclude -inst test.DUT.I1...
```

To exclude all coverage objects in type `dma`, use:

```
exclude -type dma
```

To exclude all blocks, expressions, and signals of instance `dtmpf_recv_core.AR_B_INST`, use:

```
exclude -inst dtmpf_recv_core.AR_B_INST -metrics code
```

To exclude all assertions and covergroups of instance `dtmpf_recv_core.AR_B_INST`, use:

```
exclude -inst dtmpf_recv_core.AR_B_INST -metrics functional
```

Examples -- Block Coverage

To exclude all of the blocks of instance `dtmpf_recv_core.SPI_INST`, use:

```
exclude -inst dtmpf_recv_core.SPI_INST -metrics block
```

To exclude all of the blocks of type `spi`, use:

```
exclude -type spi -metrics block
```

To exclude all of the blocks of instance RAM_256x16_TEST_INST/RAM_256x16_INST under hierarchy FIFO/code, use:

```
exclude -hierarchy FIFO/code -inst RAM_256x16_TEST_INST/RAM_256x16_INST -metrics block
```

Note: You can also use -vplan_node instead of -hierarchy whenever the corresponding path includes only vplan nodes (perspectives, sections, and ports). In the above command, you can also use -vplan_node instead of -hierarchy.

To exclude blocks 3 and 5 of instance dtmf_recv_core.TEST_CONTROL_INST, use:

```
exclude -inst dtmf_recv_core.TEST_CONTROL_INST -block 3 5
```

To exclude blocks 7 to 9 of instance dtmf_recv_core.TEST_CONTROL_INST, use:

```
exclude -inst dtmf_recv_core.TEST_CONTROL_INST -block 7-9
```

Examples -- Expression Coverage

To exclude all of the expressions in instance dtmf_recv_core.DECODE_INST, use:

```
exclude -inst dtmf_recv_core.DECODE_INST -metrics expression
```

To exclude all of the expressions in type dma, use:

```
exclude -type dma -metrics expression
```

To exclude expression 1.2, use:

```
exclude -inst dtmf_recv_core.TDSP_CORE_INST.ALU_32_INST -expression 1.2
```

To exclude expressions 2.2.2 and 2.2.3, use:

```
exclude -inst dtmf_recv_core.TDSP_CORE_INST.ALU_32_INST -expression 2.2.2-3
```

Examples -- Toggle Coverage

To exclude all of the signals of instance dtmf_recv_core.AR_B_INST, use:

```
exclude -inst dtmf_recv_core.AR_B_INST -metrics toggle
```

To exclude the signal bits present_state[0] and present_state[1], use:

```
exclude -inst dtmf_recv_core.AR_B_INST -toggle {present_state[0-1]}
```

To exclude only the rise transition of signal bit present_state[2], use:

```
exclude -inst dtmf_recv_core.AR_B_INST -toggle present_state\[2.rise\]
```

To exclude only the rise transition of all the signal bits of next_state, use:

```
exclude -inst dtmf_recv_core.AR_B_INST -toggle next_state\[*.rise\]
```

To exclude only the fall transition of signal `reset`, use:

```
exclude -inst dtmf_recv_core.AR_B_INST -toggle reset\[0.fall\]
```

Examples -- FSM Coverage

To exclude all state machines of instance `dtmf_recv_core.SPI_INST`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -metrics fsm
```

To exclude `FSM present_state` from coverage analysis, use:

```
exclude -inst dtmf_recv_core.SPI_INST -fsm present_state
```

To exclude all states of all of the FSMs in instance `dtmf_recv_core.SPI_INST`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -metrics state
```

To exclude the state `State_1`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -state present_state.State_1
```

You can also exclude multiple states using a single `exclude` command. For example, to exclude `State_0` and `State_3`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -state present_state.State_0  
present_state.State_3
```

To exclude all transitions of all of the FSMs in instance `dtmf_recv_core.SPI_INST`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -metrics transition
```

To exclude the transition `State_1 -> State_0`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -transition present_state.State_1.State_0
```

To exclude all arcs of all of the FSMs in instance `dtmf_recv_core.SPI_INST`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -metrics arc
```

To exclude the first arc condition of `State_2 -> State_3`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -arc present_state.State_2.State_3.0
```

Examples -- Assertion Coverage

To exclude all of the assertions in the instance `dtmf_recv_core.SPI_INST`, use:

```
exclude -inst dtmf_recv_core.SPI_INST -metrics assertion
```

To exclude assertion `spi_fs_start` from coverage results, use:

```
exclude -inst dtmf_recv_core.SPI_INST -assertion spi_fs_start
```

Examples -- Covergroup Coverage

To exclude all covergroups of instance dtmf_recv_core.TDSP_CORE_INST, use:

```
exclude -inst dtmf_recv_core.TDSP_CORE_INST -metrics covergroup
```

To exclude covergroup cg of instance dtmf_recv_core.TDSP_CORE_INST from coverage analysis, use:

```
exclude -inst dtmf_recv_core.TDSP_CORE_INST -covergroup cg
```

To exclude all of the covergroups under hierarchy FIFO/code, use:

```
exclude -hierarchy FIFO/code -metrics covergroup
```

Note: In the above command, if code is a port, then you can also use -vplan_node instead of -hierarchy.

To exclude coverpoints B and B2 from coverage results, use:

```
exclude -inst dtmf_recv_core.TDSP_CORE_INST -coveritem cginst1.B cginst1.B2
```

To exclude coverbin vector_low[20] of coverpoint A2, use:

```
exclude -inst dtmf_recv_core.TDSP_CORE_INST -coverbin cg.A2.vector_low\[20\]
```

Examples -- Unreachable Items

To exclude the unreachable block 4 of instance top.c1 from coverage results, use:

```
exclude -inst top.c1 -block 4 -unr
```

To exclude all of the unreachable blocks of instance top.c1 from coverage results, use:

```
exclude -inst top.c1 -metrics block -unr
```

To exclude all of the unreachable expressions of instance top.c1 from coverage results, use:

```
exclude -inst top.c1 -metrics expression -unr
```

To exclude UNR expression 1.1.2 of instance top.c1 from coverage results, use:

```
exclude -inst top.c1 -expression 1.1.2 -unr
```

Examples -- vPlan Objects

To exclude all coverage objects under FIFO/code, use:

```
exclude -hierarchy FIFO/code
```

To exclude all coverage objects in instance RAM_256x16_TEST_INST under FIFO/code, use:

```
exclude -hierarchy FIFO/code/RAM_256x16_TEST_INST
```

or

```
exclude -hierarchy FIFO/code -inst RAM_256x16_TEST_INST
```

To exclude instance RAM_256x16_TEST_INST/RAM_256x16_INST under FIFO/code, use:

```
exclude -vplan_node FIFO/code -inst RAM_256x16_TEST_INST/RAM_256x16_INST
```

To exclude coverpoints tx_fifo_empty from coverage results under the perspective new_plan27_3, use:

```
exclude -hierarchy tx_fifo_empty/FIFO/code -coveritem register_e/tx_fifo_empty
```

or

```
exclude -hierarchy tx_fifo_empty/FIFO/code/register_e/tx_fifo_empty
```

Note: You must load the session and vPlan before using the -hierarchy and -vplan_node options.

B.2.2.3 unexclude

Un-excludes the excluded items.

Syntax

```
unexclude
{[-vplan_node <node>] | [-hierarchy <hierarchy>]}
{-inst <inst_name> | -type <type_name>}
[-metrics <metrics_type>] | <coverage_object_specification>
[-unr]
[-comment <comment>]
[-reviewer <reviewer_name>]
```

where

```
<hierarchy> ::= 
[<perspective_name>] / [<section_name>] * / [<port_name>] / [<instance>|<type>] /
[<entity_name>]

<entity_name> ::= <covergroup_name> | [<covergroup_name>].<coverpoint_name> |
<fsm_name> | <assertion_name>

<node> ::= [<perspective_name>] / [<section_name>] * / [<port_name>]

<metrics_type> ::= [all] [block] [expression] [toggle] [fsm] [assertion]
[covergroup] [state] [transition] [arc] [code] [functional]

<coverage_object_specification> ::= -assertion <assertion_name> |
-covergroup <covergroup_name> |
-coveritem <covergroup_name>.<coverpoint_name> |
-coverbin <covergroup_name>.<coverpoint_name>.<coverbin_name> |
-fsm <fsm_name> |
-state <fsm_name>.<state_name> |
-transition <fsm_name>.<from_state>/|.<to_state> |
-arc <fsm_name>.<from_state>.<to_state>.<arc_index>
```

```
-toggle <signal_full_name> [[<bit_specification>] [.rise | .fall]]|  
-block <index_specification> |  
-expression <index_specification>  
  
<index_specification> ::= <index> | <range>  
<range> ::= <index>-<index>  
<bit_specification> ::= [<index>] | [<range>]
```

See the [exclude](#) command for details on syntax description. This command reverses the effect of a previous exclude command.

B.2.2.4 save

Allows you to save the exclusions (metrics and vplan) applied to a file. Later, when required, you can load the saved refinements file. This saves you the effort of making exclusions again.

Syntax

```
save [-refinement <file>]  
[-vplan_refinement <file>]
```

where

- `-refinement <file>` specifies the file (`.vRefine`) in which the metrics refinement information will be saved. This refinement file can be reloaded later, when required, to quickly apply the refinements.
- `-vplan_refinement <file>` specifies the file (`.vpRefine`) in which the vPlan refinement information will be saved. This vPlan refinement file can be reloaded later, when required, to quickly apply the refinements.

Using this command, you can also save refinements from multiple refinement files to a single file.

Examples

To save the metrics refinements to a file named `my_exclude.vRefine`, use:

```
save -refinement my_exclude.vRefine
```

To save the vPlan refinements to a file named `my_vplanexclude.vpRefine`, use:

```
save -vplan_refinement my_vplanexclude.vpRefine
```

Consider another example where refinements from multiple files are saved to a single file.

Assume that you applied and saved refinements in three files:

```
my_exclude1.vRefine  
my_exclude2.vRefine  
my_exclude3.vRefine
```

To apply refinements from all the three files, you need to load all the three files. You can use following commands:

```
load -refinement my_exclude1.vRefine  
load -refinement my_exclude2.vRefine  
load -refinement my_exclude3.vRefine
```

After loading all the files, you can consider saving the refinements to a single file so that you do not need to load all three files again in the next invocation of vManager. To save the refinements from all of these files, to `my_allexclude.vRefine`, use:

```
save -refinement my_allexclude.vRefine
```

With the above command, refinements from all the three files are saved to a single file `my_allexclude.vRefine`. In the next invocation of vManager, you can load only the `my_allexclude.vRefine` file.

Note: The above flow of saving refinements from multiple files to a single file is applicable to vPlan refinement files also. In the case of vPlan refinement files, you will have to use the `-vplan_refinement` option instead of the `-refinement` option.

B.2.2.5 convert_icf

The `convert_icf` command allows you to quickly convert an ICCR Configuration File (ICF) into an exclude file.

Note: An ICCR Configuration File (ICF) cannot be used directly in vManager. This is because in ICCR, marking (refinement) was supported through `mark` and `undo_mark` commands. However, in vManager, refinement is supported through `exclude` and `unexclude` commands. Therefore, to apply marks from an ICF, you must convert it to an exclude file.

The `convert_icf` command converts the `mark/undo_mark` commands written in an ICF to equivalent `exclude/unexclude` commands and saves them in a separate file.

Note: Coverage data must be loaded before executing the `convert_icf` command.

Syntax

```
convert_icf <icf_filename> -out <converted_exclude_filename> [-ies92]  
[-filter <mark> | <undo_mark>]
```

where

- <icf_filename> is the name of the ICF that must be converted.
- <converted_exclude_filename> is the name of the file that will be generated after conversion.
- -ies92 is required to make the ICF saved with IES9.2 release version compatible with the IES10.2 release version. An ICF saved with IES9.2 release version is not compatible with the IES10.2 release version due to change in expression indexing. If the ICF includes only block coverage marks, then -ies92 option can be ignored. However, it is recommended that ICFs saved with IES9.2 release version are converted with the -ies92 option.

For more details on ICF and marking in ICCR, see the *ICC User Guide*.

- -filter <mark> | <undo_mark> filters/ignores all the mark / undo_mark commands specified in the <icf_filename> during conversion.

When converting an ICF to an exclude file, remember that:

- The exclude file generated with the convert_icf command cannot be loaded using the load -refinement command. To apply the exclusions from the file generated after conversion, use the source command at the vManager prompt. For example, a file named my_excl.bat is generated with the convert_icf command. To apply exclusions from this file, use source my_excl.bat command at the vManager prompt.
- Currently, during conversion vManager treats both items marked as IGN and COV (using the mark command of ICCR) as **excluded**. For details on marking items as IGN and COV in ICCR, see the *ICC User Guide*.

Examples

To convert an ICF file named mymarks.icf to an exclude file (name it as myexclude.bat), use the following command:

```
convert_icf mymarks.icf -out myexclude.bat
```

With the above command, the commands written in the mymarks.icf file are converted to equivalent exclude and unexclude commands and the converted commands are saved in the file myexclude.bat.

To apply the exclusions from the file myexclude.bat, use the following command:

```
source myexclude.bat
```

The above command will apply the exclusion commands written in file myexclude.bat to the loaded run.

B.2.3 vManager General Commands

This section describes the following general commands of vManager:

- [browse](#)
- [import](#)
- [export](#)
- [export_merge](#)
- [relocate](#)
- [launch](#)
- [suspend](#)
- [resume](#)
- [automation](#)
- [collect](#)
- [compact](#)
- [delete](#)
- [config](#)
- [export_config](#)
- [import_config](#)
- [export_views](#)
- [import_views](#)
- [edit](#)
- [stop](#)
- [convert](#)
- [snapshot](#)
- [server_info](#)
- [terminate_connection](#)
- [import_attributes](#)

- export_attributes

B.2.3.1 browse

Displays a list of sessions and views in the database.

Note: The `browse` operation does not require loading of sessions or views.

Syntax

```
browse [-sessions [-view <sessions_view> [-filter <filter>]] |  
-views [-kind {sessions | tests | runs | messages | metrics | vplan}]  
[-tracking_configs]
```

- `-sessions` is used to list all of the sessions in the database. In case you want to filter the columns and rows such that it shows only the columns and rows based on a particular session view, use the following option with the `-sessions` option:

- `-view <sessions_view>` shows the columns and rows based on the specified sessions view.
 - `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

[attribute_name] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the `attribute_name` specified with the `-filter` option does not exist.

- `-views` is used to list all the available views (both predefined and user-defined) in the database. By default, all of the views are listed. To list views of only a specified type, you can use any of the following options with the `-views` options:

- `-kind sessions` lists only the session views.
 - `-kind tests` lists only the test views.
 - `-kind runs` lists only the run views.
 - `-kind messages` lists only the message views.

- ❑ -kind metrics lists only the metrics views.
- ❑ -kind vplan lists only the vplan views.
- -tracking_configs is used to list all the available tracking configurations.

Note: If -sessions or -views is not specified, then -sessions is assumed as it is the default option.

Examples

To view a list of all the sessions in the database, use:

```
vmanager> browse -sessions
```

Figure B-7 on page 432 shows the output of the above command.

Figure B-7 Result of browse -sessions Command

Session	Status	Name	Tot Runs	#Passed	#Fail	#Waiting	#Running	#Other	Start Time	Owner
completed		cdn_uart.ruchikas.12_06_13_14_59_26_6338	18	0	18	0	0	0	Wed Jun 13 14:59:26 IST 2012	ruchikas
completed		cdn_uart.segal.11_01_04_23_14_13_1797	18	16	2	0	0	0	Wed Jan 05 02:44:13 IST 2011	ruchikas

In case you want to list the sessions such that it shows columns based on a particular view, use the -view option with the browse -sessions command:

For example, to list the sessions from the database such that it shows only the columns based on the view named segal, use the following command:

```
vmanager> browse -sessions -view segal
```

To view a list of available views (both predefined and user-defined), use:

```
vmanager> browse -views
```

Figure B-8 on page 433 shows the output of the above command.

Figure B-8 Result of browse -views Command

```
vmgr> browse -views

Sessions views:
View All_Sessions
View My Sessions

View segal created by ruchikas
View ruchikas created by ruchikas

Runs views:
View Incomplete_Runs
View Pass_Runs
View Failed_Runs
View All_Runs

Tests views:
View Test_Hierarchy

Messages views:
View First_Failure
View All_Messages
View All_Failures

Vplan views:
View All_Vplan
View Vplan_FSM
View Vplan_Functional
View Vplan_Toggle
View Vplan_Block
View Vplan_Expression
View Vplan_Assertion
View Vplan_Code
View Vplan_CoverGroup
View Relative_Grade
View Completion_Grade
```

The above result shows both predefined and user-defined views of all the kinds, such as sessions, runs, tests, messages, and vPlan.

To display a list of only the views of kind runs, use:

```
vmanager> browse -views -kind runs
```

[Figure B-9 on page 433](#) shows the output of the above command.

Figure B-9 Result of browse -views -kind runs Command

```
vmgr> browse -views -kind runs
View Incomplete_Runs
View Pass_Runs
View Failed_Runs
View All_Runs
```

The above result now displays only the runs views.

B.2.3.2 import

Imports a session from `vsوف` or `vsوفx` file to vManager database.

Syntax

```
import
[-session[s]] {<file> | <directory> [-recursive]}
[-load]
[-top_dir <top_dir>]
```

where

- `-session[s]` indicates that you want to import a session.
- `<file>` is the name of the `vsوف` file to be imported.
- `<directory>` is the name of the directory from which the `vsوف` file(s) must be imported. You can also specify an optional parameter `-recursive` with the directory name to import the `vsوفs` recursively from the specified directory.

Note: You can use wildcards in file and directory names and also import multiple `vsوف` or `vsوفx` files.

- `-load` loads the session after importing. By default, the session is not loaded.
- `-top_dir <top_dir>` specifies the location where the session directory must be uncompressed. This option is used if the `-with_session_dir` option was used at the time of exporting the session.

Examples

For example, to import session named

`cdn_uart.segal.11_01_04_23_14_13_1797/cdn_uart_backup.vsوف`, use the following command:

```
import -sessions
/servers/scratch03/segal/doc_examples/cdn_uart.segal.11_01_04_23_14_13_1797/cdn_u
art_backup.vsوف
```

B.2.3.3 export

Allows you to export a session from the database to a `vssofx` file. Exporting a session helps you at the time of archiving and also when you want to move sessions from one server to another.

The exported file (`vssofx`) can be imported only in vManager using the `import` command.

Note: After you import the `vssofx` file, you should use the `relocate` command to update the references to vManager database.

Syntax

```
export
[-session[s]] <list_of_session_names> | <sessions_view_name> [-filter <filter>]
[-vssofx <file>]
[-overwrite]
[-with_session_dir]
[-delete_session]
[-delete_session_with_session_dir]
```

where

- `-session[s]` lets you specify the sessions to be exported. Sessions to be exported can be specified as:

- `<list_of_session_names>` specifies the name(s) of the sessions to be exported. You can specify more than one session on a single command line by separating the session names with a comma (,).
 - `<sessions_view_name>` specifies the name of the view that was created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.
 - `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

`[attribute_name]:[operator][value]`

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `!=`, `range`, `~`, `!~`}

For example:

`-filter name:~comp`

You can also specify multiple filters on the command line. For example:

`-filter owner:~shaim -filter status:=passed`

Note: The filter specified on the command line will override the existing value in the

view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

Note: When exporting multiple sessions, it is a best practice is to use view name.

- -vssofx <file> specifies the file where the session will be exported.

If <file> includes just the name of the file, then the command creates the <file> in the current working directory. For example, if <file> is specified as:

```
export -sessions cdn_uart.segal.11_01_04_23_14_13_1797 -vssofx my_file1
```

then file named my_file1 is created in the current working directory.

Note: If <file> already exists, the file is not overwritten unless the -overwrite option is used.

If <file> includes the complete path, then the command creates a file by the name mentioned at the lowest level in the path. For example, if <file> is specified as:

```
export -sessions cdn_uart.segal.11_01_04_23_14_13_1797 -vssofx  
data/mysessions/my_file_nov1
```

then file named my_file_nov1 is created under data/mysessions. In this case, the path (data/mysessions) must exist. The file my_file_nov1 must not exist, as it will be created and not overwritten unless the -overwrite option is used.

- -overwrite enables overwriting of the existing vssofx file. By default, the vssofx file is not overwritten and an error is reported.
- -with_session_dir includes the session directory in the exported VSOFX file. By default, the session directory is not included in the exported VSOFX file.

Note: If you use the -with_session_dir option while exporting, then at the time of import, you must use the -top_dir option to specify the location where the session directory must be uncompressed.

- -delete_session deletes the session after exporting it.
- -delete_session_with_session_dir deletes session as well as the session directory after the export action.

Examples

For example, to export session named cdn_uart.segal.11_01_04_23_14_13_1797 to a vssofx file named my_session_nov1, use the following command:

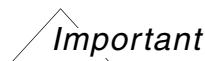
```
export -sessions cdn_uart.segal.11_01_04_23_14_13_1797 -vssofx my_session_nov1
```

To export session `cdn_uart.segal.11_01_04_23_14_13_1797` (including the session directory) and to delete the session after export, use:

```
export -sessions cdn_uart.segal.11_01_04_23_14_13_1797 -with_session_dir  
-delete_session
```

The above command will:

- Export the specified session including the session directory to a vsofx file named `cdn_uart.segal.11_01_04_23_14_13_1797.vssofx` (in the current working directory), and
- Delete the session after exporting it.



To import the above vssofx file, you must specify the location where the session directory must be uncompressed using the `-top_dir` option of the `import` command, as shown below:

```
import cdn_uart.segal.11_01_04_23_14_13_1797.vssofx -top_dir  
/servers/scratch03/my_sessions/
```

In the above command, `-top_dir` specifies the location where the session directory must be uncompressed.

B.2.3.4 `export_merge`

By default, vManager uses an internal automatic merge mechanism for any operation that requires merge (for example —collect runs, analyze session coverage, compaction, and so on). The merge results in such cases are not available to the user for analysis outside of vManager.

The `export_merge` command allows you to save the automatically merged coverage to a location so that it can be used later for analysis outside of vManager (for example, IMC).

Syntax

```
export_merge  
  [-session[s]]  
  {<list_of_session_names> | <sessions_view_name> [-filter <filter>]}  
  -target <target_location>
```

where

- `-session[s]` lets you specify the sessions to be exported. Sessions to be exported can be specified as:

- ❑ <list_of_session_names> specifies the name(s) of the sessions to be exported. You can specify more than one session on a single command line by separating the session names with a comma (,).
- ❑ <sessions_view_name> specifies the name of the view that was created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.
 - -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute_name] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

-filter name:~comp

You can also specify multiple filters on the command line. For example:

-filter owner:~shaim -filter status:=passed

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

Note: When exporting multiple sessions, it is a best practice is to use view name.

- -target <target_location> specifies the location where the merged coverage must be exported/stored. The specified <target_location> must exist before you execute the command. After executing the command, a folder with the name as the session name (that is being exported) is created under the <target_location>. It includes the merged coverage files that you can use later for analysis. The following structure gets created:

```
<target_location>
  /<session_name>
    /<verification_scope>
      <tool>_<file_name>.ucd(merged)
      <tool>_<file_name>.ucm(merged)
      ...
      ...
```

Note: If multiple sessions are specified, then for each session, a folder (<session_name>) will be created in the <target_location>.

Each leaf of this tree structure can be loaded into IMC for analysis.

Note: You can also export merged coverage data of a session from vManager GUI. For more details, see [Exporting Merged Coverage of Sessions](#) on page 92.

Examples

To export merged coverage of session named `cdn_uart.segal.11_01_04_23_14_13_1797` to target location `/hm/mike/exported_merged_sessions`, use the following command:

```
export_merge -session cdn_uart.segal.11_01_04_23_14_13_1797 -target  
/hm/mike/exported_merged_sessions/
```

The above command will create a directory named

`cdn_uart.segal.11_01_04_23_14_13_1797` in the target

location `/hm/mike/exported_merged_sessions/` to store exported merged coverage data. The following structure will be created:

```
/hm/mike/exported_merged_sessions  
  /cdn_uart.segal.11_01_04_23_14_13_1797  
    /default  
      icc_5e1e1a4b_00000000.ucd(merged)  
      icc_5e1e1a4b_00000000.ucm(merged)  
      sn_00000000_7f23ee73.ucd(merged)  
      sn_00000000_7f23ee73.ucm(merged)
```

Next time when you export the merged coverage for this session using the same command, the already merged coverage files will be overwritten.

B.2.3.5 relocate

The `relocate` command enables you to relocate the specified sessions.

Syntax

```
relocate  
[-session]  <list_of_session_names> |  <sessions_view_name> [-filter <filter>]  
-top_dir <new_top_dir>  
[-delete_old]
```

where

- `-session` lets you specify the sessions to be relocated. Sessions to be relocated can be specified as:
 - `<list_of_session_names>` specifies the name(s) of the sessions to be relocated. You can specify more than one session on a single command line by separating the session names with a comma (,).

- ❑ <sessions_view_name> specifies the name of the view that was pre-defined or created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.
- -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute_name] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

-filter name:~comp

You can also specify multiple filters on the command line. For example:

-filter owner:~shaim -filter status:=passed

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

- -top_dir <new_top_dir> lets you specify the new session's top directory. The <new_top_dir> is created in the current working directory.
- -delete_old deletes the old relocated session directory, if it exists.

Examples

To relocate session cdn_uart.segal.11_01_04_23_14_13_1797 to projects/ben, use:

```
vmanager> relocate -session cdn_uart.segal.11_01_04_23_14_13_1797 -top_dir  
projects/ben
```

The above command will create a directory path projects/ben in the current working directory and relocate the session cdn_uart.segal.11_01_04_23_14_13_1797 to the projects/ben directory.

To relocate session cdn_uart.segal.11_01_04_23_14_13_1797 to mike and to delete the earlier relocated session directory, use:

```
vmanager> relocate -session cdn_uart.segal.11_01_04_23_14_13_1797 -top_dir mike  
-delete_old
```

The above command will create a directory mike in the current working directory and relocate the session cdn_uart.segal.11_01_04_23_14_13_1797 to the mike directory. The command will also delete the session from the location where it was earlier relocated. In this case, the session will be deleted from the projects/ben directory.

You can also relocate a session from vManager GUI. For more details, see [Relocating Session](#) on page 105.

B.2.3.6 launch

The `launch` command enables you to launch a new session using the specified vsif file.

Syntax

```
launch [-vsif] <vsif_path>
[-parse_only]
[-chain <session_name>]
[-attribute <name=value>]
[-define <def_name>[=<def_value>]]
[-load]
[-wait]
```

where

- `-vsif <vsif_path>` specifies the path of the vsif file that must be used to launch the session.
- `-parse_only` only parses the specified vsif file, and checks for its validity. It does not launch the session.
- `-chain <session_name>` chains the newly launched session to an existing session. When this option is used, then the results of newly launched session are appended to the existing session `<session_name>`, if it is found.
- `-attribute <name=value>` allows you to override the attributes defined in the vsif file. For values with spaces, use the following syntax:
`-attribute "<name>=<value>"`
- `-define <def_name>[=<def_value>]` allows you to define the specified preprocessing directive (`#define def-name def-value`) as if it was added at the beginning of the vsif file.

Note: Multiple defines can be specified on the command line by separating them with a comma (,). For example:

```
launch -define A,B=C -vsif my_vsif.vsif
```

- `-load` loads the session after launching it. By default, the session is not loaded.
- `-wait` causes the `vmanager` command-line to wait for session completion before executing the next command.

Note: The environment variable `MY_WORK_AREA` and `MY_REGRESSION_AREA` must be set before using the `launch` command. For example, to set these variables, you can execute the following commands before invoking vManager:

```
setenv MY_WORK_AREA /servers/scratch03/bob/doc_examples/  
setenv MY_REGRESSION_AREA $MY_WORK_AREA/sessions
```

Examples

To launch a session from vsif file named `cdn_uart_apb_regress.vsif`, use:

```
launch -vsif  
/servers/scratch03/bob/doc_examples/cdn_uart.segal.11_01_04_23_14_13_1797/chain_0  
/cdn_uart_apb_regress.vsif -wait
```

The above command will launch the session and the results will be stored in the regression area defined using the environment variable `MY_REGRESSION_AREA`. The use of `wait` option in the above command will cause the command-line to wait until the session completes.

Consider following set of commands:

```
launch test1.vsif -wait  
launch test2.vsif -chain session1.ben.14_03_09_18_24_09_8740
```

The first `launch` command launches `test1.vsif` and waits till session completes. The next `launch` command launches `test2.vsif` and appends the results to existing session `session1.ben.14_03_09_18_24_09_8740` (which was created by the execution of the first `launch` command).

B.2.3.7 suspend

Suspends the specified sessions. Applicable only for running session and only for sessions launched with new runner.

Syntax

```
suspend  
-session[s] {<list_of_session_names> | <sessions_view_name> [-filter <filter>]}
```

where

- `-session[s]` lets you specify the sessions to be suspended. Sessions to be suspended can be specified as:

- ❑ <list_of_session_names> specifies the name(s) of the sessions to be suspended. You can specify more than one session on a single command line by separating the session names with a comma (,).
- ❑ <sessions_view_name> specifies the name of the view that was pre-defined or created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,).
 - -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute_name] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

-filter name:~comp

You can also specify multiple filters on the command line. For example:

-filter owner:~shaim -filter status:=passed

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

Examples

To suspend a session named cdn_uart.segal.11_01_04_23_14_13_1797 from the database, use:

```
vmanager> suspend -sessions cdn_uart.segal.11_01_04_23_14_13_1797
```

To suspend sessions named cdn_uart.segal.11_01_04_23_14_13_1795 and cdn_uart.bob.12_06_13_14_59_26_6338, use:

```
vmanager> suspend -sessions cdn_uart.segal.11_01_04_23_14_13_1795,  
cdn_uart.bob.12_06_13_14_59_26_6338
```

or, you can use multiple suspend -sessions commands as:

```
vmanager> suspend -sessions cdn_uart.segal.11_01_04_23_14_13_1795
```

```
vmanager> suspend -sessions cdn_uart.bob.12_06_13_14_59_26_6338
```

B.2.3.8 resume

Resumes the specified suspended sessions. Applicable only for suspended sessions.

Syntax

```
resume  
-session[s] {<list_of_session_names> | <sessions_view_name> [-filter <filter>]}
```

where

- -session[s] lets you specify the sessions to be resumed. Sessions to be resumed can be specified as:

- <list_of_session_names> specifies the name(s) of the sessions to be resumed. You can specify more than one session on a single command line by separating the session names with a comma (,).
- <sessions_view_name> specifies the name of the view that was pre-defined or created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,).
 - -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute_name]:[operator][value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

-filter name:~comp

You can also specify multiple filters on the command line. For example:

-filter owner:~shaim -filter status:=passed

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

Examples

To resume a session named cdn_uart.segal.11_01_04_23_14_13_1797 from the database, use:

```
vmanager> resume -sessions cdn_uart.segal.11_01_04_23_14_13_1797
```

To resume sessions named cdn_uart.segal.11_01_04_23_14_13_1795 and cdn_uart.bob.12_06_13_14_59_26_6338, use:

```
vmanager> resume -sessions cdn_uart.segal.11_01_04_23_14_13_1795,  
cdn_uart.bob.12_06_13_14_59_26_6338
```

or, you can use multiple resume -sessions commands as:

```
vmanager> resume -sessions cdn_uart.segal.11_01_04_23_14_13_1795
vmanager> resume -sessions cdn_uart.bob.12_06_13_14_59_26_6338
```

B.2.3.9 automation

The `automation` command is used to set the automation options for user actions such as, clean, report, email, and rerun.

Syntax

```
automation
[-clean <clean_script_path>]
[-email <email_list>]
[-stop_session <stop_session_rules>]
[-merge]
[-rerun [-rerun_view <view> [-rerun_filter <filter>] -rerun_scheme <scheme>
         [-rerun_failures <view> [-rerun_max_runs_per_failure <num>]]]]
[-report_vplan]
    [-report_vplan_file <vplan_file>]
    [-report_vplan_refinement_file <refinement_file>]
    [-report_vplan_out <output_directory>]
    [-report_vplan_title <title>]
    [-report_vplan_overwrite]
    [-report_vplan_email <address>]
    [-report_vplan_view <view> [-report_vplan_filter <filter>]]
    [-report_vplan_perspective <perspective>]
[-report_tests]
    [-report_tests_out <output_directory>]
    [-report_tests_title <title>]
    [-report_tests_overwrite]
    [-report_tests_email <address>]
    [-report_tests_view <view>]
[-report_sessions]
    [-report_sessions_out <output_directory>]
    [-report_sessions_title <title>]
    [-report_sessions_overwrite]
    [-report_sessions_email <address>]
    [-report_sessions_view <view> [-report_sessions_filter <filter>]]
[-report_metrics]
    [-report_metrics_refinement_file <refinement_file>]
    [-report_metrics_out <output_directory>]
    [-report_metrics_title <title>]
    [-report_metrics_overwrite]
    [-report_metrics_email <address>]
    [-report_metrics_view <view> [-report_metrics_filter <filter>]]
    [-report_metrics_verification_scope <scope>]
    { [-report_metrics_summary] |
      [-report_metrics_detail [-report_metrics_metrics all | code | fsm |
                               functional | block | expression | toggle |
                               state | transition | arc | assertion |
                               covergroup] ]
      [-report_metrics_kind expand | compact ]
      [-report_metrics_source <on | off>]
      [-report_metrics_covered | - report_metrics_uncovered |
       -report_metrics_both | -report_metrics_all |
       -report_metrics_excludes]} }
```

```
[-report_metrics_types | -report_metrics_instances]
[-report_metrics_list <paths_list>]
[-report_summary]
  [-report_summary_out <output_file>]
  [-report_summary_overwrite]
  [-report_summary_title <title>]
  [-report_summary_email <address>]
  [-report_summary_all]
  [-report_summary_sessions] [-report_summary_sessions_view <view>]
  [-report_summary_tests] [-report_summary_tests_view <view>]
    [-report_summary_tests_depth <tree_depth>]
  [-report_summary_metrics] [-report_summary_metrics_view <view>]
    [-report_summary_metrics_depth <tree_depth>]
    [-report_summary_metrics_verification_scope <scope>]
    [-report_summary_metrics_refinement_file <refinement_file>]
  [-report_summary_vplan] [-report_summary_vplan_view <view>]
    [-report_summary_vplan_depth <tree_depth>]
    [-report_summary_perspective <perspective>]
    [-report_summary_vplan_file <vplan_file>]
    [-report_summary_vplan_refinement_file <refinement_file>]
```

where

- `-clean <clean_script_path>` runs the specified clean script after session end.
Note: The clean script is executed from the server, and can include only the server environment details.
- `-email <email_list>` sends an email to addresses specified in `<email_list>` after session end.
- `-merge` enables merging of coverage results after session end. This speeds-up the analysis tasks.

Note: Merging by default is an incremental process. It is time-consuming and involves significant computation and heavy disk access. To save time, you can accelerate the merge calculation by configuring vManager such that the merge calculation is dispatched to multiple processes. For more details, see [Incisive vManager Installation and Configuration Guide](#).

- `-stop_session <stop_session_rules>` enables you to stop session based on conditions specified in the `stop_session_rules`. You can specify more than one rule by separating the rule names with a comma (,). For example, to enable stopping of session based on conditions specified in rules `rule_1` and `rule_3`, use:

```
automation -stop_session rule_1,rule_3
```

The rules specified on command line are already created using vManager GUI. For more details, see [Stopping Session Based on Conditions](#) on page 100.

You can also enable stopping of session by setting the configuration using the following command:

```
config automation.auto_session_stop_rules -set "rule_1,rule_2"
```

Note: If automation -stop_session command is not used, then the settings from the configuration (config automation command) are used. If configuration is set using the config command and automation -stop_session command is also used, then the automation -stop_session command overrides the configuration settings.

- -rerun reruns the failed runs after the session ends.
- -rerun_view <view> specifies the view for automatic rerun.
 - -rerun_filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

For example, -rerun_filter name:~comp will apply filter on specified view to consider only the items that have comp in the name field.

- -rerun_scheme <rerun_scheme> specifies the rerun scheme for automatic rerun. If not specified, BATCH_DEBUG will be used.
- -rerun_failures <view> specifies the *Failures* view for automatic rerun.
- -rerun_max_runs_per_failure <num> specifies the maximum number of runs to be rerun from each failure kind.

Note: The maximum number of runs per failure is limited to 1000 runs. In addition, the total rerun runs (from all failures) is limited to 10,000 runs.

- -report_sessions allows you to create a sessions report.

-report_sessions_out <output_directory> redirects the report files to an alternate location instead of the default html_<timestamp> directory. In the absence of this option, by default, a directory named html_<timestamp> is created in the current working directory, and coverage reports along with the top-level summary page (index.html) are stored in the html_<timestamp> directory.

- -report_sessions_title <title> allows you to specify a title for the sessions report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Sessions Report*.
- -report_sessions_overwrite enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- -report_sessions_email <address> specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).

- `-report_sessions_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *My Sessions* is used to generate the report.

- `-report_sessions_filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

`[attribute] : [operator] [value]`

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `=`, `!=`, `range`, `~`, `!~`}

For example:

`-report_sessions_filter name:~comp`

You can also specify multiple filters on the command line. For example:

`-report_sessions_filter owner:~shaim -filter status:=passed`

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the `attribute_name` specified with the filter does not exist.

- `-report_tests` allows you to create a tests report.
- `-report_tests_out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestamp>` directory. In the absence of this option, by default, a directory named `html_<timestamp>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestamp>` directory.
- `-report_tests_title <title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Tests Report*.
- `-report_tests_overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- `-report_tests_email <address>` email address, a link to the generated report would be sent to the specified address. Multiple addresses can be specified in a comma separated list.
- `-report_tests_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *Test Hierarchy* is used to generate the report.
- `-report_vplan` allows you to create a vPlan report.
- `-report_vplan_file <vplan_file>` specifies the vPlan file that will be loaded before generating the report.

- `-report_vplan_refinement_file <refinement_file>` specifies the vPlan refinement file that will be loaded before generating the report.
- `-report_vplan_out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestamp>` directory. In the absence of this option, by default, a directory named `html_<timestamp>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestamp>` directory.
- `-report_vplan_title <title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *vPlan Report*.
- `-report_vplan_overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- `-report_vplan_email <address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-report_vplan_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All_vPlan* is used to generate the report.

- `-report_vplan_filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `!=`, `range`, `~`, `!~`}

For example:

```
-report_vplan_filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-report_vplan_filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the filter does not exist.

- `-report_vplan_perspective <perspective>` specifies the perspective to be used for generating the report. If not specified, the name of the plan is used.
- `-report_metrics` allows you to create a metrics report.
- `-report_metrics_all` reports all items in detailed pages.
- `-report_metrics_covered` reports only the covered items in detailed pages.

- `-report_metrics_uncovered` report only the uncovered items in detailed pages.
- `-report_metrics_both` reports both covered and uncovered items in detailed pages.
- `-report_metrics_excludes` reports only the excluded items in detailed pages.
- `-report_metrics_instances` indicates that the report will be generated only for the instances hierarchy.
- `-report_metrics_types` indicates that the report will be generated only for the types hierarchy.
- `-report_metrics_list <list_for_report>` specifies the names of types or instances for which the report will be generated. The default value is * . . . , which matches all the top-level instances and their children. It can include wildcard characters * and ?. Use . . . to specify all descendants of an instance. Instance names can be specified as hierarchical names, for example, top.inst1.inst2.
- `-report_metrics_detail` includes all the data (including bins) in the reports.
- `-report_metrics_kind <report_metrics_kind>` impacts the items presentation in detailed reports. The `<report_metrics_kind>` can be any of the following:
 - `expand`—Prints the bin information as a list without grouping similar bins. This is the default behavior. This option is supported only with HTML reports.
 - `compact`—Prints the bin information after grouping similar bins. This option is supported only with HTML reports.
- `-report_metrics_source <on|off>` enables or disables reporting of the source text and line numbers in the detailed report. By default, this information is printed in the report.
- `-report_metrics_summary` includes only instances/types hierarchy (no bins level details) in the report.
- `-report_metrics_metrics <metrics_type>` specifies the metrics types for which the report must be generated. The `<metrics_type>` can be any, or a combination of the following:
 - `code` for printing a detailed block, expression, and toggle coverage report
 - `fsm` for printing a detailed state, transition, and arc coverage report
 - `functional` for printing a detailed assertion and covergroup coverage report
 - `block` for printing a detailed block coverage report

- ❑ expression for printing a detailed expression coverage report
- ❑ toggle for printing a detailed toggle coverage report
- ❑ state for printing a detailed state coverage report
- ❑ transition for printing a detailed transition coverage report
- ❑ arc for printing a detailed arc coverage report
- ❑ assertion for printing a detailed assertion coverage report
- ❑ covergroup for printing a detailed covergroup coverage report
- ❑ overall for printing a detailed report for all metrics
- ❑ all for printing a detailed report for all metrics

Note: You can specify more than one metrics type by separating the metric types with a colon (:). If <metrics_type> is not specified, a detailed report for all metrics is generated.

- -report_metrics_out <output_directory> redirects the report files to an alternate location instead of the default html_<timestep> directory. In the absence of this option, by default, a directory named html_<timestep> is created in the current working directory, and coverage reports along with the top-level summary page (index.html) are stored in the html_<timestep> directory.
- -report_metrics_overwrite enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- -report_metrics_refinement_file <refinement_file> specifies the refinement file to be loaded before generating the report.
- -report_metrics_title <title> allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Metrics Report*.
- -report_metrics_email <address> specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- -report_metrics_view <view> allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All_Metrics* is used to generate the report.
 - ❑ -report_metrics_filter <filter> specifies filter on the specified view. The format for specifying the filter is:
[attribute] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-report_metrics_filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-report_metrics_filter owner:~shaim -report_metrics_filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the filter does not exist.

- `-report_metrics_verification_scope <scope>` specifies the verification scope to be used for generating the report. In the absence of this option, the default scope is used.
- `-report_summary` allows you to create a summary report.
- `-report_summary_all` includes all the sections (sessions, tests, metrics, and vPlan) in the summary report.
- `-report_summary_out <output_file>` redirects the report to an alternate file instead of the default `html_<timestamp>.html` file. In the absence of the `-out` option, by default, a file named `html_<timestamp>.html` is created in the current working directory.
- `-report_summary_overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- `-report_summary_email <address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-report_summary_metrics` includes the *Metrics* section in the report.
- `-report_summary_metrics_depth <tree_depth>` specifies the depth of the metrics tree to be included in the report. It can be any of the following:
 - level number — to indicate the number of levels to be included in the report.
 - all — to include all the levels (complete metrics tree) in the report.

Note: Default value is all.

- `-report_summary_metrics_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All_Metrics* is used to generate the report.

- `-report_summary_metrics_refinement_file <refinement_file>` specifies the refinement file to be loaded before generating the report.
- `-report_summary_metrics_verification_scope <scope>` specifies the verification scope to be used for generating the report. In the absence of this option, the default scope is used.
- `-report_summary_title <title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Summary Report*.
- `-report_summary_sessions` includes the *Sessions* section in the report.
- `-report_summary_sessions_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *My_Sessions* is used to generate the report.
- `-report_summary_tests` includes the *Tests* section in the report.
- `-report_summary_tests_depth <tree_depth>` specifies the depth of the tests tree to be included in the report. It can be any of the following:
 - level number — to indicate the number of levels to be included in the report.
 - all — to include all the levels (complete tests tree) in the report.

Note: Default value is all.

- `-report_summary_tests_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *Test_Hierarchy* is used to generate the report.
- `-report_summary_vplan` includes the *vPlan* section in the report.
- `-report_summary_vPlan_depth <tree_depth>` specifies the depth of the vPlan tree to be included in the report. It can be any of the following:
 - level number — to indicate the number of levels to be included in the report.
 - all — to include all the levels (complete vPlan tree) in the report.

Note: Default value is all.

- `-report_summary_vplan_view <view>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All_vPlan* is used to generate the report.

- `-report_summary_perspective <perspective>` lets you specify the vPlan perspective to be used when generating the summary report. If not specified, the default vPlan perspective is used.
- `-report_summary_vplan_file <vplan_file>` specifies the vplan file which be loaded before generating the automatic summary report.
- `-report_summary_vplan_refinement_file <refinement_file>` specifies the refinement file to be loaded before generating the report.

Examples

If you want to set automation such that whenever a session is launched, a sessions report is generated post session launch and a report is sent to `bob@cadence.com`, use the following command:

```
automation -report_sessions -report_sessions_email bob@cadence.com
```

B.2.3.10 collect

Allows you to create a new session by collecting runs result from the disk. This is a typical flow for users who run their regressions using external regression runner.

Syntax

```
collect
[-attribute <name=value>]
[-refine <pattern>]
[-depth <depth_limit>]
[-follow]
[-cwd]
{-log_only | -vsot_only}
[-append <session_name>]
[-load]
[-remove]
[-rescan]
[-merge <merge_dir>]
[-primary_runs <runs_dir>]
<dir1> <dir2> ...
```

where

- `-attribute <name=value>` allows you to specify the run's attribute as a name-value pair (for example '`owner=David`'), which will override the value of the specified attribute in the collected runs at the time of creating the new session.

Note: The format of the attributes for the collect command is the same as the format that is used for the vsif file; and in general, it means using lower case and replacing spaces

with '_'. For example, *Name* = name and *Start Time* = start_time.

Note: For values with spaces, use the following syntax:

```
-attribute "<name>=<value>"
```

For example:

```
collect -attribute "description=some description" cov_work
```

- **-refine <pattern>** specifies the regular expression pattern to refine the search on the runs that are being collected. If the specified <pattern> matches the full path to a directory under the top directory, it will be collected. The <pattern> can include wildcards or regular expression. For example, to refine the run directories and perform the collect operation only on the run directories whose names start with my_run, use one of the following options:

```
-refine /my_run/
```

```
-refine my_run*
```

- **-depth <depth_limit>** specifies the limit below which the run directories will not be collected.
- **-follow** causes the collect command to follow symbolic links.
- **-cwd** causes the collect command to scan logs (at the time of collection) in the current working directory instead of the collected run directory.
- **-log_only** enables collection of only the log files. With this option, the collection of vsos files and coverage run directories is ignored. The logs collected are then scanned, and failures are extracted using a custom scan script (if available) and default scan script.

Note: To scan logs, you must add scan attributes to your collect command. You can consider specifying a custom scan script, or use the default scanning and specify the scan arguments.

Example: Using a custom scan script:

```
collect -log_only -attribute scan_script=my_scan_script.sh cov_work
```

Example: Using default scanning:

```
collect -log_only -attribute auto_scan_args=ius.flt cov_work
```

- **-vsos_only** enables collection of only a single run vsos file. With this option, run directories that include no vsos (even directories with coverage databases) is ignored.
- **-append <session_name>** appends the collected runs to an existing specified session. The specified session must be a session previously created using the collect command. For example, if my_collected_sess is a session that was created using the collect command, then you can append other collected runs to it by running the following:

```
collect -append my_collected_sess cov_work
```

where `cov_work` is the target directory that contains the runs to be collected.

- `-load` loads the created session to the current context. For example, consider the following command:

```
collect -load cov_work
```

This command will collect the runs that are in the target directory `cov_work` to a session, and then load this session to the current context.

- `-remove` deletes the single run vs of files after the collect operation.
- `-rescan` rescans logs instead of considering scan results from single run vs of files.
- `-merge <merge_dir>` enables merging of coverage in the collected session to the specified `<merge_dir>`. This improves the performance because when the user later performs merge analysis, it is faster as the tool uses the merged results instead of merging all `ucd` and `ucm` files.

Note: Merging by default is an incremental process. It is time-consuming and involves significant computation and heavy disk access. To save time, you can accelerate the merge calculation by configuring vManager such that the merge calculation is dispatched to multiple processes. This speeds up the merge computation. For more details, see [Incisive vManager Installation and Configuration Guide](#).

- `-primary_runs <runs_dir>` specifies the primary run for the merge operation. Only ICC runs can be specified as primary runs. An error is reported if a JG/UXE run is specified as the primary run. In the absence of this option, the model with the most recent timestamp is taken as the primary run.
- `<dir1> <dir2> ...` specifies target directory where actual runs can be found.

Examples

To enable collection of a single run vs of file and also to load the created session, use:

```
collect -vs of_only  
/servers/scratch03/bob/doc_examples/cdn_uart.segal.11_01_04_23_14_13_1797
```

B.2.3.11 compact

Creates a new compact session from the specified runs.

Syntax

```
compact
[-view <runs_view_name> [-filter <filter>]]
[-top_dir <directory>]
[-session_name <name>]
[-include_errors]
[-include_warnings]
[-append]
```

where

- **-view <runs_view_name>** lets you specify the view to be used for the compact operation. The runs are compacted from the loaded sessions that match the filter of the specified view. If no view is specified, then the default runs analysis page view is used.
 - **-filter <filter>** specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

- **-top_dir <directory>** lets you specify the top directory of the compacted session. In the absence of this option, the <directory> is created in the current working directory.
- **-session_name <name>** lets you specify the name of the compacted session. In the absence of this option, the compacted session is named as compact_username_dd-MM-yyyy_hh-mm-ss.
- **-include_errors** copies the errors from the flat runs to the compact runs.
- **-include_warnings** copies the warnings from the flat runs to the compact runs.
- **-append** appends the new compacted runs to an existing compacted session.

Note: The -append option can be specified only with a session which was compacted earlier using the -session_name option in a previous compact command. For example, consider the following commands:

```
compact -session_name mike_mar13  
compact -session_name mike_mar13 -append
```

The first command creates a compact session named `mike_mar13` from the runs in the loaded session. The second command will append new runs to already compacted session `mike_mar13`.

Examples

To create a compact session named `oct_13` from the runs in the loaded session and to include associated errors in the compacted session, use:

```
compact -session_name oct_13 -include_errors
```

The above command will create a compacted session named `oct_13` and also include errors from the original session to the compacted session. As the `top_dir` is not specified, the compacted session will be created in the current working directory.

To create a compact session such that it includes the runs specified in the view `failed_oct`, use:

```
compact -view failed_oct
```

The above command will create a compacted session with the default name `compact_username_dd-MM-yyyy_hh~mm~ss` in the current working directory and will include the runs from the view `failed_oct`.

For details on compacting sessions in GUI, see [Creating Compact Sessions](#) on page 111.

B.2.3.12 delete

Deletes the specified sessions from the database.

Syntax

```
delete  
-session[s] {<session_name> | <sessions_view_name> [-filter <filter>]}  
[-with_session_dir]
```

where

- `-session[s]` indicates that you want to delete a session. Sessions to be deleted can be specified as:
 - `<session_name>` specifies the name of the session to be deleted.

You can specify more than one session on a single command line by separating the session names with a comma (,).

- ❑ <sessions_view_name> specifies the name of the view (that was created using vManager GUI) to be deleted. You can specify more than one view on a single command line by separating the view names with a comma (,).
 - -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

-filter name:~comp

You can also specify multiple filters on the command line. For example:

-filter owner:~shaim -filter status:=passed

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

- -with_session_dir causes removal of session directory also from the file system at the time of delete operation. In the absence of this option, session is deleted only from the database.

Examples

To delete a session named cdn_uart.segal.11_01_04_23_14_13_1797 from the database, use:

```
vmanager> delete -sessions cdn_uart.segal.11_01_04_23_14_13_1797
```

To delete sessions named cdn_uart.segal.11_01_04_23_14_13_1795 and cdn_uart.bob.12_06_13_14_59_26_6338, use:

```
vmanager> delete -sessions cdn_uart.segal.11_01_04_23_14_13_1795,  
cdn_uart.bob.12_06_13_14_59_26_6338
```

or, you can use multiple delete -sessions commands as:

```
vmanager> delete -sessions cdn_uart.segal.11_01_04_23_14_13_1795  
vmanager> delete -sessions cdn_uart.bob.12_06_13_14_59_26_6338
```

To delete sessions that are a part of view named my_failed_sessions, use:

```
vmanager> delete -sessions my_failed_sessions
```

To delete sessions that are a part of views named `my_passed_sessions` and `my_cdn_sessions`, use:

```
vmanger> delete -sessions my_passed_sessions,my_cdn_sessions
```

B.2.3.13 config

Lets you view the value, set the value, or restore the value of configurable items.

By default, when you launch vManager, configurable items are set to a specific value.

Using the `config` command, you can perform the following tasks:

- [Viewing Configurable Items](#)
- [Setting Values for Configurable Items](#)
- [Restoring Values for Configurable Items](#)
- [Exporting Non-Default Configurable Values to a TCL File](#)

Viewing Configurable Items

The `config` command allows you to view a list of root-level categories, sub-categories, configurable items within a category, and their current values.

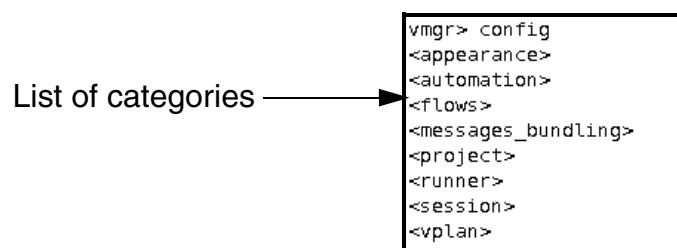
The `config` command without any parameters, list all the categories at the root-level.

For example, to view a list of all categories at the root-level, use:

```
config
```

[Figure B-10 on page 460](#) shows the output of the above command.

Figure B-10 Output of config Command



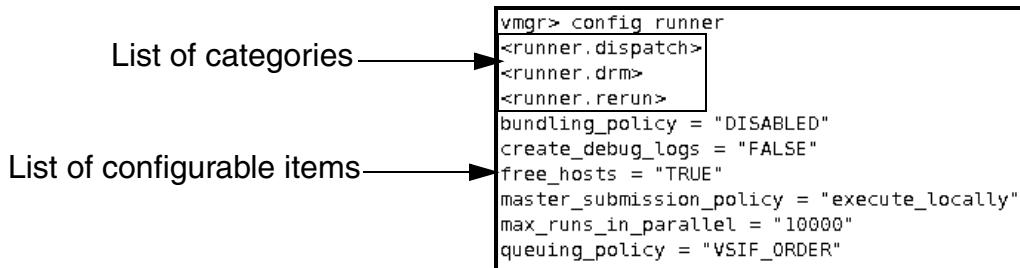
Note: The angular brackets in the command output indicate that the listed item is a category. You can view the list of items available within each of these categories by specifying the name of the category along with the `config` command.

To view a list of items within the category `runner`, use:

```
config runner
```

[Figure B-11](#) on page 461 shows the output of the above command.

Figure B-11 Output of config runner Command



The items with angular brackets indicate that it is a category. The items without the angular brackets are configurable items. The values of the configurable items are listed along with the item name.

To view value of a specific item, use:

```
config <config_item>
```

For example, to view the value of `auto_refresh_delta`, which belongs the category `session` and sub-category `refresh`, use:

```
vmanager> config session.refresh.auto_refresh_delta
```

Note: The configurable items are stated on the command-line with the complete path, which is specified as:

```
<category>.<config_item>
```

For example, to reference the configurable item `float_precision`, which is under category `appearance`, use:

```
appearance.float_precision
```

If a configurable item is within a sub-category, the path should also include the sub-category name as:

```
<category>.<sub_category>.<config_item>
```

For example, to reference the configurable item `auto_refresh_delta`, which belongs the category `session` and sub-category `Refresh`, use:

```
session.refresh.auto_refresh_delta
```

To view a list of all the configurable items along with their current values, use:

```
config -deep
```

The above command will list all the configurable items along with their current values.

Setting Values for Configurable Items

To set a value of a configurable item, use:

```
config <config_item> -set <value>
```

where

- *<config_item>* is the name (with the complete path) of the configurable item whose value you want to set.
- *-set* indicates that you want to set the value.
- *<value>* is the value you want to set for the configurable item.

Examples:

To set the value as 10 of configurable item `auto_refresh_delta`, which belongs the category `session` and sub-category `refresh`, use:

```
vmanager> config session.refresh.auto_refresh_delta -set 10
```

To disable automatic merging of coverage after session end, use:

```
vmanager> config automation.auto_merge -set false
```

Note: Automatic merging of coverage is enabled by default.

If automatic merging is enabled and you further want to enable parallel merge, use:

```
config parallel_merge.enable_parallel_merge -set true
```

Note: For more details, see [Incisive vManager Installation and Configuration Guide](#).

Restoring Values for Configurable Items

To restore the original values of configurable item, use:

```
config [<config_item>] -restore
```

where

- *<config_item>* is the name (with the complete path) of the configurable item whose value you want to restore.
- *-restore* indicates that you want to restore the default value.

For example, to restore the original value of configurable item `auto_refresh_delta`, which belongs the category `session` and sub-category `refresh`, use:

```
vmanager> config session.refresh.auto_refresh_delta -restore
```

To restore the original value of all configurable items, use:

```
vmanager> config -restore
```

Exporting Non-Default Configurable Values to a TCL File

To export all non-default configurable values to a TCL file, use:

```
config -export <filepath>
```

where

- `-export` indicates that you want to export all the non-default configuration values to a TCL file.
- `<filepath>` is the name (with the complete path) of the file to which the configuration values will be exported.

For example, to export all non-default configurable values to a TCL file named `conf_vals`, use:

```
vmanager> config -export conf_vals
```

After the above command is executed, all non-default configurable values are exported to a TCL file named `conf_vals.tcl`.

Note: A `.tcl` extension is automatically added to the file name if it is not specified at the command line.

B.2.3.14 `export_config`

Allows you to export all users configurations to a target location in an `xml` file format. The exported directory containing all users configurations can be imported in vManager using the `import_config` command.

Syntax

```
export_config <export_location>
```

where

- `<export_location>` is the location where the configurations will be exported. The specified `<export_location>` must exist before you execute the command.

After executing the command, a folder named `project-configurations-<date>` gets created under the `<export_location>`. This folder contains different xml files for each user in the system.

Note: In case you run multiple `export_config` commands on the same day, then the directory is named as `project-configurations-<date>_1` and so on.

Examples

To export all users configurations to target location `/hm/mike/exported_confs`, use the following command:

```
export_config /hm/mike/exported_confs/
```

The above command will create a directory named `project-configurations-<date>` in the export location `/hm/mike/exported_confs/` and the xml files will be stored in it.

When required, you can import the above exported directory using the `import_config` command.

B.2.3.15 import_config

Imports all users configurations available at the specified location.

The `import_config` operation will override all user's configuration in the system.

Syntax

```
import_config <configuration_location>
```

where

- `<configuration_location>` is the location of configuration files (one xml file per user) to be imported. The configurations directory specified for import must have been exported earlier using the `export_config` command.

Examples

To import configurations from

`/hm/mike/exported_confs/project-configurations-2015.11.20`, use the following command:

```
import_config /hm/mike/exported_confs/project-configurations-2015.11.20/
```

The above command will import all users configurations available at the specified location.

B.2.3.16 **export_views**

Allows you to export all the views that were created/saved in vManager GUI to a target location. The views are exported in form of `xml` files that can be imported later using the `import_views` command.

Syntax

```
export_views <export_location>
```

where

- `<export_location>` is the location where the views will be exported. The specified `<export_location>` must exist before you execute the command.

After executing the command, a folder named `exported-views-<date>` gets created under the `<export_location>`. This folder includes separate `xml` files for each of the view that was created/saved in vManager GUI.

Note: In case you run multiple `export_views` commands on the same day, then the directory is named as `exported-views-<date>_1` and so on.

Examples

Consider that a user has created two views named `runs_aug` (a RUNS view) and `sessions_aug` (a SESSION view) using vManager GUI.

To export these views to target location `/hm/mike/views`, use the following command:

```
export_views /hm/mike/views/
```

The above command creates a directory named `exported-views-<date>` in the export location `/hm/mike/views/` and stores following files in it:

```
mike.RUNS.runs_aug.xml  
mike.SESSIONS.sessions_aug.xml
```

When required, you can import the above exported view files using the `import_views` command.

B.2.3.17 **import_views**

Imports all the views from the specified location.

Syntax

```
import_views <import_location>
```

where

- <import_location> is the location from where the views will be imported. The views specified for import must have been exported earlier using the export_views command.

Note: If a view with the same name already exists, it will be updated.

Examples

To import the views from /hm/mike/views/exported-views-2015.11.20, use the following command:

```
import_views /hm/mike/views/exported-views-2015.11.20/
```

The above command will import all the views from the specified location.

Note: If you want to import both views as well as configurations, it is recommended to import views before importing the configurations. This is because configurations might contain references to views that must be available on the server at the time of importing configurations.

B.2.3.18edit

The edit command is used for editing sessions and runs attributes.

Syntax

```
edit
<sessions_specification>
[-runs <runs_view> [-runs_filter <filter>]
 [-messages <messages_view> [-messages_filter <filter>]]]
-attribute <attribute=value>
<sessions_specification> ::= <session_name> | <sessions_view_name>
```

where

- <sessions_specification> can be specified as any of the following:
 - <session_name> specifies the name of the session to be edited.

You can specify more than one session on a single command line by separating the session names with a comma (,).

Note: There is no restriction on number of sessions that can be edited.

- ❑ <sessions_view_name> specifies the name of the view that was created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.
- -runs <runs_view> specifies the name of the runs view that must be edited.
 - ❑ -runs_filter <filter> specifies filter on the specified runs view. The format for specifying the filter is:
[attribute] : [operator] [value]
where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}
For example:
-runs_filter name:~comp
You can also specify multiple filters on the command line. For example:
-runs_filter owner:~shaim -runs_filter status:=passed
Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the filter does not exist.
- -messages <messages_view> specifies the name of the messages view that must be edited.
 - ❑ -messages_filter <filter> specifies filter on the specified messages view. The format for specifying the filter is:
[attribute] : [operator] [value]
where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}
For example:
-messages_filter name:~comp
You can also specify multiple filters on the command line. For example:
-messages_filter owner:~shaim -messages_filter status:=passed
Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the filter does not exist.
- **Note:** If the -runs option is used, the command will apply on the matched runs attributes. If the -messages flag is also used, the command will apply on the matched messages.
- -attribute <attribute=value> specifies the name of the attribute to be edited and the value that must be assigned to it. For example, to edit the attribute

max_runs_in_parallel as 8 for session cdn_uart.segal.11_01_04_23_14_13_1797, use:

```
vmanager> edit cdn_uart.segal.11_01_04_23_14_13_1797 -attribute max_runs_in_parallel=8
```

Note: For values with spaces, use the following syntax:

```
-attribute '<name>=<value>'
```

For example:

```
edit my_sessions -runs my_runs -attribute 'comment=These are my runs'
```

- -list lists the attributes (related to session, runs, messages) that can be edited.

Examples

To list the attributes that can be edited, use:

```
vmanager> edit -list
```

Note: The changes made to any of the attributes immediately take into effect.

B.2.3.19 stop

Stops the specified sessions.

Syntax

```
stop -session[s] {<session_name> | <sessions_view_name> [-filter <filter>]}
```

where

-session[s] indicates that you want to stop a session. Sessions to be stopped can be specified as:

- <session_name> specifies the name of the session to be stopped.
- You can specify more than one session on a single command line by separating the session names with a comma (,).
- <sessions_view_name> specifies the name of the view (that was created using vManager GUI) to be stopped. All the sessions in the specified view are stopped. You can specify more than one view on a single command line by separating the view names with a comma (,).
 - -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

Examples

To stop a session named cdn_uart.segal.11_01_04_23_14_13_1797 from the database, use:

```
vmanager> stop -session[s] cdn_uart.segal.11_01_04_23_14_13_1797
```

To stop sessions named cdn_uart.segal.11_01_04_23_14_13_1795 and cdn_uart.bob.12_06_13_14_59_26_6338, use:

```
vmanager> stop -session[s] cdn_uart.segal.11_01_04_23_14_13_1795,  
cdn_uart.bob.12_06_13_14_59_26_6338
```

or, you can use multiple delete -session commands as:

```
vmanager> stop -session[s] cdn_uart.segal.11_01_04_23_14_13_1795
```

```
vmanager> stop -session[s] cdn_uart.bob.12_06_13_14_59_26_6338
```

To stop sessions that are a part of view named my_failed_sessions, use:

```
vmanager> stop -session[s] my_failed_sessions
```

To stop sessions that are a part of views named my_passed_sessions and my_cdn_sessions, use:

```
vmanager> stop -session[s] my_passed_sessions,my_cdn_sessions
```

B.2.3.20convert

The convert command enables you to convert legacy vPlans (created in EMGR 10.2).

Syntax

```
convert  
{-vplan <file> [-vext <file>]}
```

```
[‐overwrite]  
[‐ep_patterns]
```

where

- `-vplan <file>` specifies the name of the vplan file to be converted. After conversion the plan is saved with the same name but in a `.vplanx` format. For example, if you specify `system.vplan` as the file for conversion, then after conversion, the file will be saved as `system.vplanx`.

Note: If a file with the specified name already exists in a `.vplanx` format, then conversion fails and an error is reported unless the `-overwrite` option is used.

- `-vext <file>` specifies a legacy `*.vext` file corresponding to the legacy vplan. If the `-vext` option is used, then the legacy refinement file (vext) is converted to the new refinement file format, which can be one or both of these:

- `*.vRefine` (Metric tree refinements)
 - `*.vpRefine` (vPlan tree refinements)

Note: A `.vext` file (verification plan refinement file) contains exclusions to a given vPlan. This file, by default, has the same name as the verification plan, but with a `.vext` filename extension.

- `-overwrite` enables overwriting in case the file being converted already exists in a `.vplanx` format. If you use the `-overwrite` option and a plan with the same name already exists in a `.vplanx` format then the file is overwritten. In the absence of this option, an error is reported and the conversion fails.
- `-ep_patterns` further optimizes the converted vPlan, assuming all mappings were done directly from the metrics tree in EPlanner.

Examples

To convert `system.vplan` to a `.vplanx` format, use the following command.

```
convert -vplan system.vplan -overwrite
```

With the above command, `system.vplan` will be converted and saved as `system.vplanx`. As the `-overwrite` option is used in the command, if a file with the name `system.vplanx` already exists, then it will be overwritten.

B.2.3.21 snapshot

The `snapshot` command is used to create a tracking snapshot of the specified tracking configuration. You can use this command to create snapshots periodically using a cron job.

Syntax

```
snapshot -config <name> [-context] [-date <date>]
```

where

- `-config <name>` specifies the name of the configuration for which the snapshot has to be created.
- `-context` indicates that the currently loaded sessions must be used to create a snapshot. In the absence of this option, the sessions are selected based on the tracking configuration definition.
- `-date <date>` lets you specify the date of the snapshot. By default, in the absence of this option, the current date (on the client's machine) is used as the date of the snapshot. The default date format is `yyyy.MM.dd`. However, you can change this format using the *Configuration* dialog box. For more details, see [Opening Session Directory in Terminal on page 97](#).

Note: To see the date pattern configured in your environment, use:

```
config general.date_pattern
```

Examples

To create a snapshot for tracking configuration `config_1` using the currently loaded sessions, use the following command.

```
vmanager> snapshot -config config_1 -context
```

To create a snapshot for tracking configuration `config_1` with snapshot date as `2013.03.09`, use the following command.

```
vmanager> snapshot -config config_1 -date 2013.03.09
```

B.2.3.22 server_info

The `server_info` command is used to view the information about the vManager server, such as server build date, server version, server up time, number of sessions and runs, free database space, and the users connected to the server.

Syntax

```
server_info
```

Example

```
vmanager> server_info
```

[Figure B-12 on page 472](#) displays the output of the above command.

Figure B-12 Command Output (server_info)

The diagram shows the output of the `server_info` command in a terminal window. Annotations with arrows point to specific parts of the output:

- An arrow labeled "Server information" points to the top section of the output, which includes the command prompt (`vmanager>`), the server profile directory (`/servers/scratch03/ruchikas/ruchikas/ruch_aug27`), the build date (`2013-08-26 02:33:55`), the server version (`13.10`), the start date (`Tuesday, August 27, 2013`), uptime (`0 days, 1 hours 28 minutes 43 seconds`), sessions (`0 sessions`), runs (`0 runs`), and free database space (`DB free space: 1215283 Mb`).
- An arrow labeled "List of connected users" points to the table of connected users. The table has columns: clientid, Username, Host, Display, License Type, Slot, and a final column. The first row shows `| 5908090921420849166 | ruchikas | ldvopt257 | vlno-ruchikas1:1 | FULL | 1 | this |`. The second row shows `| 5908090921420849165 | ruchikas | ldvopt257 | vlno-ruchikas1:1 | FULL | 1 | |`.
- An arrow labeled "Current user's connection" points to the word `this` in the last column of the first row of the user list, indicating that the current user's connection is highlighted.

clientid	Username	Host	Display	License Type	Slot	
5908090921420849166	ruchikas	ldvopt257	vlno-ruchikas1:1	FULL	1	this
5908090921420849165	ruchikas	ldvopt257	vlno-ruchikas1:1	FULL	1	

The above output displays the server details, such as server build date, server version, server up time, number of sessions and runs, free database space, and the list of users connected to the server. For each user, information, such as client identification number, user name, host, display name, and license type is displayed.

The text `this` in the row of list of connected users indicates that it the current user's connection.

B.2.3.23 terminate_connection

The `terminate_connection` command is used to terminate a remote user's connection.

Syntax

```
terminate_connection -clientid <client_identification_number>
```

where `-clientid` specifies the client identification number of the remote user that must be terminated.

You can determine the `<client_identification_number>` of the remote user that must be terminated using the `server_info` command.

Example

```
vmanager> terminate_connection -clientid 5908090921420849165
```

The above command will terminate connection of the remote user with specified client identification number, close the GUI or CLI application of the remote user, and free up all the resources used by the remote user.

B.2.3.24 import_attributes

The `import_attributes` command is used to import the specified user-defined attributes file.

Syntax

```
import_attributes <attributes_files>
```

where

- `attributes_files` specifies the user-defined attributes file (CSV) to be imported. You can specify more than one files by separating the file names with a space.

Note: If multiple files are specified and some of the specified files are erroneous (cannot load properly), then vManager tries to import all the files; however, only correct files are applied and erroneous files are ignored.

Example

```
vmanager> import_attributes /home/<user_dir>/attrs.csv
```

The above command imports the user-defined attributes specified in the `attrs.csv` file.

B.2.3.25 export_attributes

The `export_attributes` command is used to export the user-defined attributes (that were imported earlier using the `import_attributes` command) to a file.

Syntax

```
export_attributes <target_attributes_files>
```

where

- `target_attributes_files` specifies the location and file (CSV) where the user-defined attributes will be exported.

Example

Consider an example where two user-defined files were imported using following commands:

```
vmanager> import_attributes /home/<user_dir>/attr1.csv  
vmanager> import_attributes /home/<user_dir>/attr2.csv
```

You can then export the user-defined attributes from both the files to a single file so that you do not need to import all the files again. For this, use the following command:

```
vmanager> export_attributes /home/<user_dir>/attr_all.csv
```

With the above command, user-defined attributes from all the files are saved to a single file attr_all.csv. Next time, instead of importing multiple CSV files, you can import just the attr_all.csv file.

B.2.3.26 recalc_uda

The `recalc_uda` command is used to recalculate the computed user-defined attributes (UDA) for the specified sessions and its sub-entities.

Syntax

```
recalc_uda  
  [-session[s]]  
  {<list_of_session_names> | <sessions_view_name> [-filter <filter>]}
```

where

- `-session[s]` indicates that you want to recalculate computed UDA for a session(s). It is the default option. Sessions for which the computed UDA has to be recalculated can be specified as:

- `<list_of_session_names>` specifies the name of sessions for which the computed UDA has to be recalculated.

You can specify more than one session on a single command line by separating the session names with a comma (,).

Note: There is no restriction on number of sessions that can be specified.

- `<sessions_view_name>` specifies the name of the view that was created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.

- `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

```
[attribute]:[operator][value]
```

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the `attribute_name` specified with the `-filter` option does not exist.

Example

To recalculate values of computed user-defined attributes in session named `cdn_uart.segal.11_01_04_23_14_13_1797`, use:

```
vmanager> recalc_uda -sessions cdn_uart.segal.11_01_04_23_14_13_1797
```

To recalculate values of user-defined attributes in session from views named `my_passed_sessions` and `my_cdn_sessions`, use:

```
vmanager> recalc_uda -sessions my_passed_sessions,my_cdn_sessions
```

B.2.4 Context-related Commands

This section describes the following context-related commands of vManager:

- [load](#)
- [show](#)
- [unload](#)
- [rerun](#)
- [vscan](#)
- [report_metrics](#)
- [report_sessions](#)
- [report_tests](#)

- [report_vplan](#)
- [report_tracking](#)
- [report_runs](#)
- [report_summary](#)
- [report_config](#)
- [rank](#)
- [correlate](#)
- [csv_export](#)
- [chart_export](#)

B.2.4.1 load

Allows you to load a session, verification plan, refinement file, or a vPlan refinement file.

Syntax

```
load {{-session[s]} {<session_name> | <sessions_view_name> [-filter <filter>]}} |  
-vplan <vplan_file> |  
-refinement <refinement_file> |  
-vplan_refinement <vplan_refinement_file>}]
```

where

- -session[s] indicates that you want to load a session. It is the default option. Sessions to be loaded can be specified as:

- <session_name> specifies the name of the session to be loaded.

You can specify more than one session on a single command line by separating the session names with a comma (,).

Note: There is no restriction on number of sessions that can be loaded.

- <sessions_view_name> specifies the name of the view that was created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma (,). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.
 - -filter <filter> specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

Note: You can also use multiple load -sessions commands to load more than one session or views.

- -vplan <vplan_file> specifies the verification plan to be loaded. You can specify only one verification plan with the load -vplan command.
Note: You cannot load more than one verification plan in a vManager session. In the case of multiple load -vplan commands, the verification plan specified with the last command is loaded. The verification plans specified with earlier load -vplan commands are automatically unloaded.
- -refinement <refinement_file> specifies the metrics refinement file to be loaded. You cannot specify multiple refinement files on a single command line but, you can load multiple refinement files using multiple load -refinement commands.
- -vplan_refinement <vplan_refinement_file> specifies the vPlan refinement file to be loaded. You cannot specify multiple vPlan refinement files on a single command line but, you can load multiple vplan refinement files using multiple load -vplan_refinement commands.

Note: A verification plan must be loaded before loading a vPlan refinement file.

Example: Loading Sessions

To load a session named cdn_uart.segal.11_01_04_23_14_13_1797, use:

```
vmanager> load -sessions cdn_uart.segal.11_01_04_23_14_13_1797
```

To load sessions named cdn_uart.segal.11_01_04_23_14_13_1795 and cdn_uart.bob.12_06_13_14_59_26_6338, use:

```
vmanager> load -sessions cdn_uart.segal.11_01_04_23_14_13_1795,  
cdn_uart.bob.12_06_13_14_59_26_6338
```

or, you can use multiple load -session commands as:

```
vmanager> load -sessions cdn_uart.segal.11_01_04_23_14_13_1795
```

```
vmanager> load -sessions cdn_uart.bob.12_06_13_14_59_26_6338
```

To load sessions from view named `my_failed_sessions`, use:

```
vmanager> load -sessions my_failed_sessions
```

To load sessions from views named `my_passed_sessions` and `my_cdn_sessions`, use:

```
vmanager> load -sessions my_passed_sessions,my_cdn_sessions
```

Example: Loading Refinement Files

To load a refinement file named `my_exclude.vRefine`, use:

```
vmanager> load -refinement /home/bob/ivm/ep/analyze/my_exclude.vRefine
```

To load refinement files named `1.vRefine` and `2.vRefine`, use:

```
vmanager> load -refinement /home/bob/ivm/ep/analyze/1.vRefine
```

```
vmanager> load -refinement /home/bob/ivm/ep/analyze/2.vRefine
```

Note: You cannot specify multiple refinement files on a single command line.

Example: Loading a Verification Plan

To load a verification plan named `newplan27_3.vplanx`, use:

```
vmanager> load -vplan /home/bob/ivm/ep/newplan27_3.vplanx
```

Note: You cannot load more than one verification plan in a vManager session.

Example: Loading vPlan Refinement Files

To load a vPlan refinement file named `my_vp_exclude.vpRefine`, use:

```
vmanager> load -vplan_refinement /home/bob/ivm/ep/my_vp_exclude.vpRefine
```

To load vPlan refinement files named `1.vpRefine` and `2.vpRefine`, use:

```
vmanager> load -vplan_refinement /home/bob/ivm/ep/analyze/1.vpRefine
```

```
vmanager> load -vplan_refinement /home/bob/ivm/ep/analyze/2.vpRefine
```

Note: You cannot specify multiple vplan refinement files on a single command line.

B.2.4.2 show

Allows you to list the currently loaded sessions, verification plans, refinement files, or vplan refinement files.

Syntax

```
show [-session[s]]  
[-vplan]  
[-refinement]  
[-vplan_refinement]
```

where

- -session[s] indicates that you want to view a list of loaded sessions.
- -vplan indicates that you want to view a list of loaded verification plans.
- -refinement indicates that you want to view a list of loaded refinement files.
- -vplan_refinement indicates that you want to view a list of loaded vPlan refinement files.

Note: In the absence of any of the options list of all the loaded sessions, verification plans, refinement files, and vPlan refinement files is displayed. In addition, you can also specify multiple options on a single command line.

Examples

To view a list of loaded sessions, use:

```
vmanager> show -sessions
```

To view a list of loaded verification plans and vPlan refinement files, use:

```
vmanager> show -vplan -vplan_refinement
```

To view a list of all the loaded sessions, verification plans, refinement files, and vPlan refinement files, use:

```
vmanager> show
```

B.2.4.3 unload

Unloads all previously loaded data (sessions, verification plans, refinement files, vplan refinement files).

Syntax

```
unload
```

B.2.4.4 rerun

The `rerun` command is used to rerun the selected runs from a context.

Syntax

```
rerun {-chain | -new_session | -write_vsif <vsif>}
[-view <runs_view_name> | <failures_view_name> | -from_ranking] [-filter <filter>]
[-failures [-max_runs_per_failure <num>]]
[-scheme <scheme_name>]
[-drm <LOAD_LEVELER | LSF | PARALEL_LOCAL | SERIAL_LOCAL | SGE>]
[-bundling]
[-queuing <VSIF_ORDER | SHORT_2_LONG | LONG_2_SHORT | ROUND_ROBIN>]
[-no_pre_session_script]
[-no_post_session_script]
[-no_pre_group_script]
[-no_post_group_script]
```

where

- `-chain | -new_session | -write_vsif <vsif>` specifies the run mode to be used at the time of `rerun`. It can be any of the following:
 - `-chain`—To chain each run to the original session. In other words, the original `vsif` file is extended, and the run results are written to a `chain_(N+1)` directory, where `chain_N` is the name of the original run directory.
 - `-new_session`—To execute the `rerun` as a separate session.
 - `-write_vsif <vsif>`—To save the `vsif` file without executing it.
- **Note:** If the run mode is not specified, then `-chain` is assumed.
- `-view` specifies the view that will be used for the `rerun` operation. You can specify any of the following:
 - `<runs_view_name>` the name of the runs view from which the runs will be `rerun`.
 - `<failures_view_name>` the name of the failures view from which the runs will be `rerun`.
 - `-from_ranking` indicates that the runs from the last ranking command will be `rerun`.

Note: If no view is specified, then all the runs that contributed to the ranking calculation (the ones that are not redundant) would be selected for `rerun`.

- `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

`[attribute]:[operator][value]`

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the -filter option does not exist.

- -failures indicates that the failures view (*First_Failures*) will be used to specify runs for the rerun operation. For each failure kind, the number of runs specified with the following option will be rerun.
 - -max_runs_per_failure <num> specifies the maximum number of runs to be rerun from each failure kind. The default is 1.

Note: The maximum number of runs per failure is limited to 1000 runs. In addition, the total rerun runs (from all failures) is limited to 10,000 runs.
- -scheme <scheme_name> specifies the rerun scheme to be used at the time of rerun. It can be any of the following:
 - AS_IS—To execute the run as specified in the original vsif.
 - BATCH—To execute the run in batch mode with timeout value as specified in the original vsif.
 - BATCH_DEBUG—To execute the run in batch-debug mode with timeout value as specified in the original vsif.
 - INTERACTIVE—To execute the run in interactive mode with timeout value as 0, which means timeout is disabled.
 - INTERACTIVE_DEBUG—To execute the run in interactive-debug mode with timeout value as 0, which means timeout is disabled.
 - RESCAN_LOGS—To rescan logs without rerunning the runs. This option is supported only with the -chain run mode option.

Note: The default rerun scheme is AS_IS.

- -drm <drm_policy> specifies the DRM policy to be used at the time of rerun. It can be any of the following:
 - LSF—Specifies the DPL-based integration with IBM Platform Load Sharing Facility (LSF).

- ❑ SGE — Specifies the DPL-based integration with the open source batch-queuing system, Sun Grid Engine (SGE), supported by Sun Microsystems.
 - ❑ LOAD_LEVELER — Specifies the DPL-based integration with IBM's LoadLeveler.
 - ❑ PARALLEL_LOCAL — Specifies the DPL-based execution of runs in parallel on the local machine.
 - ❑ SERIAL_LOCAL — Specifies the DPL-based execution of runs serially on the local machine.
- `-bundling` enables vManager to bundle calls to users scripts at the time of rerun. By default, bundling is disabled.
 - `-queuing <queueing_policy>` specifies the queuing policy to be used at the time of rerun. It can be any of the following:
 - ❑ VSIF_ORDER — Runs the tests in the order in which they are defined in the vsif.
 - ❑ SHORT_2_LONG — Makes the shortest runs first in the queue. This allows you to get the first test results from the regression as quickly as possible.
 - ❑ LONG_2_SHORT — Makes the longest runs first in the queue. This reduces the “long run tail” at the end of a session.
 - ❑ ROUND_ROBIN — Runs the tests in an arbitrary order, but creates a round-robin ordering between the groups and tests with a count greater than 1 or with multiple seeds. This mode is useful for vsifs that are arranged in logical groups, where you want to quickly see each type of test run once.

Note: If queuing policy is not specified, then by default VSIF_ORDER is assumed.

- `-no_post_group_script` disables running of `post_group_script` at the time of rerun.
- `-no_post_session_script` disables running of `post_session_script` at the time of rerun.
- `-no_pre_group_script` disables running of `pre_group_script` at the time of rerun.
- `-no_pre_session_script` disables running of `pre_session_script` at the time of rerun.

Examples

To rerun the runs from a view named `my_failed_runs` and to write a separate vsif without executing it, use:

```
rerun -write_vsif new.vsif -view my_failed_runs
```

To rerun the runs from the previous rank command and to write a separate vsif without executing it, use:

```
rank -hierarchy MEMORY/Phase2/toggle -html  
rerun -write_vsif /home/ztal/vsif/a.vsif -from_ranking
```

B.2.4.5 vscan

The vscan command initiates scanning of a vPlan tree with a given vPlan visitor defined in a TCL file. This command opens the vPlan visitor script file so that the user can explore the vPlan tree programmatically.

For more details on vPlan visitor, see [vPlan Visitor Interface](#) on page 316.

Syntax

```
vscan -visitor <file>
```

where

- `-visitor <file>` specifies the path of the visitor TCL file in which the vPlan details are specified.

Examples

To initiate the scanning of vPlan tree in `visitor.tcl` file, use:

```
vscan -visitor visitor.tcl
```

B.2.4.6 report_metrics

The `report_metrics` command generates metrics report in an HTML format. The report generated using the `report_metrics` command includes hyperlinks to navigate through the design hierarchy and is also aligned with the reports generated from GUI.

Note: The `report_metrics` command considers the configuration option *Show extend metrics tree* while generating the report. If the *Show extend metrics tree* option is enabled (using the *Configuration* dialog box in GUI or using the CLI `config` command), then the covergroups, cover items, FSMs, and assertions are also listed in the metrics hierarchy tree in the report.

Syntax

```
report_metrics  
[-out <output_directory>]
```

```
[-overwrite]
[-title <report_title>]
[-view <view_name> [-filter <filter>]]
[-report_type regular | tabulated | extended]
[-extended true | false]
{[-summary] | [-detail [-kind expand | aggregate] [-metrics <metrics_type>]
[-source on|off]
[-exclComments]
[-email <email_address>]
[-verification_scope <scope>]
[-covered | -uncovered | -both | -all | -excludes]}
[-aspect sim | formal | both]
[-assertionStatus]
[-allAssertionCounters]
[-type |-inst]
[list]

metrics_type ::= [all] [code] [fsm] [functional] [block] [expression] [toggle]
[state] [transition] [arc] [covergroup] [assertion]
```

where

- `-out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestep>` directory.

In the absence of the `-out` option, by default, a directory named `html_<timestep>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestep>` directory.

Note: A top-level summary page `index.html` is used to navigate through the HTML reports available in the `html_<timestep>` directory.

With the `-out` option, the command redirects the HTML output to `<output_directory>`. If `<output_directory>` includes just the name of the output directory, then the command creates the output directory named `<output_directory>` in the current working directory, and stores the HTML report files along with the top-level summary page in `<output_directory>`. For example, if `<output_directory>` is specified as:

```
report_metrics -out day1
```

then directory named `day1` is created in the current working directory to store HTML report files.

Note: If `<output_directory>` already exists, the output directory is not overwritten. You must use the `-overwrite` option to overwrite an existing output directory.

If `<output_directory>` includes the complete path, then the command creates a directory by the name mentioned at the lowest level in the path, and stores the report files at that location. For example, if `<output_directory>` is specified as:

```
report_metrics -out data/myreports/day1
```

then directory named `day1` is created under `data/myreports` to store report files. In this case, the path (`data/myreports`) must exist. The directory `day1` must not exist, as it will be created and not overwritten unless the `-overwrite` option is used.

- `-overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Metrics Report*.
- `-view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All Metrics* is used to generate the report. For details on how to create a view in GUI, see [Defining and Organizing Views](#) on page 43.

For example, if a view named `Mike` is already defined and you want to generate a report using the view `Mike`, use the following command:

```
report_metrics -view Mike
```

- `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

```
[attribute] : [operator] [value]
```

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `=`, `!=`, range, `~`, `!~`}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Usage example:

```
report_metrics -view <view name> -filter <filter> [...other report flags]
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the `attribute_name` specified with the `-filter` option does not exist.

- `-report_type` allows you to specify the type of bin-level report to be generated. It can be any of the following:
 - Regular: This is the default report. It lists nodes and sub-nodes and provides hyperlinks for navigating further. To generate a regular report, use:

```
report_metrics -report_type regular
```

- ❑ Tabulated: This generates a tabulated bin-level report. This report shows the *Metrics* tree and bins details in tables side-by-side. To generate a bin-level tabulated report, use:

```
report_metrics -report_type tabulated
```

- ❑ Extended: This generates an extended report. This report contains a single table in which the *Metrics* tree and the bins information for each of the cover items is shown in the tree. To generate an extended report, use:

```
report_metrics -report_type extended
```

Note: In the absence of `-report_type` option, a regular report is generated.

- `-extended true | false` enables or disables extended report that is — showing the extended tree (list covergroups, cover items, FSMs, and assertions along with the types and instances) in the hierarchy tree. By default, this option is disabled. If not specified, the report will be extended according to the configuration option *Show extend metrics tree*.
- `-summary | -detail` specifies the type of report that must be generated. If not specified, `-summary` is assumed. With the `-detail` option, you can specify the metrics types for which the report must be generated. The metrics types can be specified using the `-metrics <metrics_type>` option. The `<metrics_type>` can be any, or a combination of the following:
 - ❑ `code` for printing a detailed block, expression, and toggle coverage report
 - ❑ `fsm` for printing a detailed state, transition, and arc coverage report
 - ❑ `functional` for printing a detailed assertion and covergroup coverage report
 - ❑ `block` for printing a detailed block coverage report
 - ❑ `expression` for printing a detailed expression coverage report
 - ❑ `toggle` for printing a detailed toggle coverage report
 - ❑ `state` for printing a detailed state coverage report
 - ❑ `transition` for printing a detailed transition coverage report
 - ❑ `arc` for printing a detailed arc coverage report
 - ❑ `assertion` for printing a detailed assertion coverage report
 - ❑ `covergroup` for printing a detailed covergroup coverage report
 - ❑ `overall` for printing a detailed report for all metrics
 - ❑ `all` for printing a detailed report for all metrics

Note: You can specify more than one metrics type by separating the metric types with a colon (:). If <metrics_type> is not specified, a detailed report for all metrics is generated.

For example, to generate a summary report, use:

```
report_metrics -summary
```

or

```
report_metrics
```

For example, to generate a detailed report, use:

```
report_metrics -detail
```

To generate a detailed block report, use:

```
report_metrics -detail -metrics block
```

To generate a detailed block and toggle report, use:

```
report_metrics -detail -metrics block:toggle
```

- **-kind** specifies the format in which the covergroup bin information must be printed in the HTML report. The format can be specified as any of the following:
 - **expand**—to print the bin information as a list without grouping similar bins. This is the default behavior.
 - **aggregate**—to print the cross tuple information in a tabular format after grouping similar bins.
- **-source off|on** enables or disables reporting of the source text and line numbers in the detailed report. By default, this information is printed in the report. To disable printing of source text and line numbers, set the **-source** option to **off**.

Note: The **-source** option is relevant only with the **-detail** option. Currently, the batch mode cannot read the compressed files, and therefore, the *Source Code* column of the report of entities from the compressed files will be blank (even if the **-source** option is turned on).

- **-email <email_address>** specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- **-verification_scope <scope>** specifies the verification scope to be used for generating the report. In the absence of this option, the **default** scope is used. You can add more scopes by specifying an attribute in the vsif file.
- **-exclComments** enables reporting of exclusion comment in the detailed report page of the HTML report. By default, this information is not printed in the report. When you use this option in the `report_metrics` command, an additional column, `Exclusion`

User, Reviewer, Comment is printed in the detailed report page, which shows the exclusion comment information as:

[<user>-<reviewer>] : <comment>

where

- ❑ <user> is the login ID of the user.
- ❑ <reviewer> is the name of the exclusion reviewer. The exclusion reviewer is specified using the -reviewer option of the exclude command. If no reviewer is specified at the time of exclusion, the default value unknown is considered as the reviewer. For more details on the exclude command, see [exclude](#) on page 417.
- ❑ <comment> is the exclusion comment. The exclusion comment is specified using the -comment option of the exclude command. For more details on the exclude command, see [exclude](#) on page 417.

For example, if the exclude command was specified as:

```
exclude -inst dtmf_recv_core.SPI_INST -block 8 -comment "This block cannot be covered" -reviewer Ben
```

and the report_metrics command was specified as:

```
report_metrics -out detail_rep1 -detail -all -inst *... -exclComments
```

then an additional column Exclusion User, Reviewer, Comment with the following value will be printed in the detailed report page:

[bensky-Ben] : This block cannot be covered

where bensky is the login ID, Ben is the exclusion reviewer, and This block cannot be covered is the exclusion comment.

If the exclusion reviewer was not specified in the exclude command, then the Exclusion User, Reviewer, Comment column will show [bensky-unknown] : This block cannot be covered.

Note: The -exclComments option enables printing of exclusion comments specified at the time of exclusion. In case exclusion comments are not specified at the time of exclusion and only the exclusion reviewer is specified, then the Exclusion User, Reviewer, Comment column will not show anything even if the -exclComments option is used.

- -all | -covered | -uncovered | -both | -excludes specifies whether to list instances/types for all, covered, uncovered, both covered and uncovered, or excluded items.

Note: If none of the options (-all, -covered, -uncovered, -both, or -excludes) is specified, -uncovered is assumed. In addition, these options are relevant only with the -detail option.

- `-aspect sim | formal | both` specifies the assertion properties to be shown in the report.

- `sim` shows only simulation assertion properties in the report.
 - `formal` shows only formal assertion properties in the report.
 - `both` shows both simulation and formal assertion properties in the report.

Note: By default, the value of the `-aspect` option is set to `sim`.

- `-assertionStatus` enables reporting of assertion status information in the report. When you use this option, an additional column, `Status` is printed in the report, which shows the status of each assertion in the specified instance or type. The status can be any of the following:

- `Passed` if the assertion finished without failure at least once, and never failed.
 - `Failed` if the assertion finished with failure at least once.
 - `Other` if the assertion never finished.

Note: If `-allAssertionCounters` option is also used with the `-assertionStatus` option, then the value `Other` in the `Status` field is further classified as any of the following:

- `Vacuous` if the assertion was vacuous pass at least once but never finished.
 - `Disabled` if the assertion transitioned to the disabled state at least once but never finished.
 - `Attempted` if the assertion was attempted at least once but never finished.
 - `Not Attempted` if the assertion was not attempted all.

Note: The `-assertionStatus` option is applicable only for assertion coverage reports.

- `-allAssertionCounters` enables reporting of `Vacuous`, `Attempt`, and `Disabled` counters in addition to the default `Finished` and `Failed` counters in the detailed report.

Note: The `-allAssertionCounters` option is applicable only if the `-abvrecordvacuous` switch is used at the time of simulation run. In case the `-abvrecordvacuous` switch is not used at simulation, then the `Vacuous`, `Attempt`, and `Disabled` columns will show `n/a`. For more details on the `-abvrecordvacuous` switch, see the *Verilog Compilation Command-Line Options* user guide.

- `-type | -inst` specifies if report should be generated only for the types hierarchy or only for the instances hierarchy. If none of these options is specified, the report is generated for instance hierarchy.
- `list` specifies the names of types or instances for which the report will be generated. The default value is `* . . .`, which matches all the top-level instances and their children. It can include wildcard characters `*` and `?`. Use `. . .` to specify all descendants of an instance. Instance names can be specified as hierarchical names, for example, `top.inst1.inst2`.

For example, to generate detailed report only for type `spi`, use:

```
report_metrics -detail -type spi
```

For example, to generate detailed report for all instances, use:

```
report_metrics -detail -inst * . . .
```

Note: The path can be prefixed with "*Verification\ Metrics*", "*Verification\ Metrics/Instances*", "*Verification\ Metrics/Types*", "*Instances*", or "*Types*". Prefix is not mandatory. However, if the *Instances* or *Types* prefix is specified, the performance of the command can be improved. For example, `report_metrics a.b.c` can be specified as:

```
report_metrics Verification\ Metrics/Instances/a.b.c
```

or

```
report_metrics Instances/a.b.c
```

Note: Wild cards (`*`, `?`, and `. . .`) are not supported in the prefix *Verification\ Metrics/Types* or *Verification\ Metrics/Instances*.

Example: Generating Instance-Based Detailed Report

Consider an example of an instance-based detailed report generated using:

```
report_metrics -out detail_rep1 -title My_Instance_Based_Report -detail -all -inst * . . .
```

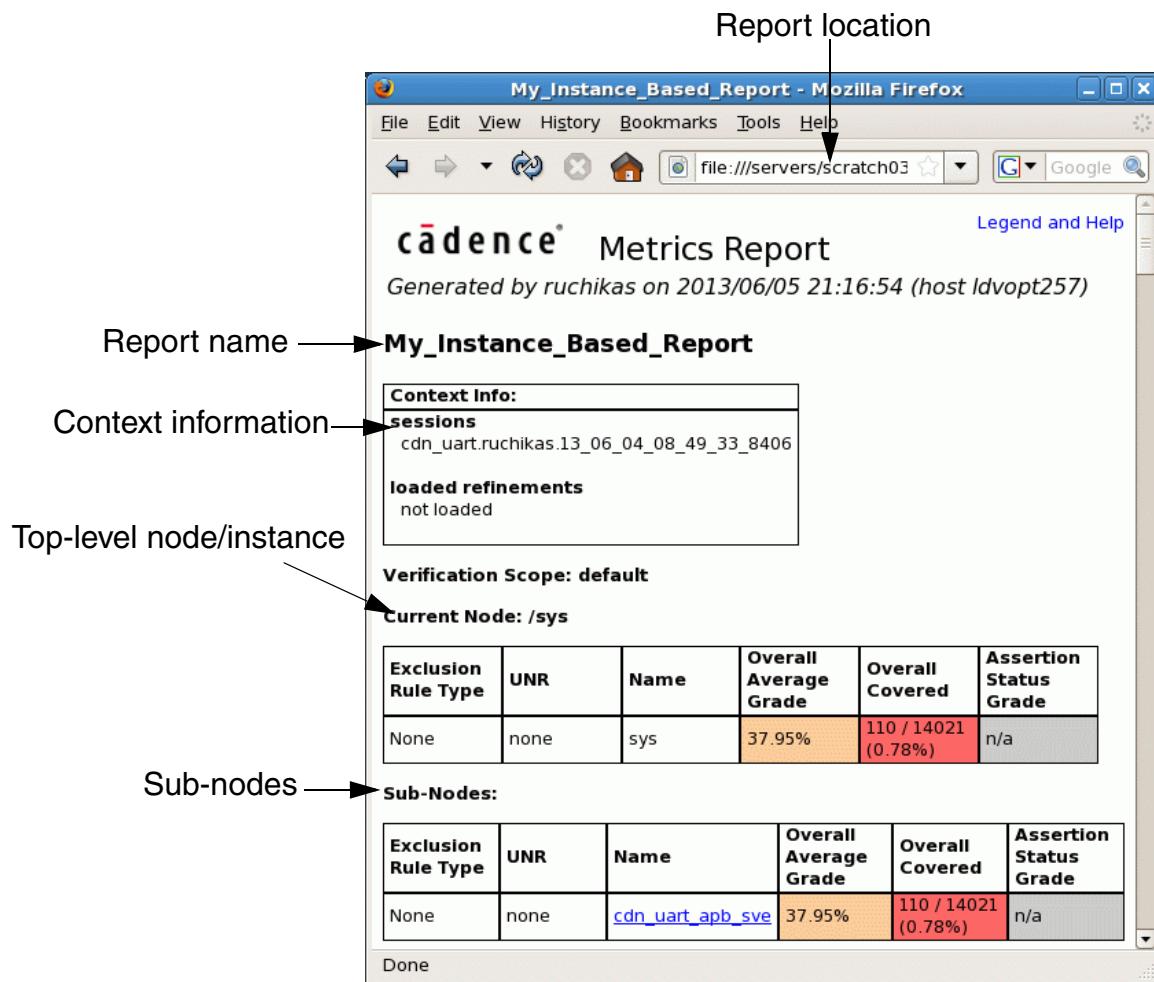
Note: The above command generates a detailed report for all metric types and for all instances. With the above command, the report is generated at:

```
<current_working_directory>/detail_rep1/index.html
```

To view the report, open `detail_rep1/index.html` in a standard web browser.

[Figure B-13](#) on page 491 shows the top-level page `index.html` in the browser.

Figure B-13 Top-Level Page (Metrics Report)



The top-level page displays information such as, who generated the report and when the report was generated. It also shows the context information, such as the loaded session and the loaded refinement file.

The other details are shown based on the options specified at the time of generating the report. For example, in this case, all the instances are listed in a tabular format because instance-based report was generated. You can sort data in the table by clicking a column header.

The report output and the columns shown in the report are based on the view selected at the time of generating the report.

You can navigate through the report using the hyperlinks available in the report.

B.2.4.7 report_sessions

The `report_sessions` command generates a session report in an HTML format. The report generated using the `report_sessions` command includes hyperlinks to navigate through the design hierarchy and is also aligned with the session report generated from the GUI.

Syntax

```
report_sessions
[-out <output_directory>]
[-overwrite]
[-title <report_title>]
[-view <view_name> [-filter <filter>]]
[-email <email_address>]
```

where

- `-out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestamp>` directory.

In the absence of the `-out` option, by default, a directory named `html_<timestamp>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestamp>` directory.

Note: A top-level summary page `index.html` is used to navigate through the HTML reports available in the `html_<timestamp>` directory.

With the `-out` option, the command redirects the HTML output to `<output_directory>`. If `<output_directory>` includes just the name of the output directory, then the command creates the output directory named `<output_directory>` in the current working directory, and stores the HTML report files along with the top-level summary page in `<output_directory>`. For example, if `<output_directory>` is specified as:

```
report_sessions -out day1
```

then directory named `day1` is created in the current working directory to store HTML report files.

Note: If `<output_directory>` already exists, the output directory is not overwritten. You must use the `-overwrite` option to overwrite an existing output directory.

If `<output_directory>` includes the complete path, then the command creates a directory by the name mentioned at the lowest level in the path, and stores the report files at that location. For example, if `<output_directory>` is specified as:

```
report_sessions -out data/myreports/day1
```

then directory named `day1` is created under `data/myreports` to store report files. In this case, the path (`data/myreports`) must exist. The directory `day1` must not exist, as it will be created and not overwritten unless the `-overwrite` option is used.

- `-overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Sessions Report*.
- `-email <email_address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *My_Sessions* is used to generate the report. For details on how to create a view in vManager GUI, see [Defining and Organizing Views](#) on page 43.

For example, if a view named `Mike` is already defined and you want to generate a report using the view `Mike`, use the following command:

```
report_sessions -view Mike
```

- `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

```
[attribute] : [operator] [value]
```

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `!=`, `range`, `~`, `!~`}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Usage example:

```
report_sessions -view <view name> -filter <filter> [...other report flags]
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the `attribute_name` specified with the `-filter` option does not exist.

Examples

Consider an example of a sessions report generated using:

```
report_sessions -out session_rep1 -view All_Sessions
```

With the above command, the session report is generated at:

<current_working_directory>/sesssion_rep1/index.html

To view the report, open sesssion_rep1/index.html in a standard web browser.

Figure B-14 on page 494 shows the top-level page index.html in the browser.

Figure B-14 Top-Level Page (Sessions Report)

Report name

Report location

View used to generate report

Sessions Report

Generated by ruchikas on 2013/06/05 21:23:46 (host ldvopt257)

Sessions Report

View Name: All_Sessions

Sorting: [Start Time (DESCENDING)]

No.	Failures	Warnings	Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner
1	0	0	completed	cdn_uart.ruchikas.13_06_04_08_56_30_6281	18	10	8	0	0	0	Tue Jun 04 09:50:39 IST 2013	ruchikas
2	0	0	completed	cdn_uart.ruchikas.13_06_04_08_49_33_8406	18	10	8	0	0	0	Tue Jun 04 09:49:33 IST 2013	ruchikas
3	0	0	completed	vm_uart_ctrl_regression.ruchikas.13_06_02_19_17_54_0101	18	7	11	0	0	0	Sun Jun 02 19:17:54 IST 2013	ruchikas
4	2	0	failed	mySession.nergaon.11_06_19_11_47_15_3538	82	38	43	0	1	0	Sun Jun 19 14:17:15 IST 2011	nergaon

Done

The report shows the information such as, who generated the report and when the report was generated. It also shows the view used to generate the report. You can sort table data by clicking on the table headers.

B.2.4.8 report_tests

The `report_tests` command generates a tests report in an HTML format. The report generated using the `report_tests` command includes hyperlinks to navigate through the design hierarchy and is also aligned with the session report generated from the GUI.

Syntax

```
report_tests  
[-out <output_directory>]  
[-overwrite]  
[-title <report_title>]  
[-view <view>]  
[-email <email_address>]
```

where

- `-out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestamp>` directory.

In the absence of the `-out` option, by default, a directory named `html_<timestamp>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestamp>` directory.

Note: A top-level summary page `index.html` is used to navigate through the HTML reports available in the `html_<timestamp>` directory.

With the `-out` option, the command redirects the HTML output to `<output_directory>`. If `<output_directory>` includes just the name of the output directory, then the command creates the output directory named `<output_directory>` in the current working directory, and stores the HTML report files along with the top-level summary page in `<output_directory>`. For example, if `<output_directory>` is specified as:

```
report_tests -out day1
```

then directory named `day1` is created in the current working directory to store HTML report files.

Note: If `<output_directory>` already exists, the output directory is not overwritten. You must use the `-overwrite` option to overwrite an existing output directory.

If `<output_directory>` includes the complete path, then the command creates a directory by the name mentioned at the lowest level in the path, and stores the report files at that location. For example, if `<output_directory>` is specified as:

```
report_tests -out data/myreports/day1
```

then directory named `day1` is created under `data/myreports` to store report files. In this case, the path (`data/myreports`) must exist. The directory `day1` must not exist, as it will be created and not overwritten unless the `-overwrite` option is used.

- `-overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.

- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Tests Report*.
- `-email <email_address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *Test_Hierarchy* is used to generate the report. For details on how to create a view in vManager GUI, see [Defining and Organizing Views](#) on page 43.

For example, if a view named `My_tests` is already defined and you want to generate a report using the view `My_tests`, use the following command:

```
report_tests -view My_tests
```

Examples

Consider an example of a tests report generated using:

```
report_tests -out tests_rep1 -view my_tests
```

With the above command, the session report is generated at:

```
<current_working_directory>/tests_rep1/index.html
```

To view the report, open `tests_rep1/index.html` in a standard web browser.

[Figure B-15](#) on page 497 shows the top-level page `index.html` in the browser.

Figure B-15 Top-Level Page (Tests Report)

The screenshot shows a Mozilla Firefox window displaying the 'Tests report' page. At the top, it says 'Generated by ruchikas on 2013/08/08 10:33:44 (host ldvopt257)'. Below that, it shows 'View Name: my_tests' with links for 'All Runs' and 'All Failures'. A section titled 'Tests Hierarchy' shows 'Current Node: /Test-Case Model' with a table:

Exclusion Rule Type	Name	Runs #	Overall Covered Grade
None	Test-Case Model	18	88.89%

Below this is a 'Sub-Nodes' table:

Exclusion Rule Type	Name	Runs #	Overall Covered Grade
None	Default	18	88.89%

A 'Coverage Color Legend' follows, showing color-coded ranges: 0, <25, <50, <75, <100, 100, n/a, Not Scored. At the bottom is a 'Done' button.

Annotations on the left side of the screenshot point to specific elements:

- An arrow points to the 'Report name' field in the URL bar with the label 'Report name'.
- An arrow points to the 'Click to view test groups' link under the 'Sub-Nodes' table with the label 'Click to view test groups'.
- An arrow points to the 'Report location' in the URL bar with the label 'Report location'.

The report shows the information such as, who generated the report and when the report was generated. It also shows the view used to generate the report. You can sort table data by clicking on the table headers. You can navigate through the report to view a list of test groups, tests, runs, and messages within each run by clicking on the available hyperlinks.

You can click the link named *All Runs* to view a flat list of runs from all of the tests. The runs are presented based on the settings of the Runs table in the selected view.

You can click the link named *All Failures* to view a flat list of failures from all runs in all of the tests. The failures are presented based on the settings of the Errors table in the selected view.

Note: In case you generated a report from a session that has formal properties, then you will see an additional link named *All Formal Properties*. You can click this link to view the view the formal properties details, as shown in [Figure B-17](#) on page 499.

Figure B-16 Tests Report (Formal Properties)

The screenshot shows the Cadence Tests Report interface. The left pane contains a summary of sessions and a tree view of tests. The right pane displays a detailed table of formal properties.

Formal properties :

No.	Name	Instance	Type	Status	Depth	Trace Status	Trace Depth	CPU Time (ms.)
1	ASSERT PASS,TRACE PASS	M.L	check	proven	n/a	covered	4	0
2	ASSERT PASS,TRACE FAIL	M.L	check	proven	n/a	unreachable	n/a	0
3	ASSERT FAIL,TRACE FAIL	M.L	check	cex	4	covered	4	12
4	ASSERT FAIL,TRACE PASS	M.L	check	cex	2	covered	2	10
5	COVER PASS	M.L	cover	covered	3	n/a	n/a	11
6	COVER FAIL	M.L	cover	unreachable	n/a	n/a	n/a	0

To start navigating and to view a list of test groups, click the link named *default*.

This will show the list of available test groups, as shown in [Figure B-17](#) on page 499.

Figure B-17 Tests Report

The screenshot shows a Mozilla Firefox browser window with the title "Tests report - Tests Tree - Mozilla Firefox". The page content is as follows:

cadence® Tests Report
Generated by ruchikas on 2013/06/05 21:32:40 (host ldvopt257)
[Back to summary](#)

Tests report - Tests Tree

Current Node: /[Test-Case Model](#)/default

Exclusion Rule Type	Name	Runs#	Overall Covered Grade
None	default	18	55.56%

Sub-Nodes:

Exclusion Rule Type	Name	Runs#	Overall Covered Grade
None	cdn_uart_apbe_tests	18	55.56%

Coverage Color Legend

0	<25	<50	<75	<100	100	n/a	Not Scored
---	-----	-----	-----	------	-----	-----	------------

Done

Click to view the tests

The above figure shows the test group `cdn_uart_apbe_tests`. To view a list of tests within this test group, click the hyperlink `cdn_uart_apbe_tests`.

This will show a list of tests along with the runs in each test, as shown in [Figure B-18](#) on page 500.

Figure B-18 Tests Report

The screenshot shows a Mozilla Firefox browser window displaying the 'Tests report - Tests Tree - Mozilla Firefox' page. The title bar reads 'Tests report - Tests Tree - Mozilla Firefox'. The address bar shows 'file:///servers/scratches/...'. The main content is titled 'cadence® Tests Report' and includes a timestamp 'Generated by ruchikas on 2013/06/05 21:32:40 (host ldvopt257)' and a link 'Back to summary'. Below this is the heading 'Tests report - Tests Tree' and the current node path 'Current Node: /Test-Case Model/default/cdn_uart_apbe_tests'. A table lists the tests:

Exclusion Rule Type	Name	Runs#	Overall Covered Grade
None	cdn_uart_apbe_tests	18	55.56%

Below the table is a section titled 'Sub-Nodes:' with another table:

Exclusion Rule Type	Name	Runs#	Overall Covered Grade
None	data_poll	10	100%
None	fill_tx_buffer	4	0%
None	test_uart	2	0%
None	data_poll_vir_seq	2	0%

A 'Coverage Color Legend' is provided at the bottom left, showing a gradient from red (0%) to green (100%), with labels '0', '<25', '<50', '<75', '<100', '100', 'n/a', and 'Not Scored'.

A horizontal progress bar at the bottom right is labeled 'Done'.

A callout line labeled 'List of tests' points to the 'fill_tx_buffer' row in the second table.

The above page shows the list of tests and the runs within each test. You can click on the hyperlinks in the Runs column and view details about runs within a test.

This will show a list of runs along with errors and messages in each run, as shown in [Figure B-19 on page 501](#).

Figure B-19 Tests Report

The screenshot shows a Mozilla Firefox window titled "Tests report - Mozilla Firefox". The page content is from a Cadence tool, specifically a "Runs Report". It was generated by "ruchikas" on "2013/06/05 21:32:41 (host ldvopt257)". A link to "Back to summary" is present. The main section is titled "Tests report" and shows "Runs of test test_uart". The table is sorted by "Start Time (DESCENDING)".

No.	Failures	Warnings	Index	Name	Status	Duration (sec.)	Top Files	Start Time
1	2	10	16	/cdn_uart_apbe_tests /test_uart	failed	8	/dv/project/avs131 /base_gather /test/install/install /kits/VerificationKit /soc_verification_lib /e_ex.lib /cdn_uart_apb /sve/tests /test_uart.e	Tue Jun 04 08:52:53 IST 2013
2	2	10	15	/cdn_uart_apbe_tests /test_uart	failed	8	/dv/project/avs131 /base_gather /test/install/install /kits/VerificationKit /soc_verification_lib /e_ex.lib /cdn_uart_apb /sve/tests /test_uart.e	Tue Jun 04 08:52:42 IST 2013

You can click on the hyperlinks in the Failures or Warnings columns to view the list of failures and warnings corresponding to the run.

B.2.4.9 report_vplan

The `report_vplan` command is used to generate a vPlan report. The report is created based on current context. The relevant sessions, vPlan, and refinement files must be loaded before executing this command.

Syntax

```
report_vplan
[-out <output_directory>]
[-overwrite]
[-title <report_title>]
[-view <view_name> [-filter <filter>]]
[-email <email_address>]
[-report_type regular | tabulated | extended]
[-extended true | false]
[-perspective {<perspective_name> | Unmapped} |-vplan_node <node> | -hierarchy
<hierarchy>]
{[-detail
    [-metrics <metrics_type>]}
```

```
[ -source on|off]
[ -exclComments]
[ -assertionStatus]
[ -covered| -uncovered| -both| -all| -excludes] ]
[-type <type> | -inst <inst>]

<hierarchy> ::= 
[<perspective_name>]/[<section_name>]*/[<port_name>]/[<instance>|<type>]

<node> ::= [<perspective_name>]/[<section_name>]*/[<port_name>]

<metrics_type> ::= [all] [code] [fsm] [functional] [block] [expression] [toggle]
[state] [transition] [arc] [covergroup] [assertion]
```

where

- `-out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestep>` directory.

In the absence of the `-out` option, by default, a directory named `html_<timestep>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestep>` directory.

Note: A top-level summary page `index.html` is used to navigate through the HTML reports available in the `html_<timestep>` directory.

With the `-out` option, the command redirects the HTML output to `<output_directory>`. If `<output_directory>` includes just the name of the output directory, then the command creates the output directory named `<output_directory>` in the current working directory, and stores the HTML report files along with the top-level summary page in `<output_directory>`. For example, if `<output_directory>` is specified as:

```
report_vplan -out day1
```

then directory named `day1` is created in the current working directory to store HTML report files.

Note: If `<output_directory>` already exists, the output directory is not overwritten. You must use the `-overwrite` option to overwrite an existing output directory.

If `<output_directory>` includes the complete path, then the command creates a directory by the name mentioned at the lowest level in the path, and stores the report files at that location. For example, if `<output_directory>` is specified as:

```
report_vplan -out data/myreports/day1
```

then directory named `day1` is created under `data/myreports` to store report files. In this case, the path (`data/myreports`) must exist. The directory `day1` must not exist, as it will be created and not overwritten unless the `-overwrite` option is used.

- `-overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.

- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *vPlan Report*.
- `-email <email_address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All_vPlan* is used to generate the report. For details on how to create a view in vManager GUI, see [Defining and Organizing Views](#) on page 43.

For example, if a view named `My_plans` is already defined and you want to generate a report using the view `My_tests`, use the following command:

```
report_vplan -view My_plans
```

- `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

[attribute] : [operator] [value]

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

Usage example:

```
report_vplan -view <view name> -filter <filter> [...other report flags]
```

Note: The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the `-filter` option does not exist.

- `-report_type` allows you to specify the type of report to be generated. It can be any of the following:
 - Regular: This is the default report. It lists nodes and sub-nodes and provides hyperlinks for navigating further. For more details, see [Example: Regular Report](#) on page 313. To generate a regular report, use:

```
report_vplan -report_type regular
```
 - Tabulated: This generates a tabulated bin-level report. This report contains *Metrics* or *vPlan* tree and bin details in tables side-by-side. For more details, see [Example:](#)

[Bin-Level Tabulated Report](#) on page 313. To generate a bin-level tabulated report, use:

```
report_vplan -report_type tabulated
```

- ❑ Extended: This generates an extended report. This report contains a single table in which the *Metrics* tree or vPlan tree and the bins information for each of the cover items is shown in the tree. For more details, see [Example: Bin-Level Extended Report](#) on page 315. To generate an extended report, use:

```
report_vplan -report_type extended
```

Note: In the absence of `-report_type` option, a regular report is generated.

- `-extended true | false` enables or disables extended report that is — showing the extended tree (list covergroups, cover items, FSMs, and assertions along with the types and instances) in the hierarchy tree. By default, this option is disabled. If not specified, the report will be extended according to the configuration option *Show extend metrics tree*.
- `-perspective {<perspective_name> | Unmapped}` specifies the perspective to be used for generating the report. The `<perspective_name>` is the name of the perspective to be used for generating the report. In case you want to use *Unmapped* perspective, use `-perspective Unmapped`. If the *Unmapped* perspective does not exist, then it is created automatically. For more details on Unmapped perspective, see [Enable Showing of Unmapped Perspective in vPlan](#) on page 281.

For example, to use Unmapped perspective for generating vPlan report, use:

```
report_vplan -perspective Unmapped -detail -inst *...
```

- `-hierarchy <hierarchy>` specifies the hierarchy in vPlan extended tree for which vPlan report must be generated. The `<hierarchy>` cannot include the wildcard characters `*`, `...`, and `?`.

Note: You can use *Unmapped* perspective in the path. If the *Unmapped* perspective does not exist, then it is created automatically. For example,

```
report_vplan -hierarchy Unmapped/a/b -detail -inst *...
```

- `-vplan_node <node>` specifies the path to the port in vPlan tree for which vPlan report must be generated. The `<node>` can include the wildcard characters `*`, `...`, and `?`.

Note: You can use *Unmapped* perspective in the path. For example,

```
report_vplan -vplan_node Unmapped/a/b -detail -inst *...
```

If the *Unmapped* perspective does not exist, then it is created automatically. However, if the wildcard characters are used instead of the perspective name, then the *Unmapped* perspective is not created automatically. For example, following command uses wildcard character and therefore, *Unmapped* perspective will not be created:

```
report_vplan -vplan_node */a/b -detail -inst *...
```

- **-inst <instance>** specifies the instance for which vPlan report must be generated. The **<instance>** can include the wildcard characters *, . . . , and ?.

Note: Special characters like [] in the instance names must be escaped with \. Else, you can use {} to avoid the use of \. For example, if the instance name is xbus0.masters[0], then specify it as:

```
-inst xbus0.masters\[0\]
```

or

```
-inst {xbus0.masters[0]}
```

- **-type <type>** specifies the module for which vPlan report must be generated. The **<type>** can include wildcard characters * and ?.

Note: You can specify only perspective, section, port, instance, or type as the selected node. To specify type or instance, you can use any of the following:

- Use the **-hierarchy** option and specify the path to the instance or type, or
 - Use both **-vplan_node** option to specify the port and **-type** or **-instance** option to specify the type or instance node under the port.
- **-detail** indicates that you want to generate a detailed vPlan report. With the **-detail** option, you can specify the metrics types for which the report must be generated. The metrics types can be specified using the **-metrics <metrics_type>** option. The **<metrics_type>** can be any, or a combination of the following:
 - code** for printing a detailed block, expression, and toggle coverage report
 - fsm** for printing a detailed state, transition, and arc coverage report
 - functional** for printing a detailed assertion and covergroup coverage report
 - block** for printing a detailed block coverage report
 - expression** for printing a detailed expression coverage report
 - toggle** for printing a detailed toggle coverage report
 - state** for printing a detailed state coverage report
 - transition** for printing a detailed transition coverage report
 - arc** for printing a detailed arc coverage report
 - assertion** for printing a detailed assertion coverage report
 - covergroup** for printing a detailed covergroup coverage report

- ❑ all for printing a detailed report for all metrics

Note: You can specify more than one metrics type by separating the metric types with a colon (:). If <metrics_type> is not specified, a vPlan detailed report for all metrics is generated.

For example, to generate a vPlan detailed block report, use:

```
report_vplan -detail -metrics block
```

To generate a vPlan detailed block and toggle report, use:

```
report_vplan -detail -metrics block:toggle
```

- -source off | on enables or disables reporting of the source text and line numbers in the report. By default, this information is printed in the report. To disable printing of source text and line numbers, set the -source option to off.
- -exclComments enables reporting of exclusion comment in the detailed report page of the HTML report. By default, this information is not printed in the report. When you use this option in the report_vplan command, an additional column, Exclusion User, Reviewer, Comment is printed in the detailed report page, which shows the exclusion comment information as:

```
[<user>-<reviewer>] : <comment>
```

where

- ❑ <user> is the login ID of the user.
- ❑ <reviewer> is the name of the exclusion reviewer. The exclusion reviewer is specified using the -reviewer option of the exclude command. If no reviewer is specified at the time of exclusion, the default value unknown is considered as the reviewer. For more details on the exclude command, see [exclude](#) on page 417.
- ❑ <comment> is the exclusion comment. The exclusion comment is specified using the -comment option of the exclude command. For more details on the exclude command, see [exclude](#) on page 417.

For example, if the exclude command was specified as:

```
exclude -inst dtmf_recv_core.SPI_INST -block 8 -comment "This block cannot be covered" -reviewer Ben
```

and the report_vplan command was specified as:

```
report_vplan -out detailvp_rep1 -detail -all -inst *... -exclComments
```

then an additional column Exclusion User, Reviewer, Comment with the following value will be printed in the detailed report page:

```
[bensky-Ben] : This block cannot be covered
```

where bensky is the login ID, Ben is the exclusion reviewer, and This block cannot be covered is the exclusion comment.

If the exclusion reviewer was not specified in the exclude command, then the Exclusion User, Reviewer, Comment column will show [bensky-unknown] : This block cannot be covered.

Note: The -exclComments option enables printing of exclusion comments specified at the time of exclusion. In case exclusion comments are not specified at the time of exclusion and only the exclusion reviewer is specified, then the Exclusion User, Reviewer, Comment column will not show anything even if the -exclComments option is used.

- -all | -covered | -uncovered | -both | -excludes specifies whether to list instances/types for all, covered, uncovered, both covered and uncovered, or excluded items.

Note: If none of the options (-all, -covered, -uncovered, -both, or -excludes) is specified, -uncovered is assumed. In addition, these options are relevant only with the -detail option.

- -assertionStatus enables reporting of assertion status information in the report. An additional column, Status is printed in the report, which shows the status of each assertion in the specified instance or type. The status can be any of the following:

- Passed if the assertion finished without failure at least once, and never failed.
- Failed if the assertion finished with failure at least once.
- Other if the assertion never finished.

Note: The -assertionStatus option is applicable only for assertion coverage reports.

Examples

To generate detailed vPlan report only for type spi, use:

```
report_vplan -detail -type spi
```

To generate detailed report for all instances, use:

```
report_vplan -detail -inst *...
```

Consider an example of a vPlan report generated using:

```
report_vplan -out vplan_rep1
```

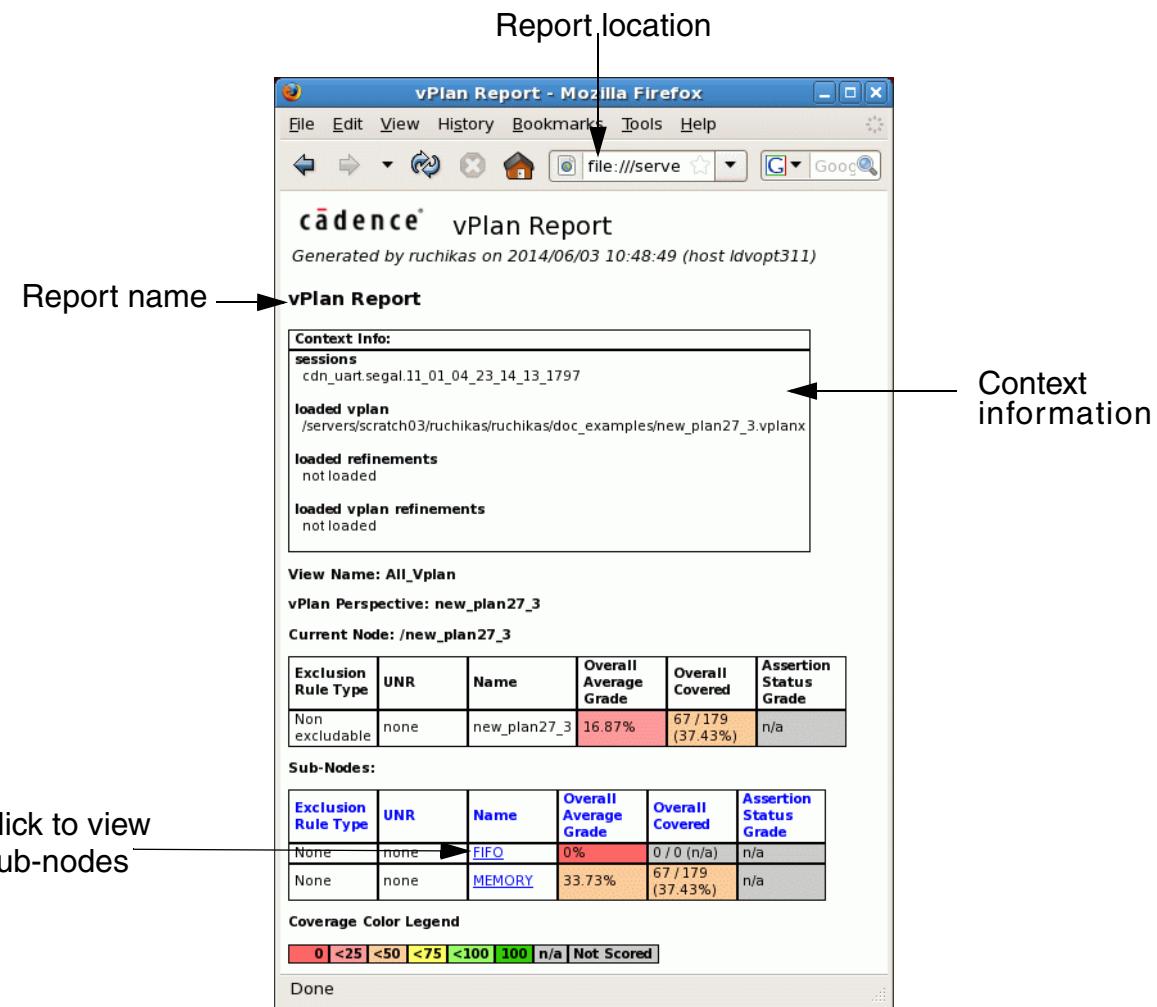
With the above command, the session report is generated at:

<current_working_directory>/vplan_rep1/

To view the report, open vplan_rep1/index.html in a standard web browser.

Figure B-20 on page 508 shows the top-level page index.html in the browser.

Figure B-20 Top-Level Page (vPlan Regular Report)



The report shows the information such as, who generated the report and when the report was generated. It also shows the context information and configuration used to generate the report. You can sort table data by clicking on the table headers. You can navigate through the report and view elements within each sub-node by clicking on the available hyperlinks.

B.2.4.10 report_tracking

The `report_tracking` command is used to generate tracking report for the selected project.

Syntax

```
report_tracking
[-out <output_directory>]
[-title <report_title>]
[-overwrite]
[-email <email_address>]
-config <tracking_config_name>
```

where

- `-out <output_directory>` redirects the report files to an alternate location instead of the default `html_<timestamp>` directory.

In the absence of the `-out` option, by default, a directory named `html_<timestamp>` is created in the current working directory, and coverage reports along with the top-level summary page (`index.html`) are stored in the `html_<timestamp>` directory.

Note: A top-level summary page `index.html` is used to navigate through the HTML reports available in the `html_<timestamp>` directory.

With the `-out` option, the command redirects the HTML output to `<output_directory>`. If `<output_directory>` includes just the name of the output directory, then the command creates the output directory named `<output_directory>` in the current working directory, and stores the HTML report files along with the top-level summary page in `<output_directory>`. For example, if `<output_directory>` is specified as:

```
report_tracking -out day1
```

then directory named `day1` is created in the current working directory to store HTML report files.

Note: If `<output_directory>` already exists, the output directory is not overwritten. You must use the `-overwrite` option to overwrite an existing output directory.

If `<output_directory>` includes the complete path, then the command creates a directory by the name mentioned at the lowest level in the path, and stores the report files at that location. For example, if `<output_directory>` is specified as:

```
report_tracking -out data/myreports/day1
```

then directory named `day1` is created under `data/myreports` to store report files. In this case, the path (`data/myreports`) must exist. The directory `day1` must not exist, as it will be created and not overwritten unless the `-overwrite` option is used.

- `-overwrite` enables overwriting of the existing output directory. By default, the output directory is not overwritten and an error is reported.
- `-email <email_address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Tracking Report*.
- `-config <tracking_config_name>` specifies the project whose report must be generated.

Examples

Consider an example of a Tracking report generated using:

```
report_tracking -out track_rep -config Tracking_Config2
```

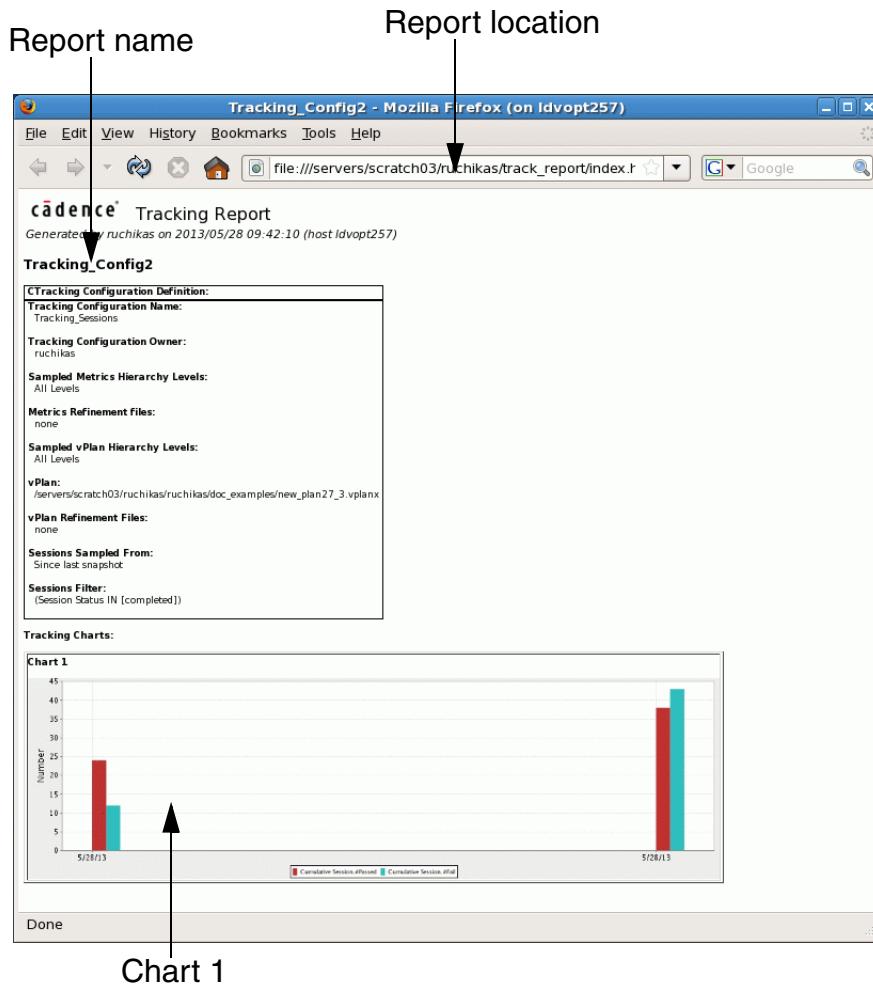
With the above command, the session report is generated at:

```
<current_working_directory>/track_report/
```

To view the report, open `track_rep/index.html` in a standard web browser.

[Figure B-21](#) on page 511 shows the top-level page `index.html` in the browser.

Figure B-21 Tracking Report



The tracking report shows information such as, who generated the report and when the report was generated. It also shows the configuration details, such as configuration name, owner, vPlan, sessions from which data is sampled and so on. It shows all the charts created for the configuration. You can scroll down and view more charts.

Note: To generate a Tracking report in GUI, see [Generating Tracking Reports](#) on page 358.

B.2.4.11 report_runs

The `report_runs` command is used to generate a runs report of the loaded sessions.

Syntax

```
report_runs  
[-out <output_file>]  
[-title <report_title>]  
[-overwrite]  
[-email <email_address>]
```

where

- `-out <output_file>` redirects the report to an alternate file instead of the default `html_<timestamp>.html` file.
In the absence of the `-out` option, by default, a file named `html_<timestamp>.html` is created in the current working directory.
Note: If `<output_file>` already exists, the output file is not overwritten. You must use the `-overwrite` option to overwrite an existing output file.
- `-overwrite` enables overwriting of the existing output file. By default, the output file is not overwritten and an error is reported.
- `-email <email_address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).
- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Runs Report*.

Examples

Consider an example of a Runs report generated using:

```
report_runs -out runs1.html
```

With the above command, the runs report is generated at:

```
<current_working_directory>/runs1.html
```

To view the report, open `runs1.html` in a standard web browser.

[Figure B-22](#) on page 513 shows the top-level page `runs1.html` in the browser.

Figure B-22 Runs Report

The screenshot shows a Mozilla Firefox browser window titled "Runs report - Mozilla Firefox". The address bar shows "file:///servers/scratch/03/ruchik". The main content area displays the "Runs Report" generated by "dan" on "2013/07/09 09:58:02 (host vlnx275)". The report includes a "Context Info" section for sessions like "vm_basic_scopes.dan.13_07_08_15_37_00_8102" and "demo_maya_tests.eyalt.11_11_29_13_39_41_2435". Below this is a table titled "Runs report" showing the following data:

No.	Name	Status	Log File
1	/basic/g1/e_test1	passed	/home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/basic/g1/run_1/specman.elog /home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/basic/g1/run_1/local_log.log /home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/vm_brun.log
2	/basic/g1/e_test1	passed	/home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/basic/g1/run_2/specman.elog; /home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/basic/g1/run_2/local_log.log; /home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/vm_brun.log
3	/basic/g1/e_test1	failed	/home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/basic/g1/run_3/specman.elog; /home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/basic/g1/run_3/local_log.log; /home/dan/vmcov_demo /vm_basic_scopes.dan.13_07_08_15_34_35_6899 /chain_0/vm_brun.log

A "List of runs" arrow points to the first row of the table. A "Report name" arrow points to the title bar. A "Report location" arrow points to the address bar.

The runs report shows information such as, who generated the report and when the report was generated. The report lists the runs in the loaded sessions. For each run, information such as, run number, name, status, and log file(s) is shown. You can click the links in the *Log File* column and view the log file.

Note: Operations such as filtering, grouping, or sorting cannot be applied on these reports.

To generate this report in GUI, see [Example: Generating Runs Report for all the Loaded Sessions](#) on page 129.

B.2.4.12report_summary

The `report_summary` command is used to generate a summary report.

A summary report is a one-page report which includes different sections to list the details of sessions, tests, metrics, and vPlan (as specified on the command line).

The summary report is generated as a single HTML file and if required, can be embedded in the email body.

Syntax

```
report_summary
[-out <output_file>]
[-title <report_title>]
[-overwrite]
[-email <email_address>]
[-title <report_title>]
[-all]
[-sessions]
[-sessions_view <view_name> [-sessions_filter <filter>]]
[-metrics]
[-metrics_view <view_name> [-metrics_filter <filter>]]
[-metrics_depth <tree_depth>]
[-tests]
[-tests_view <view_name>]
[-tests_depth <tree_depth>]
[-vplan]
[-vplan_view <view_name> [-vplan_filter <filter>]]
[-vplan_depth <tree_depth>]
[-verification_scope <scope>]
[-perspective {<perspective_name> | Unmapped}]
```

Note: If none of the options is specified with the `report_summary` command, the summary report includes all the sections and the report is generated in the current working directory.

where

- `-out <output_file>` redirects the report to an alternate file instead of the default `html_<timestamp>.html` file.
In the absence of the `-out` option, by default, a file named `html_<timestamp>.html` is created in the current working directory.
Note: If `<output_file>` already exists, the output file is not overwritten. You must use the `-overwrite` option to overwrite an existing output file.
- `-overwrite` enables overwriting of the existing output file. By default, the output file is not overwritten and an error is reported.
- `-email <email_address>` specifies the email address to which the generated report will be sent. You can specify more than one email address by separating the email addresses with a comma (,).

- `-title <report_title>` allows you to specify a title for the report. The report title cannot include blank spaces. In the absence of this option, by default, the title is specified as *Summary Report*.
 - `-all` prints all the sections (sessions, tests, metrics, and vPlan) in the summary report.
 - `-sessions` prints the Sessions section in the summary report.
 - `-sessions_view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *My_Sessions* is used to generate the report. For details on how to create a view in vManager GUI, see [Defining and Organizing Views](#) on page 43.
 - `-sessions_filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

```
[attribute] : [operator] [value]
```

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}
- For example:
- ```
-sessions_filter name:~comp
```
- You can also specify multiple filters on the command line. For example:
- ```
-sessions_filter owner:~shaim -sessions_filter status:=passed
```
- Note:** The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute_name specified with the filter does not exist.
- `-metrics` prints the Metrics section in the summary report.
- Note:** The *Metrics* section of the summary report can be configured using the `report_config` command. For more details, see [report_config](#) on page 519.
- `-metrics_view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All_Metrics* is used to generate the report. For details on how to create a view in GUI, see [Defining and Organizing Views](#) on page 43.
 - `-metrics_filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

```
[attribute] : [operator] [value]
```

where operator can be any of the: {>, <, =, <=, >=, !=, range, ~, !~}
- For example:
- ```
-metrics_filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-metrics_filter owner:~shaim -metrics_filter status:=passed
```

**Note:** The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute\_name specified with the filter does not exist.

- `-metrics_depth <tree_depth>` specifies the depth of the metrics tree to be included in the report. It can be any of the following:
  - level number — to indicate the number of levels to be included in the report.
  - `all`— to include all the levels (complete metrics tree) in the report.
- Note:** Default value is `all`.
- `-tests` prints the Tests section in the summary report.
- `-tests_view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *Test\_Hierarchy* is used to generate the report. For details on how to create a view in vManager GUI, see [Defining and Organizing Views](#) on page 43.
- `-tests_depth <tree_depth>` specifies the depth of the tests hierarchy tree to be included in the report. It can be any of the following:
  - level number — to indicate the number of levels to be included in the report.
  - `all`— to include the all levels (complete tests tree) in the report.
- Note:** Default value is `all`.
- `-vplan` prints the vPlan section in the summary report.
- Note:** The *vPlan* section of the summary report can be configured using the `report_config` command. For more details, see [report config](#) on page 519.
- `-vplan_view <view_name>` allows you to generate a report using a specific view that would have been created using vManager GUI. In the absence of this option, the default view, which is *All\_vPlan* is used to generate the report. For details on how to create a view in vManager GUI, see [Defining and Organizing Views](#) on page 43.
- `-vplan_filter <filter>` specifies filter on the specified view. The format for specifying the filter is:  

```
[attribute] : [operator] [value]
```

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `!=`, range, `~`, `!~`}

For example:

```
-vplan_filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-vplan_filter owner:~shaim -vplan_filter status:=passed
```

**Note:** The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the attribute\_name specified with the filter does not exist.

- `-vplan_depth <tree_depth>` specifies the depth of the vPlan hierarchy tree to be included in the report. It can be any of the following:
  - level number — to indicate the number of levels to be included in the report.
  - `all`— to include all levels (complete vPlan tree) in the report.
- **Note:** Default value is `all`.
- `-verification_scope <scope>` lets you specify the verification scope to be used when generating the summary report. If not specified, the default scope is used.
- `-perspective {<perspective_name> | Unmapped}` lets you specify the vPlan perspective to be used when generating the summary report. If not specified, the default vPlan perspective is used. The `<perspective_name>` is the name of the perspective to be used for generating the report. In case you want to use *Unmapped* perspective, use `-perspective Unmapped`. If the *Unmapped* perspective does not exist, then it is created automatically. For more details on Unmapped perspective, see [Enable Showing of Unmapped Perspective in vPlan](#) on page 281.

### Examples

Consider an example of a summary report generated using:

```
report_summary -out rep_summary.html
```

The above command generates the summary report `rep_summary.html` in the current working directory.

To view the report, open `rep_summary.html` in a standard web browser.

[Figure B-23](#) on page 518 shows the summary report `rep_summary.html` in the browser.

**Figure B-23 Summary Report**

The screenshot shows a Mozilla Firefox browser window with the title "Summary report - Mozilla Firefox". The address bar displays "file:///servers/scratch03/ruchikas/summary\_rep.html". The page content is a summary report generated by "ruchikas" on "2014/05/14 15:44:47 (host ldvopt310)". The report includes sections for Context Info, Sessions, Tests Hierarchy, Metrics, and vPlan. The Context Info section lists loaded vplan, loaded refinements, and loaded vplan refinements. The Sessions section shows one completed session named "cdn\_uart.segal.11\_01\_04\_23\_14\_13\_1797" with 18 total runs, 16 passed, 2 failed, 0 running, 0 waiting, and 0 other. The Tests Hierarchy section shows a tree node "Test-Case Model" with an overall average grade of 75% and 16/18 (88.89%) covered. The Metrics section shows an overall average grade of 60.43% and 2496/22500 (11.09%) covered. The vPlan section shows a new\_plan27\_3 with an overall average grade of 7.7% and 327/84 (38.1%) covered.

| No. | Session Status | Name                                  | Total Runs | #Passed | #Failed | #Running | #Waiting | #Other | Start Time                   | Owner |
|-----|----------------|---------------------------------------|------------|---------|---------|----------|----------|--------|------------------------------|-------|
| 1   | completed      | cdn_uart.segal.11_01_04_23_14_13_1797 | 18         | 16      | 2       | 0        | 0        | 0      | Wed Jan 05 02:44:13 IST 2011 | segal |

| Name              | Overall Average Grade | Overall Covered  | Test Status |
|-------------------|-----------------------|------------------|-------------|
| + Test-Case Model | 75%                   | 16 / 18 (88.89%) | 88.89%      |

| Exclusion Rule Type | UNR  | Name                   | Overall Average Grade | Overall Covered       | Assertion Status Grade |
|---------------------|------|------------------------|-----------------------|-----------------------|------------------------|
| None                | none | + Verification Metrics | 60.43%                | 2496 / 22500 (11.09%) | n/a                    |

| Exclusion Rule Type | UNR  | Name           | Overall Average Grade | Overall Covered | Assertion Status Grade |
|---------------------|------|----------------|-----------------------|-----------------|------------------------|
| Non excludable      | none | + new_plan27_3 | 7.7%                  | 327/84 (38.1%)  | n/a                    |

The summary report shows information such as, who generated the report and when the report was generated. The report includes sections, such as *Sessions*, *Tests Hierarchy*, *Metrics*, and *vPlan* (based on options specified at the time of generating the report). You can expand or collapse the hierarchy tree.

**Note:** You can configure the *vPlan* section and the *Metrics* section of the summary report using the [report\\_config](#) command.

To generate this report in GUI, see [Generating Summary Report](#) on page 51.

### B.2.4.13 report\_config

The `report_config` command allows you to configure *vPlan* section and the *Metrics* section of the summary report. You can use this command before the [report\\_summary](#) command to configure the *vPlan* and the *Metrics* section of the report, as required.

#### Syntax

```
report_config
{{{-vplan [-perspective <perspective_name> | Unmapped>]
[-vplan_node <node>]
[-type <type> | -inst <inst>] |
[-vplan_entity <path> [-entity_type {inst|type}]]} |
-metrics [-verification_scope <scope>
[-type <type> | -inst <inst> | -entity <path> [-entity_type {inst|type}]]}
[-depth <depth>]
[-clear | -print]}
```

where

- `-vplan` indicates that the report configuration is for *vPlan*.
- `-metrics` indicates that the report configuration is for metrics.

**Note:** It is mandatory to indicate whether the configuration is for *vPlan* or for metrics. Therefore, `-vPlan` or `-metrics` has to be specified in the command.

- `-perspective {<perspective_name> | Unmapped}` lets you specify the *vPlan* perspective to be used when generating the summary report. If not specified, the default *vPlan* perspective is used. The `<perspective_name>` is the name of the perspective to be used for generating the report. In case you want to use *Unmapped* perspective, use `-perspective Unmapped`. If the *Unmapped* perspective does not exist, then it is created automatically. For more details on *Unmapped* perspective, see [Enable Showing of Unmapped Perspective in vPlan](#) on page 281.
- `-vplan_node` lets you specify the root of the *vPlan* tree. In the absence of this option, the perspective's root is used.
- `-type <type>` lets you specify the type(s) for which the report must be generated. It can be a type (for metrics hierarchy) or a type mapped under the *vPlan* metrics port. If not specified, report is generated from the root node.
- `-inst <inst>` lets you specify the instance(s) for which the report must be generated. It can be an instance (for metrics hierarchy) or an instance mapped under the *vPlan* metrics port. If not specified, report is generated from the root node.
- `-vplan_entity <entity>` specifies path to entity (instance or type) in the *vPlan* tree, *vPlan* node, or mapped metrics entity for which the report must be generated.

The `-entity_type` option can be specified with the `-vplan_entity` option. Valid values for entity type are: `type` and `inst`.

- `-verification_scope <scope>` lets you specify the verification scope to be used when generating the summary report. If not specified, the default scope is used.
- `-entity <entity>` specifies path to entity (instance or type) in the metrics tree for which the report must be generated. The `-entity_type` option can be specified with the `-entity` option. Valid values for entity type are: `type` and `inst`.
- `-depth <tree_depth>` specifies the depth of the hierarchy tree to be included in the summary report. It can be any of the following:
  - level number — to indicate the number of levels to be included in the report.
  - `all`— to include all levels (complete tree) in the report.

**Note:** Default value is `all`.

- `-clear` clears the defined configuration settings.
- `-print` shows the defined configuration settings.

### *Important*

For each `report_config` command, a separate tree is printed in the summary report. For example, if you need to see the two perspectives in the report, use the following commands before the `report_summary` command:

```
report_config -vplan -perspective p1
report_config -vplan -perspective p2
```

The above commands will configure the vPlan section of the summary report such that two trees are reported: one for perspective `p1` and the other for perspective `p2`.

### **Examples**

To configure the vPlan options such that summary report is generated for `perspective1`, use:

```
report_config -vplan -perspective perspective1
```

To configure the vPlan options such that summary report is generated for instance `tb_uart` mapped under vPlan metrics port `xyz/aaa` and depth included in report is 2, use:

```
report_config -vplan -vplan_node xyz/aaa/ -inst tb_uart -depth 2
```

To configure the metrics options such that summary report is generated for type `apb_pkg`, use:

```
report_config -metrics -type apb_pkg
```

To configure the metrics options such that summary report is generated for root of default scope and depth included in the report is 2, use:

```
report_config -metrics -depth 2
```

To show the configuration defined for metrics, use:

```
report_config -metrics -print
```

To clear the configuration defined for vPlan, use:

```
report_config -vplan -clear
```

### B.2.4.14 rank

The `rank` command is used to rank the runs.

**Note:** A session must be loaded before you use the `rank` command.

#### Syntax

```
rank
-hierarchy <hierarchy> | -vplan_node <node> | -type <type> | -inst <inst> |
-entity <entity> [-entity_type <entity_type>] |
-vplan_entity <entity> | -entity_type <entity_type>
[<metrics>] | [-elements_file <elements_file>]
[-name <column_name>]
[-verification_scope <verification_scope>]
[-view <view_name>]
[-html] [-out_html <html_output_file>]
[-text] [-out_text <text_output_file>]
[-out <out_path>]
[-overwrite]
[-cost <cost_attribute>]
[-group_by <group_by_attributes>]
```

where

```
<hierarchy> ::=
[<perspective_name>] / [<section_name>]* / [<port_name>] / [<instance>|<type>] /
[<entity_name>]

<entity_name> ::= <covergroup_name> | [<covergroup_name>].<coverpoint_name> |
<fsm_name> | <assertion_name>

<node> ::= [<perspective_name>] / [<section_name>]* / [<port_name>]

<entity_type> ::= type | inst | fsm | assertion | covergroup | coveritem | coverbin
| toggle

<metrics> ::= -assertion <assertion_name> |
-covergroup <covergroup_name> |
-coveritem <covergroup_name>. <coverpoint_name> |
-coverbin <covergroup_name>. <coverpoint_name>. <coverbin_name> |
```

```
-fsm <fsm_name> |
-toggle <signal_full_name> [[<bit_specification>] [.rise | .fall]] |

<index_specification> ::= <index> | <range>
<range> ::= <index>-<index>
<bit_specification> ::= [<index>] | [<range>]
```

- **-hierarchy <hierarchy>** specifies the hierarchy in vPlan extended tree for which coverage grading must be calculated during ranking. The <hierarchy> cannot include the wildcard characters \*, . . . , and ?.

**Note:** You can use *Unmapped* perspective in the path. If the *Unmapped* perspective does not exist, then it is created automatically. For example,

```
rank -hierarchy Unmapped/a/b -inst c
```

- **-vplan\_node <node>** specifies the path to the port in vPlan tree for which coverage grading must be calculated during ranking. The <node> can include the wildcard characters \*, . . . , and ?.

**Note:** You can use *Unmapped* perspective in the path. For example,

```
rank -vplan_node Unmapped/a/b -inst c
```

If the *Unmapped* perspective does not exist, then it is created automatically. However, if the wildcard characters are used instead of the perspective name, then the *Unmapped* perspective is not created automatically. For example, following command uses wildcard character and therefore, *Unmapped* perspective will not be created:

```
rank -vplan_node */a/b -inst c
```

- **-inst <instance>** specifies the instance for which coverage grading must be calculated during ranking. The <instance> can include the wildcard characters \*, . . . , and ?.
- **-type <type>** specifies the module for which coverage grading must be calculated during ranking. The <type> can include wildcard characters \* and ?.
- **-entity <entity>** specifies path to entity (instance or type) in the metrics tree targeted for ranking. The path can be prefixed with "Verification\ Metrics", "Verification\ Metrics/Instances", "Verification\ Metrics/Types", "Instances", or "Types". Prefix is not mandatory. However, if the *Instances* or *Types* prefix is specified, the performance of the rank command can be improved. The **-entity\_type** option can be specified with the **-entity** option. Valid values for entity type are: type, inst, fsm, assertion, covergroup, coveritem, coverbin, and toggle. If entity type is not specified, the rank operation will search for the entities under all metric options and it can be an expensive operation. Using the **-entity\_type** option can improve the performance of the rank command. For examples, see [Examples: Using -entity on page 526](#).

**Note:** Wild cards (\*, ?, and ...) are not supported in the prefix *Verification\ Metrics/Types* or *Verification\ Metrics/Instances*.

- `-vplan_entity <entity>` specifies path to entity (instance or type) in the vPlan tree, vPlan node, or mapped metrics entity targeted for ranking. The `-entity_type` option can be specified with the `-vplan_entity` option. Valid values for entity type are: type, inst, fsm, assertion, covergroup, coveritem, coverbin, and toggle. If entity type is not specified, the `rank` operation will search for the entities under all metric options and it can be an expensive operation. Using the `-entity_type` option can improve the performance of the `rank` command. For examples, see [Examples: Using -vplan entity](#) on page 527.
- `<metrics>` specifies the elements for which coverage grading is calculated during ranking. The `<metrics>` can be any of the following:
  - `-assertion <assertion_name>` to specify the assertions for which coverage grading must be calculated during ranking. It can include wildcard characters \* and ?.
  - `-covergroup <covergroup_name>` to specify the covergroups for which coverage grading must be calculated during ranking. It can include wildcard characters \* and ?.
  - `-coveritem <covergroup_name>. <coverpoint_name>` to specify the coverpoints for which coverage grading must be calculated during ranking. It can include wildcard characters \* and ?.
  - `-coverbin <covergroup_name>. <coverpoint_name>. <coverbin_name>` to specify the coverpoint bins for which coverage grading must be calculated during ranking. It can include wildcard characters \* and ?.
  - `-fsm <fsm_name>` to specify the state machines for which coverage grading must be calculated during ranking. It can include wildcard characters \* and ?.
  - `-toggle <signal_full_name> [[<bit_specification>] [.rise | .fall]]` to specify the signals for which coverage grading must be calculated during ranking. The `<signal_full_name>` can include wildcard characters \* and ?. The `<bit_specification>` cannot include wildcard character ?, but it can include wildcard character \*. Optional keywords `rise` and `fall` indicate if rise transition or the fall transition must be considered during ranking.
- `-elements_file <elements_file>` to specify multiple entities for ranking. You can specify the entities to be ranked in a file `<elements_file>` and then pass this file as an argument to the `rank` command. The number of entities specified in the `<elements_file>` should not exceed 1000.

For example, you can specify the following entities in a file named `rankelements` and then pass this file as an argument to the `rank` command:

```
-inst tb_uart
-type vr_ad_reg -covergroup reg_access
-type vr_ad_reg -coveritem reg_access/kind
-type vr_ad_reg -covergroup reg_access*LINE*
```

You can then pass this file to the `rank` command as:

```
rank test1 test2 test3 test4 -elements_file rankelements
```

**Note:** The syntax for specifying items in an elements file is same as indicated above in the description of `-metrics` option.

- `-html` specifies that the ranking results must be reported in an HTML format.
- `-out_html <html_out_path>` specifies the location where ranking results in HTML format must be stored. If the `-html` option is used, then, by default, the ranking results are redirected to a file named `rank_html_<timestamp>.html` in the current working directory. You can change the default location using the `-out_html` option.
- `-text` specifies that the ranking results must be reported in a text format on the standard output.
- `-out_text <text_out_path>` specifies the location where ranking results in text format must be stored. If the `-text` option is used, then by default, the rank report in text format is shown on the standard output. You can redirect the output to a file using the `-out_text` option.
- `-out <out_path>` specifies the location where ranking results must be stored. This option cannot be used with `-out_html` or `-out_text` options. The `-out` option will be deprecated soon. If the `-out` option is used without the `-html` or the `-text` option, then the `-text` option is assumed.
- `-overwrite` enables overwriting of the existing output file. By default, the output file is not overwritten and an error is reported.
- `-name <column_name>` specifies the name of the new column that is created as a result of ranking command. This column includes the ranking results (that is the overall contribution of the run to total coverage). In the absence of this option, the name of the column is automatically generated as `cov_<num>`, where `<num>` is incremented each time you execute a `rank` command. The first `rank` command would generate the column name as `cov_1`, the next `rank` command would generate the column name as `cov_2`, and so on until you exit the session.
- `-verification_scope <scope>` specifies the verification scope to be used when ranking runs. In the absence of this option, the default scope is used.

- `-view <view_name>` allows you to rank data and apply a specific view on the results page that would have been created using vManager GUI. In the absence of this option, the default view (*Correlation\_Rank*) is applied.
- `-cost <cost_attribute>` specifies the cost to be considered while ranking runs. It can be any of following:
  - `cpu` to consider CPU time while ranking runs.
  - `simulation` to consider simulation time while ranking runs.
  - `user` to consider user time while ranking runs.
  - `system` to consider system time while ranking runs.
- `-group_by <group_by_attributes>` specifies the attributes based on which the groups will be created while ranking runs. You can specify more than one attribute by separating the attribute names with a comma ( , ).

When ranking runs, remember that:

- The `rank` action does not support functional coverage attributes `weight`, `goal`, and `at_least`. These attributes are not considered while ranking.
- The tool is now running multi-thread upon `rank` action. By default, the number of threads is set as 8 and should not be changed unless absolutely required. However, if required for any reason, you can modify it. For this, open `.mdv/internal.properties` from the location from which vManager is run and then edit the `application.threadpool.rank-thread-pool.max_pool_size` to the required size.

### Examples

To generate a rank report in which coverage grading must be on the basis of instance `tb_uart uart_dut` and the new column must be named as `rank_r1`, use the following command:

```
rank -inst tb_uart uart_dut -name rank_r1
```

To generate a rank report in HTML format for vPlan hierarchy `MEMORY/Phase2/toggle` and to redirect the output to `rank_ht.html`, use the following command:

```
rank -hierarchy MEMORY/Phase2/toggle -html -out_html /servers/scratch/rank_ht.html
```

If `a` is section, `b` is port and `c` is the instance that directly mapped to the port, then you can use the following commands:

```
rank -vplan_node a/b -inst c
```

```
rank -hierarchy a/b/c
```

```
rank -vplan_node a/b -inst c -block 1
```

If the name of the entity includes space, for example, if entity name is mic 1, use:

```
rank -vplan_node \"a/mic 1\"
```

**Note:** You must load the session and vPlan before using the `-hierarchy` and `-vplan_node` options.

To generate a rank report such that the results are grouped based on attribute name, use:

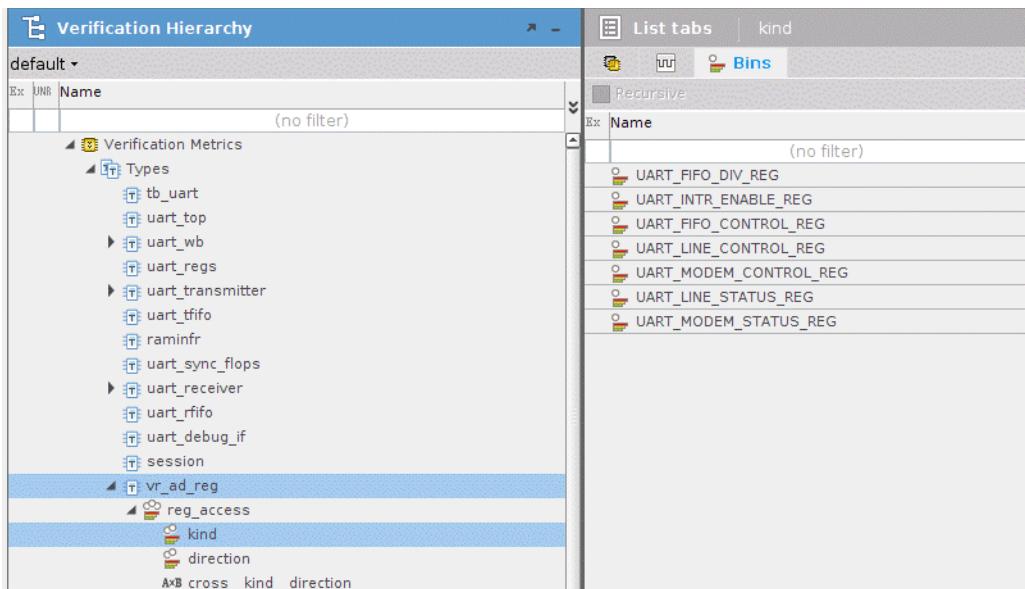
```
rank -inst uvm_pkg -group_by name
```

**Note:** No grouping is allowed after the rank or correlation operation.

### Examples: Using `-entity`

Consider the verification hierarchy in [Figure B-24](#) on page 526.

**Figure B-24 Verification Hierarchy**



To rank based on item `UART_FIFO_DIV_REG`, following `rank` commands can be used:

```
rank -entity Verification\Metrics/Types/vr_ad_reg/reg_access/kind/UART_FIFO_DIV_REG -entity_type coverbin
```

or, you can consider not specifying `Verification\ Metrics`, as shown below:

```
rank -entity Types/vr_ad_reg/reg_access/kind/UART_FIFO_DIV_REG -entity_type coverbin
```

or, you can consider not specifying *Verification\ Metrics/Types*, as shown below:

```
rank -entity vr_ad_reg/reg_access/kind/UART_FIFO_DIV_REG -entity_type coverbin
```

**Note:** Not specifying prefix *Types* or *Instances* will cause the rank operation to search the entity in both Types as well as Instances hierarchy.

or, you can consider not specifying the type of entity, as shown below:

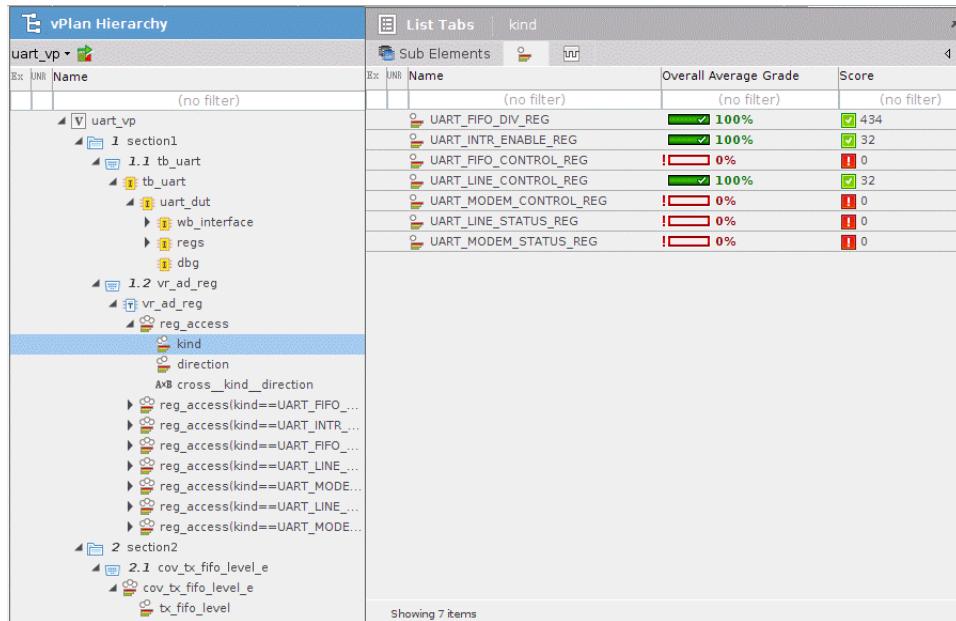
```
rank -entity Verification\
Metrics/Types/vr_ad_reg/reg_access/kind/UART_FIFO_DIV_REG
```

**Note:** Not specifying the type of entity, which in this case is `-entity_type coverbin` will cause the rank operation to search for the entities under all metric options and it can be an expensive operation.

### Examples: Using `-vplan_entity`

Consider the vPlan hierarchy in [Figure B-25](#) on page 527.

**Figure B-25 vPlan Hierarchy**



The screenshot shows the vPlan Hierarchy interface. On the left is a tree view of the vPlan structure, starting with `uart_vp`. It branches into `section1` and `section2`. `section1` further branches into `tb_uart`, `uart_dut`, `vr_ad_reg`, and `section2`. `uart_dut` contains `wb_interface`, `regs`, and `dbg`. `vr_ad_reg` contains `reg_access`, which is expanded to show `kind`, `direction`, and `cross_kind_direction`. `cross_kind_direction` contains multiple entries for `reg_access` with different kinds. `section2` contains `cov_tx_fifo_level_e`, which has `cov_tx_fifo_level_e` and `tx_fifo_level` as children. On the right is a table titled "List Tabs" with "kind" selected. The table lists various registers with their overall average grade and score. The table has columns: Ex, UNR, Name, Overall Average Grade, and Score. The data is as follows:

| Ex | UNR | Name                   | Overall Average Grade | Score       |
|----|-----|------------------------|-----------------------|-------------|
|    |     | (no filter)            | (no filter)           | (no filter) |
|    |     |                        |                       |             |
|    |     | UART_FIFO_DIV_REG      | ✓ 100%                | ✓ 434       |
|    |     | UART_INTR_ENABLE_REG   | ✓ 100%                | ✓ 32        |
|    |     | UART_FIFO_CONTROL_REG  | ! 0%                  | ! 0         |
|    |     | UART_LINE_CONTROL_REG  | ✓ 100%                | ✓ 32        |
|    |     | UART_MODEM_CONTROL_REG | ! 0%                  | ! 0         |
|    |     | UART_LINE_STATUS_REG   | ! 0%                  | ! 0         |
|    |     | UART_MODEM_STATUS_REG  | ! 0%                  | ! 0         |

To rank based on item `UART_FIFO_DIV_REG`, following rank commands can be used:

```
rank -vplan_entity
uart_vp/section1/vr_ad_reg/vr_ad_reg/reg_access/kind/UART_FIFO_DIV_REG
-entity_type coverbin
```

or, you can consider not specifying the type of entity, as shown below:

```
rank -vplan_entity
uart_vp/section1/vr_ad_reg/vr_ad_reg/reg_access/kind/UART_FIFO_DIV_REG
```

**Note:** Not specifying the type of entity, which in this case is `-entity_type coverbin` will cause the `rank` operation to search for the entities under all metric options and it can be an expensive operation.

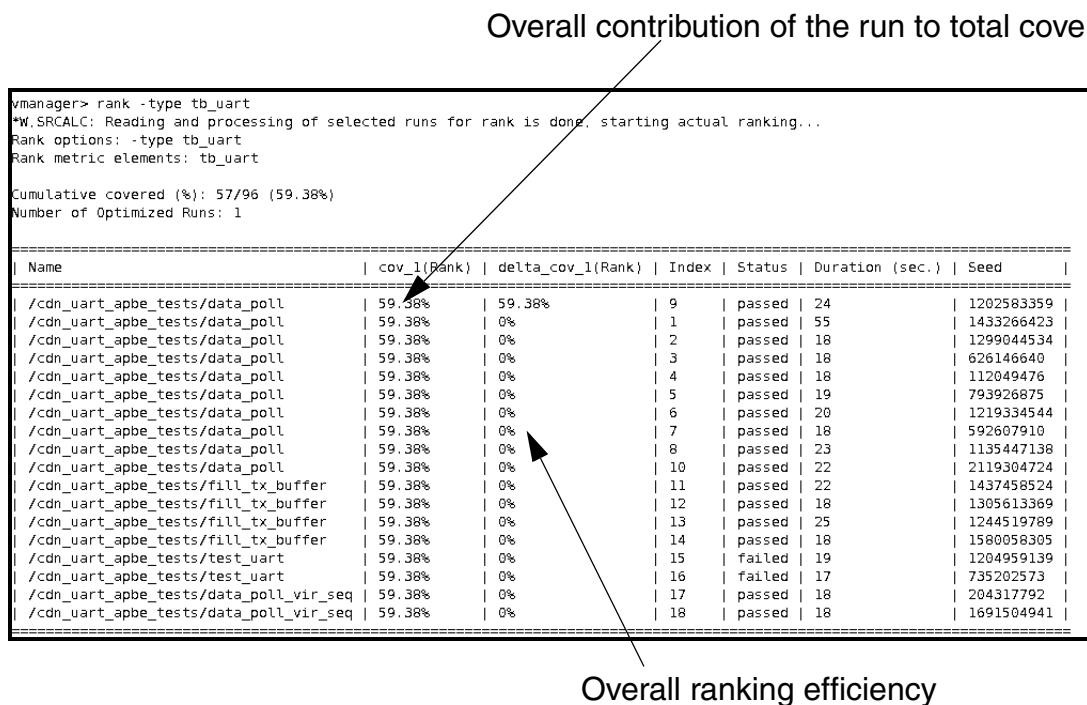
### **Example-- Generating Text Based Rank Reports**

To generate a rank report in text format in which coverage grading must be on the basis of type `tb_uart`, use the following command:

```
rank -type tb_uart
```

[Figure B-26 on page 528](#) shows output of above command.

### **Figure B-26 Output of Rank Command**



The screenshot shows the vmanager terminal output for the rank command. It starts with a summary of runs and then displays a detailed ranking table. The table has columns for Name, cov\_1(Rank), delta\_cov\_1(Rank), Index, Status, Duration (sec.), and Seed. The 'cov\_1(Rank)' column is highlighted by two arrows pointing from the text labels 'Overall contribution of the run to total coverage' and 'Overall ranking efficiency'.

| Name                                   | cov_1(Rank) | delta_cov_1(Rank) | Index | Status | Duration (sec.) | Seed       |
|----------------------------------------|-------------|-------------------|-------|--------|-----------------|------------|
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 59.38%            | 9     | passed | 24              | 1202583359 |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 1     | passed | 55              | 1433266423 |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 2     | passed | 18              | 1299044534 |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 3     | passed | 18              | 626146640  |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 4     | passed | 18              | 112049476  |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 5     | passed | 19              | 793926875  |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 6     | passed | 20              | 1219334544 |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 7     | passed | 18              | 592607910  |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 8     | passed | 23              | 1135447138 |
| /cdn_uart_apbe_tests/data_poll         | 59.38%      | 0%                | 10    | passed | 22              | 2119304724 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 59.38%      | 0%                | 11    | passed | 22              | 1437458524 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 59.38%      | 0%                | 12    | passed | 18              | 1305613369 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 59.38%      | 0%                | 13    | passed | 25              | 1244519789 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 59.38%      | 0%                | 14    | passed | 18              | 1580058305 |
| /cdn_uart_apbe_tests/test_uart         | 59.38%      | 0%                | 15    | failed | 19              | 1204959139 |
| /cdn_uart_apbe_tests/test_uart         | 59.38%      | 0%                | 16    | failed | 17              | 735202573  |
| /cdn_uart_apbe_tests/data_poll_vir_seq | 59.38%      | 0%                | 17    | passed | 18              | 204317792  |
| /cdn_uart_apbe_tests/data_poll_vir_seq | 59.38%      | 0%                | 18    | passed | 18              | 1691504941 |

The following new columns are created during ranking:

- `<name> (Rank)`—Indicates the overall contribution of the run to total coverage. `<name>` indicates the name specified with the `-name` option in the `rank` command. If the `-name` option is not used, the automatically generated name is `cov_1`.

- ***delta\_<name>* (Rank)** —Indicates the additional contribution made by the run to the overall coverage grade in comparison to the run listed above it. For example, in the above figure, the first listed run shows 59.38% and the second run shows 0%. This indicates that the second listed run contributes nothing to the overall coverage grade. For redundant runs, this column displays a 0% because there are no additional items that contribute to the coverage grade.

### Example-- Generating HTML Based Rank Reports

To generate a rank report in HTML format in which coverage grading must be on the basis of instance `tb_uart.uart_dut` and the new column must be named as `rankresult` and the output must be redirected to file named `rank_ht.html`, use the following command:

```
rank -inst tb_uart.uart_dut -html -out_html /servers/scratch03/ruchikas/ruchikas/rank_ht.html -name rankresult
```

Figure B-27 on page 529 shows output of above command.

### Figure B-27 Output of Rank Command

Overall contribution of the run to total coverage

Overall ranking efficiency

The screenshot shows a Mozilla Firefox window displaying a 'Ranking Report - Mozilla Firefox' page. The page title is 'cadence Ranking Report' and it was generated by 'ruchikas' on '2015/09/29 15:14:57 (host ldvopt311)'. The report includes a summary table and a detailed table of 10 runs. Arrows point from the text descriptions to the corresponding columns in the table.

**Summary Table (Top):**

|                           |                                                                                                           |
|---------------------------|-----------------------------------------------------------------------------------------------------------|
| Rank options:             | -inst tb_uart.uart_dut -html -out_html /servers/scratch03/ruchikas/ruchikas/rank_ht.html -name rankresult |
| Rank metric elements:     | tb_uart/uart_dut                                                                                          |
| Cumulative covered (%):   | 1068/1639 (65.16%)                                                                                        |
| Number of Optimized Runs: | 6                                                                                                         |

**Detailed Table (Bottom):**

| No. | Name                                    | rankresult(Rank) | delta_rankresult(Rank) | Index | Status | Duration (sec.) | Seed       |
|-----|-----------------------------------------|------------------|------------------------|-------|--------|-----------------|------------|
| 1   | /cdn_uart_apbe_tests/_data_poll_vir_seq | 63.76%           | 63.76%                 | 17    | passed | 18              | 204317792  |
| 2   | /cdn_uart_apbe_tests/_data_poll         | 64.43%           | 0.67%                  | 5     | passed | 19              | 793926875  |
| 3   | /cdn_uart_apbe_tests/_data_poll         | 64.86%           | 0.43%                  | 3     | passed | 18              | 626146640  |
| 4   | /cdn_uart_apbe_tests/_fill_tx_buffer    | 65.04%           | 0.18%                  | 12    | passed | 18              | 1305613369 |
| 5   | /cdn_uart_apbe_tests/_fill_tx_buffer    | 65.1%            | 0.06%                  | 13    | passed | 25              | 1244519789 |
| 6   | /cdn_uart_apbe_tests/_data_poll_vir_seq | 65.16%           | 0.06%                  | 18    | passed | 18              | 1691504941 |
| 7   | /cdn_uart_apbe_tests/_data_poll         | 65.16%           | 0%                     | 1     | passed | 55              | 1433266423 |
| 8   | /cdn_uart_apbe_tests/_data_poll         | 65.16%           | 0%                     | 2     | passed | 18              | 1299044534 |
| 9   | /cdn_uart_apbe_tests/_data_poll         | 65.16%           | 0%                     | 4     | passed | 18              | 112049476  |
| 10  | /cdn_uart_apbe_tests/_data_poll         | 65.16%           | 0%                     | 6     | passed | 20              | 1219334544 |

Done

The following new columns are created during ranking:

- *<name> (Rank)*—Indicates the overall contribution of the run to total coverage. *<name>* indicates the name specified with the `-name` option in the `rank` command. If the `-name` option is not used, the automatically generated name is `cov_1`. In this case, `rankresult` was specified with the `-name` option, therefore the column is named as `rankresult`.
- *delta\_<name> (Rank)*—Indicates the additional contribution made by the run to the overall coverage grade in comparison to the run listed above it. For example, in the above figure, the first listed run shows `63.76%` and the second run shows `0.67%`. This indicates that the second listed run contributes `0.67%` to the overall coverage grade. For redundant runs, this column displays a `0%` because there are no additional items that contribute to the coverage grade.

### B.2.4.15 correlate

The `correlate` command is used to correlate the runs.

**Note:** A session must be loaded before you use the `correlate` command.

#### Syntax

```
correlate
 -hierarchy <hierarchy> | -vplan_node <node> | -type <type> | -inst <inst> |
 -entity <entity> [-entity_type <entity_type>] |
 -vplan_entity <entity> | -entity_type <entity_type>
 [<metrics>]
 [-name <column_name>]
 [-html] [-out_html <html_output_file>]
 [-text] [-out_text <text_output_file>]
 [-out <out_path>]
 [-overwrite]
 [-overall_average_grade | -overall_covered | -assertion_status_grade]
 [-group_by <group_by_attributes>]
```

where

```
<hierarchy> ::=
[<perspective_name>] / [<section_name>]* / [<port_name>] / [<instance>|<type>] /
[<entity_name>]

<entity_name> ::= <covergroup_name> | [<covergroup_name>].<coverpoint_name> |
 <fsm_name> | <assertion_name>

<node> ::= [<perspective_name>] / [<section_name>]* / [<port_name>]

<entity_type> ::= type | inst | fsm | assertion | covergroup | coveritem | coverbin |
 toggle | block | expression | state | transition | arc

<metrics> ::= -assertion <assertion_name> |
 -covergroup <covergroup_name> |
 -coveritem <covergroup_name>. <coverpoint_name> |
 -coverbin <covergroup_name>. <coverpoint_name>. <coverbin_name> |
 -fsm <fsm_name> |
 -state <fsm_name>. <state_name> |
```

```
-transition <fsm_name>.<from_state>/|.<to_state> |
-arc <fsm_name>.<from_state>.<to_state>.<arc_index>
-toggle <signal_full_name> [<bit_specification>] [.rise | .fall] |
-block <index_specification> |
-expression <index_specification>

<index_specification> ::= <index> | <range>
<range> ::= <index>-<index>
<bit_specification> ::= [<index>] | [<range>]
```

- **-hierarchy <hierarchy>** specifies the hierarchy in vPlan extended tree for correlation operation. The <hierarchy> cannot include the wildcard characters \*, . . . , and ?.

**Note:** You can use *Unmapped* perspective in the path. If the *Unmapped* perspective does not exist, then it is created automatically. For example,

```
correlate -hierarchy Unmapped/a/b -inst c
```

- **-vplan\_node <node>** specifies the path to the port in vPlan tree for correlation operation. The <node> can include the wildcard characters \*, . . . , and ?.

**Note:** You can use *Unmapped* perspective in the path. For example,

```
correlate -vplan_node Unmapped/a/b -inst c
```

If the *Unmapped* perspective does not exist, then it is created automatically. However, if the wildcard characters are used instead of the perspective name, then the *Unmapped* perspective is not created automatically. For example, following command uses wildcard character and therefore, *Unmapped* perspective will not be created:

```
rank -vplan_node */a/b -inst c
```

- **-inst <instance>** specifies the instance for which correlation operation will be performed. The <instance> can include the wildcard characters \*, . . . , and ?.
- **-type <type>** specifies the module for which correlation operation will be performed. The <type> can include wildcard characters \* and ?.
- **-entity <entity>** specifies path to entity (instance or type) in the metrics tree for which correlation operation will be performed. The path can be prefixed with "Verification\ Metrics", "Verification\ Metrics\ Instances", "Verification\ Metrics\ Types", "Instances", or "Types". Prefix is not mandatory. However, if the *Instances* or *Types* prefix is specified, the performance of the correlate command can be improved. The **-entity\_type** option can be specified with the **-entity** option. Valid values for entity type are: type, inst, fsm, assertion, covergroup, coveritem, coverbin, and toggle. If entity type is not specified, the correlate operation will search for the entities under all metric options and it can be an expensive operation. Using the **-entity\_type** option can improve the performance of the correlate command. The examples are same as the rank command. For details, see [Examples: Using -entity](#) on page 526.

**Note:** Wild cards (\*, ?, and ...) are not supported in the prefix *Verification\ Metrics/Types* or *Verification\ Metrics/Instances*.

- `-vplan_entity <entity>` specifies path to entity (instance or type) in the vPlan tree, vPlan node, or mapped metrics entity for which correlation operation will be performed. The `-entity_type` option can be specified with the `-vplan_entity` option. Valid values for entity type are: type, inst, fsm, assertion, covergroup, coveritem, coverbin, and toggle. If entity type is not specified, the correlate operation will search for the entities under all metric options and it can be an expensive operation. Using the `-entity_type` option can improve the performance of the correlate command. The examples are same as the rank command. For details, see [Examples: Using -vplan\\_entity](#) on page 527.
- `<metrics>` specifies the elements for which correlation operation will be performed. The `<metrics>` can be any of the following:
  - `-assertion <assertion_name>` to specify the assertions for correlation operation. It can include wildcard characters \* and ?.
  - `-covergroup <covergroup_name>` to specify the covergroups for correlation operation. It can include wildcard characters \* and ?.
  - `-coveritem <covergroup_name>. <coverpoint_name>` to specify the coverpoints for correlation operation. It can include wildcard characters \* and ?.
  - `-coverbin <covergroup_name>. <coverpoint_name>. <coverbin_name>` to specify the coverpoint bins for correlation operation. It can include wildcard characters \* and ?.
  - `-fsm <fsm_name>` to specify the state machines for correlation operation. It can include wildcard characters \* and ?.
  - `-state <fsm_name>. <state_name>` to specify the states for correlation operation. It can include wildcard characters \* and ?.
  - `-transition <fsm_name>. <from_state>/ | . <to_state>` to specify the transitions for correlation operation. It can include wildcard characters \* and ?.
  - `-arc <fsm_name>. <from_state>/ | . <to_state>. <arc_index>` to specify the arcs for correlation operation. The `<arc_index>` cannot include wildcard character ?, but it can include wildcard character \*.
  - `-toggle <signal_full_name> [[<bit_specification>] [.rise | .fall]]` to specify the signals for correlation operation. The `<signal_full_name>` can include wildcard characters \* and ?. The `<bit_specification>` cannot include wildcard character ?, but it can include

wildcard character \*. Optional keywords `rise` and `fall` indicate if rise transition or the fall transition must be considered during correlation operation.

- ❑ `-block <index_specification>` to specify the blocks for correlation operation. The `<index_specification>` cannot include wildcard character ?, but it can include wildcard character \*.
- ❑ `-expression <index_specification>` to specify the expressions for correlation operation. The `<index_specification>` cannot include wildcard character ?, but it can include wildcard character \*.
- `-html` specifies that the correlation results must be reported in an HTML format.
- `-out_html <html_out_path>` specifies the location where correlation results in HTML format must be stored. If the `-html` option is used, then, by default, the correlation results are redirected to a file named `correlate_html_<timestamp>.html` in the current working directory. You can change the default location using the `-out_html` option.
- `-text` specifies that the correlation results must be reported in a text format on the standard output.
- `-out_text <text_out_path>` specifies the location where correlation results in text format must be stored. If the `-text` option is used, then by default, the correlation report in text format is shown on the standard output. You can redirect the output to a file using the `-out_text` option.
- `-out <out_path>` specifies the location where ranking results must be stored. This option cannot be used with `-out_html` or `-out_text` options. The `-out` option will be deprecated soon. If the `-out` option is used without the `-html` or the `-text` option, then the `-text` option is assumed.
- `-overwrite` enables overwriting of the existing output file. By default, the output file is not overwritten and an error is reported.
- `-name <column_name>` specifies the name of the new column that is created as a result of correlation command. This column includes the correlation results. In the absence of this option, the name of the column is automatically generated as `cov_<num>`, where `<num>` is incremented each time you execute a `correlate` command. The first `correlate` command would generate the column name as `cov_1`, the next `correlate` command would generate the column name as `cov_2`, and so on until you exit the session.
- `-view <view_name>` allows you to correlate data and apply a specific view on the results page that would have been created using vManager GUI. In the absence of this option, the default view (`Correlation_Rank`) is applied.

- -overall\_average\_grade specifies that the *Overall Average Grade* attribute will be used in the correlation operation.
  - -overall\_covered specifies that the *Overall Covered* attribute will be used in the correlation operation.
  - -assertion\_status\_grade specifies that the *Assertion Status Grade* attribute will be used in the correlation operation.
- Note:** If none of the options (-overall\_average\_grade, -overall\_covered, -assertion\_status\_grade) is specified, then the *Overall Average Grade* attribute is used in the correlation operation.
- -group\_by <group\_by\_attributes> specifies the attributes based on which the groups will be created while correlating runs. You can specify more than one attribute by separating the attribute names with a comma ( , ).

### Examples

To generate a correlation report on the basis of instance `tb_uart.uart_dut` and the new column must be named as `correlateresult`, use the following command:

```
correlate -inst tb_uart.uart_dut -html -name correlateresult
```

If a is section, b is port and c is the instance that directly mapped to the port, then you can use the following commands:

```
correlate -vplan_node a/b -inst c
correlate -hierarchy a/b/c
correlate -vplan_node a/b -inst c -block 1
```

If the name of the entity includes space, for example, if entity name is `mic 1`, use:

```
correlate -vplan_node \"a/mic 1\"
```

**Note:** You must load the session and vPlan before using the `-hierarchy` and `-vplan_node` options.

### Example-- Generating Text Based Correlate Reports

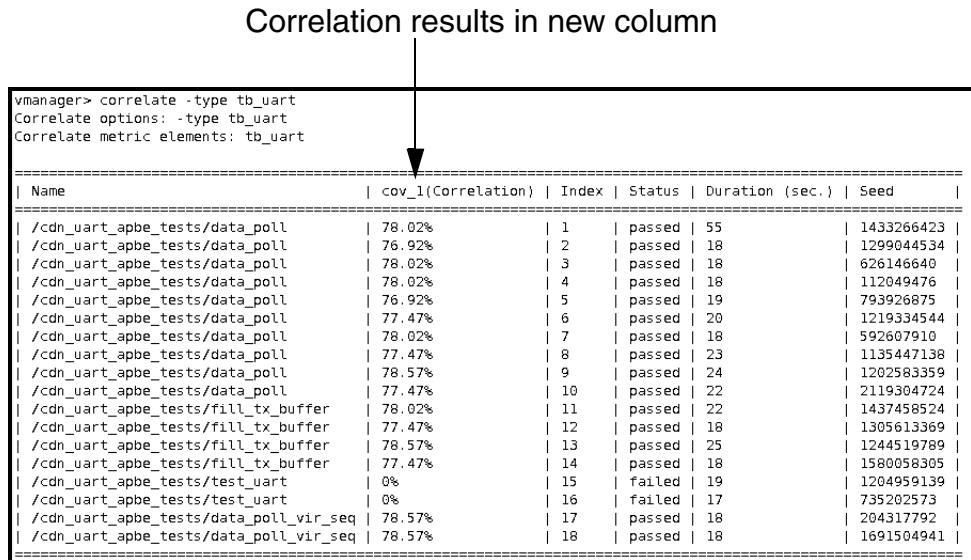
To generate a correlation report in text format in which correlation must be on the basis of type `tb_uart`, use the following command:

```
correlate -type tb_uart
```

Figure B-26 on page 528 shows output of above command.

**Figure B-28 Output of Correlate Command**

Correlation results in new column



| Name                                   | cov_1(Correlation) | Index | Status | Duration (sec.) | Seed       |
|----------------------------------------|--------------------|-------|--------|-----------------|------------|
| /cdn_uart_apbe_tests/data_poll         | 78.02%             | 1     | passed | 55              | 1433266423 |
| /cdn_uart_apbe_tests/data_poll         | 76.92%             | 2     | passed | 18              | 1299044534 |
| /cdn_uart_apbe_tests/data_poll         | 78.02%             | 3     | passed | 18              | 626146640  |
| /cdn_uart_apbe_tests/data_poll         | 78.02%             | 4     | passed | 18              | 112049476  |
| /cdn_uart_apbe_tests/data_poll         | 76.92%             | 5     | passed | 19              | 793926875  |
| /cdn_uart_apbe_tests/data_poll         | 77.47%             | 6     | passed | 20              | 1219334544 |
| /cdn_uart_apbe_tests/data_poll         | 78.02%             | 7     | passed | 18              | 592607910  |
| /cdn_uart_apbe_tests/data_poll         | 77.47%             | 8     | passed | 23              | 1135447138 |
| /cdn_uart_apbe_tests/data_poll         | 78.57%             | 9     | passed | 24              | 1202583359 |
| /cdn_uart_apbe_tests/data_poll         | 77.47%             | 10    | passed | 22              | 2119304724 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 78.02%             | 11    | passed | 22              | 1437458524 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 77.47%             | 12    | passed | 18              | 1305613369 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 78.57%             | 13    | passed | 25              | 1244519789 |
| /cdn_uart_apbe_tests/fill_tx_buffer    | 77.47%             | 14    | passed | 18              | 1580058305 |
| /cdn_uart_apbe_tests/test_uart         | 0%                 | 15    | failed | 19              | 1204959139 |
| /cdn_uart_apbe_tests/test_uart         | 0%                 | 16    | failed | 17              | 735202573  |
| /cdn_uart_apbe_tests/data_poll_vir_seq | 78.57%             | 17    | passed | 18              | 204317792  |
| /cdn_uart_apbe_tests/data_poll_vir_seq | 78.57%             | 18    | passed | 18              | 1691504941 |

When you correlate the runs, a new column is created. The column is named as *<name>* (*Correlation*), where *<name>* indicates the name specified with the *-name* option in the correlate command. If the *-name* option is not used, the automatically generated name is *cov\_1*. This column shows the contribution of the run to the attribute selected at the time of correlation.

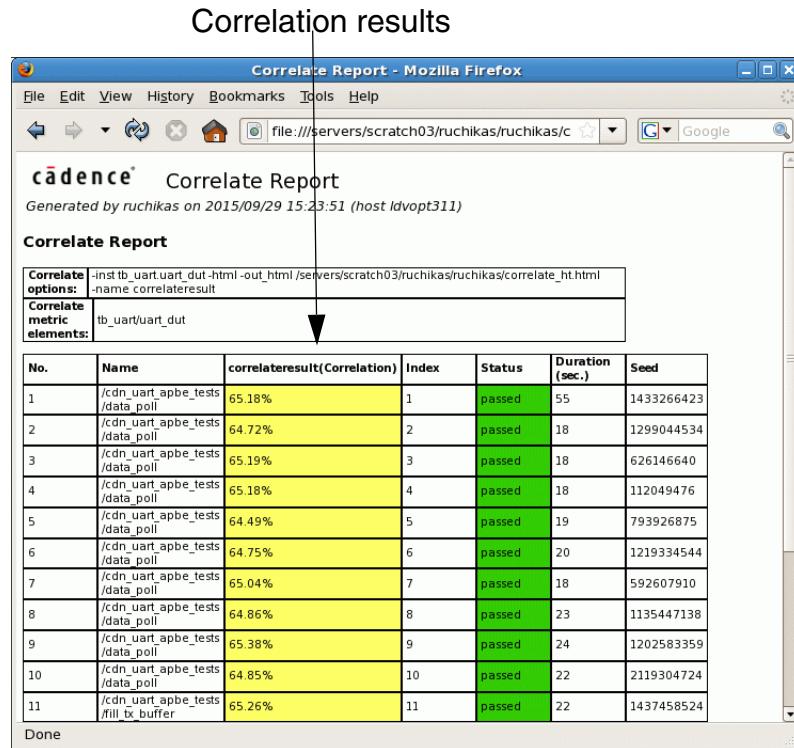
### **Example-- Generating HTML Based Correlation Reports**

To generate a correlation report in HTML format in which coverage grading must be on the basis of instance *tb\_uart.uart\_dut* and the new column must be named as *correlateresult* and the output must be redirected to file named *correlate\_ht.html*, use the following command:

```
correlate -inst tb_uart.uart_dut -html -out_html
/servers/scratch03/correlate_ht.html -name correlateresult
```

Figure B-27 on page 529 shows output of above command.

**Figure B-29 Output of Correlate Command**



When you correlate the runs, a new column is created. The column is named as *<name>* (*Correlation*), where *<name>* indicates the name specified with the *-name* option in the *correlate* command. If the *-name* option is not used, the automatically generated name is *cov\_1*. In this case, *correlateresult* was specified with the *-name* option, therefore the column is named as *correlateresult*. This column shows the contribution of the run to the attribute selected at the time of correlation.

### B.2.4.16 csv\_export

The *csv\_export* command is used to export sessions, runs, tests, vplan, and metrics tree to a CSV file.

#### Syntax

```
csv_export
-sessions | -runs | -messages | -tests | -metrics | -vplan | -formal_properties |
 -tracking
[-bins]
[-view <view_name> [-filter <filter>]]
[-depth <tree_depth>]
[-config <config_name>]
[-chart <chart_name>]
```

```
[-out <output_file>]
[-overwrite]
[-verification_scope <verification_scope>]
[-perspective <perspective_name | Unmapped>]
```

where

- `-sessions` exports all the sessions in the database to a csv file.
- `-runs` exports all the runs in the context to a csv file.
- `-messages` exports all the messages in the context to a csv file.
- `-tests` exports all the tests trees in the context to a csv file.
- `-metrics` exports the metrics tree of the context to a csv file.
- `-vplan` exports the vplan tree of the context to a csv file.
- `-formal_properties` exports the formal properties of the context to a csv file.
- `-tracking` exports the chart data of the context to a csv file.
- `-bins` specifies that the vPlan tree or metrics tree must be exported to the level of cover bins. In the absence of this option, the hierarchy tree is exported to the level of cover items.
- `-view <view_name>` allows you to export data using a specific view that would have been created using vManager GUI. In the absence of this option, the default view related to the specified session, run, test, metrics, vplan, or formal properties is used.

- `-filter <filter>` specifies filter on the specified view. The format for specifying the filter is:

```
[attribute] : [operator] [value]
```

where operator can be any of the: {`>`, `<`, `=`, `<=`, `>=`, `=`, `!=`, `range`, `~`, `!~`}

For example:

```
-filter name:~comp
```

You can also specify multiple filters on the command line. For example:

```
-filter owner:~shaim -filter status:=passed
```

**Note:** The filter specified on the command line will override the existing value in the view. In addition, an error is reported if the `attribute_name` specified with the `-filter` option does not exist.

- `-depth <tree_depth>` specifies the depth of the tree to be exported. It can be any of the following:
  - level number — to indicate the number of levels to be exported to a CSV file.

- ❑ `all`— to export all the levels (complete tree) to a CSV file.

**Note:** Default value is `all`.

- `-config <config_name>` specifies the name of the configuration to which the chart belongs.
- `-chart <chart_name>` specifies the name of the chart that has to be exported. In the absence of this option, all charts in the given configuration are exported.
- `-out <output_file>` redirects the csv output to an alternate file instead of the default:
  - ❑ `<timestampl>.csv` (in the case of sessions, tests, runs, messages, metrics, vPlan data)
  - ❑ `<chart_name>_<timestampl>.csv` (in the case of chart data)

**Note:** In case you are exporting multiple charts (that is when `-chart` option is not used or when `csv_export -tracking` command or `csv_export -tracking -config <config_name>` command is used), then by default, a folder named `<config_name>_<timestampl>` is created under the current working directory and individual charts are stored within this folder.

- `-overwrite` enables overwriting of the existing output file. By default, the output file is not overwritten and an error is reported.
- `-verification_scope <scope>` specifies the verification scope to be used when exporting data to csv file. In the absence of this option, the default scope is used.
- `-perspective {<perspective_name> | Unmapped}` specifies the perspective for the vplan tree to be used when exporting data to csv file. If not specified, the default vPlan perspective is used. The `<perspective_name>` is the name of the perspective to be used for exporting data to csv file. In case you want to use *Unmapped* perspective, use `-perspective Unmapped`. If the *Unmapped* perspective does not exist, then it is created automatically. For more details on Unmapped perspective, see [Enable Showing of Unmapped Perspective in vPlan](#) on page 281.

### Examples

To export all the sessions to a CSV file named `all_sessions.csv`, use the following command:

```
vmanager> csv_export -sessions -out all_sessions.csv
```

The above command will export all the sessions in the database to a CSV file `all_sessions.csv`.

To export the loaded vPlan to a CSV file named `vplan1.csv`, use the following command:

```
vmanager> csv_export -vplan -out vplan1.csv
```

**Note:** The above command will export the vPlan to the level of cover items. In case you want to export the vPlan to the level of cover bins, use:

```
vmanager> csv_export -vplan -bins -out vplan1.csv
```

To export all the charts of configuration conf1 to a CSV file, use the following command:

```
vmanager> csv_export -tracking -config conf1
```

The above command will create a folder named `conf1_<timestamP>` in the current working directory and save individual charts (csv files) in this folder.

To export only the chart named ch1 of configuration conf1 to a CSV file, use the following command:

```
vmanager> csv_export -tracking -config conf1 -chart ch1
```

The above command will create a file named `ch1_<timestamP>.csv` in the current working directory.

You can also export data to a CSV file using the vManager GUI. For more details, see [Exporting Table Data \(Sessions, Runs, Failures, Tests, vPlan, Metrics, Snapshots\) to a CSV File](#).

### B.2.4.17 chart\_export

The `chart_export` command is used to export the specified chart to an output file in a `.png` format. Using this command, you can also convert a regular chart to a triage chart or a stacked chart (at the time of export).

#### Syntax

```
chart_export
 [-chartid <chart_id> | -chartname <chart_name> -config <config_name>]
 -outfile <file>
 [-overwrite]
 [-type <stacked | triage>]
 [-period <days>]
```

where

- `-chartid <chart_id>` specifies the ID of the chart that has to be exported.
- `-chartname <chart_name>` specifies the name of the chart that has to be exported.
- `-config <config_name>` specifies the name of the configuration to which the chart belongs.

- `-outfile <output_file>` specifies the name and location of the `.png` file to which the chart will be exported.
  - `-overwrite` enables overwriting of the existing output file (`.png`). By default, the output file is not overwritten and an error is reported.
  - `-type <stacked | triage>` specifies the type of chart to which the regular chart will be converted at the time of export. A regular chart can be converted to a stacked chart or triage chart at the time of export. In the absence of this option, regular chart is exported.
- Note:** Not all charts can be converted to triage or stacked. Only the ones with same numeric type of attributes (for example, all attributes are numbers or percentages) can be exported to a triage or stacked chart.
- `-period <period>` specifies the number of days for which the chart has to be exported. This option is applicable only in the case of triage charts (generated using the `-type triage` option).

### Examples

To export chart named `Chart1` (that belongs to `Config1`) to `/hm/mike/ch1.png` file, use the following command:

```
vmanager> chart_export -chartname Chart1 -config Config1 -outfile /hm/mike/ch1.png
```

After you execute the above command, `Chart1` is exported to `/hm/mike/ch1.png` file.

To export chart named `Chart1` (that belongs to `Config1`) to `/hm/mike/ch1.png` file and to convert it to stacked chart at the time of export, use the following command:

```
vmanager> chart_export -chartname Chart1 -config Config1 -outfile /hm/mike/ch1.png
-overwrite -type stacked
```

After you execute the above command, `Chart1` is exported to `/hm/mike/ch1.png` file and is also converted to a stacked chart.

To export chart named `Chart1` (that belongs to `Config1`) to `/hm/mike/ch2.png` file and to convert it to triage chart at the time of export for a period of 2 days, use the following command:

```
vmanager> chart_export -chartname Chart1 -config Config1 -outfile /hm/mike/ch2.png
-overwrite -type triage -period 2
```

After you execute the above command, `Chart1` is exported to `/hm/mike/ch2.png` file and is also converted to a triage chart.

**Note:** You can also convert a regular chart to a stacked chart or a triage chart from the Web dashboard. For more details, see [Converting Regular Chart to Triage or Stacked Chart](#) on page 360.

### B.2.4.18 triage\_chart

The `triage_chart` command is used to generate a chart from selected sessions and export it to an output file in a `.png` format.

```
triage_chart
 [-chartname <chartname>]
 [-session[s] {<list_of_session_names> | <sessions_view_name>}]
 [-session_attribute <session_attribute>]
 {-runs_view <runs_view> | -group_by <attributes>}
 -outfile <file>
 [-overwrite]
 [-type {triage|stacked|colum|line}]
```

where

- `-chartname <chartname>` lets you specify the name of the chart. The name you specify will appear at the top of the chart as the heading of the chart.
- `-session[s]` lets you specify the sessions for which chart has to be generated. Sessions can be specified as:
  - `<list_of_session_names>` specifies the name(s) of the sessions for which chart has to be generated. You can specify more than one session on a single command line by separating the session names with a comma ( , ).
  - `<sessions_view_name>` specifies the name of the view that was created using vManager GUI. You can specify more than one view on a single command line by separating the view names with a comma ( , ). For more details on creating session views, see [Defining and Organizing Views](#) on page 43.

**Note:** If no sessions are specified it takes the default view, which is *My\_Sessions*.

- `-session_attribute <session_attribute>` specifies the data to be shown on the X axis of the chart. By default, *name* (the name of the session) is taken as the value to be shown on the X axis.
- `-group_by <group_by_attributes>` specifies the data to be shown on the Y axis of the chart. It specifies the attributes based on which the groups will be created while generating the chart. You can specify more than one attribute by separating the attribute names with a comma ( , ).
- `-runs_view <runs_view>` specifies the runs view to be used for generating the chart.

- `-outfile <output_file>` specifies the name and location of the .png file where the chart will be saved. For example:  
`-outfile /hm/mike/charts/ch1.png`
- `-overwrite` enables overwriting of the existing chart file (.png). By default, the output file is not overwritten and an error is reported.
- `-type <triage | stacked | column | line>` specifies the type of chart to be generated. It can be triage, stacked, column, or line. In the absence of this option, triage chart is exported.

### **Examples**

To generate a column chart from sessions view `rk_dec`, with session names on X axis and status on Y axis, and to export it to `/hm/mike/charts/ch1.png`, use the following command:

```
vmanager> triage_chart -type column -group_by status -outfile
/hm/mike/charts/ch1.png -overwrite -sessions rk_dec
```

After you execute the above command, chart named `ch1.png` is generated and saved at `/hm/mike/charts/ch1.png` file. I

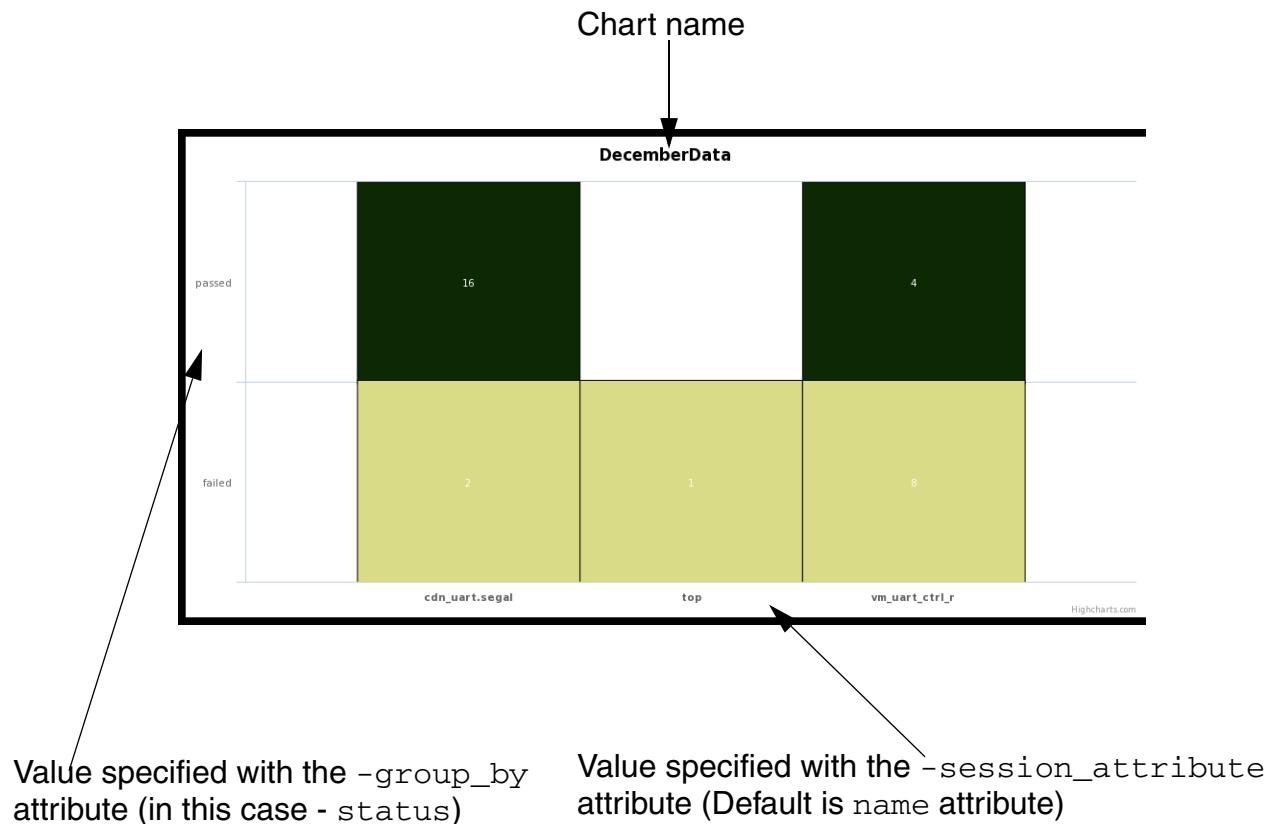
Consider the following command:

```
vmanager> triage_chart -group_by status -outfile /hm/ruchikas/charts/c4.png
-overwrite -sessions rk_dec -chartname DecemberData
```

The above command creates a column chart from sessions available in `rk_dec` view and also names the chart as `DecemberData`. In this case, as the chart type is not specified, triage chart will be generated.

Figure B-30 on page 543 shows output of above command.

**Figure B-30 Triage\_chart Command Result**



You can also generate these charts using GUI. For more details, see [Generating Charts](#).

## **Incisive vManager User Guide**

---

---

## Known Gaps in vManager

---

Incisive Enterprise Manager (IEM or eManager) and Desktop Manager (DMGR) is in the process of being replaced by a new client-server architecture, and re-named back to Incisive vManager. vManager is a functional replacement of IEM and DMGR, but not a pin-for-pin replacement, and with many more features and capabilities which users can enjoy.

Note that -- IEM will no longer be sold after October 2015. End of support for IEM is planned and scheduled for October 2017, and end of support for DMGR is planned with the end of support for the 16.1 releases of INCISIVE, as DMGR is a feature within the IES-XL Incisive Simulator.

For more information about vManager, visit our corporate website; or for a direct link to the vManager Solution page, visit:

<http://www.cadence.com/products/fv/vmanager/pages/default.aspx>

### C.1 Known Gaps in vManager in Comparison with IEM

#### C.1.1 GUI

Currently, vManager GUI does not have the following support:

- Progress bar for long operations

#### C.1.2 Runner

Currently, following runner capabilities are not supported:

- Runner Debugability Report
- DRM monitor

### C.1.3 Analysis (Runs and Coverage)

Currently, following items are not supported in the *Analysis* center:

- Formal results (ABVC) currently, not presented side-by-side with the simulation results
- Merge point, unload vPlan

### C.1.4 vPlan Limitations

Currently, following vPlan limitations exist:

- No ability to load vsif for mapping
- No support for Word template in vPlan
- No support for vPlan review capability from a web-client

### C.1.5 Ranking limitations

- Test case, refinement on raking (at\_least, weight, goal)

### C.1.6 API and CLI Compatibility

- e-based API is replaced with TCL API
- Ease of use — `vm-utils` package is fully replaced
- CLI cannot be launched from GUI

# Index

---

## A

Activity Centers [25](#)  
add columns [29](#)  
Adding Columns [29](#)  
advanced filtering [37](#)  
alias [405, 406](#)  
All Failures View [168](#)  
Analysis Center [25](#)  
    Menu Bar [135](#)  
    Toolbars [137](#)  
Analysis Center Interface [134](#)  
Analysis Navigation Panel [152](#)  
Analyzing Code Coverage [229](#)  
Analyzing Failures [165](#)  
Analyzing FSM Coverage [229](#)  
Analyzing Functional Coverage [229](#)  
Analyzing Runs [154](#)  
Apply only the valid rules [305](#)  
Attribute editing history [109, 339](#)  
Attribute Selector [29, 30](#)  
Attributes tab [227](#)  
Automatically Loading Saved  
    Refinements [303](#)  
automation [445](#)

## B

browse [431](#)

## C

chart\_export [539](#)  
Charts (Advanced) [346](#)  
Charts (Basic) [343](#)  
CLI commands  
    alias [406](#)  
    cpu [408](#)  
    exclude [418](#)  
    exit [405](#)  
    help [405](#)  
    history [405](#)  
    memory [408](#)  
    save -refinement [427](#)

sourcemap [407](#)  
unalias [406](#)  
collect [454](#)  
Collecting Runs [93](#)  
Compact Sessions [111](#)  
compressed files [228](#)  
Computing Grade of a Section [264](#)  
config [460](#)  
Configurable Items [54](#)  
Configuration [23](#)  
Configuring Rerun Options [148](#)  
Configuring Session Refresh Options [96](#)  
convert [469](#)  
convert\_icf [428](#)  
correlate command [530](#)  
correlate runs [196](#)  
Correlating Runs [195](#)  
coverage  
    total grade  
        goal-relative grade [266](#)  
        section goal [265](#)  
Coverage Coming from Palladium [235](#)  
Coverage Grading [239](#)  
CoverGroup analysis  
    View attributes [160](#)  
cpu [408](#)  
Create Context [149](#)  
Creating a Configuration [325](#)  
Creating a New vPlan [258](#)  
csv\_export [536](#)  
Cumulative Data Tracking [340](#)  
Cumulative grade [240, 248](#)  
Cumulative Grade Calculation [248](#)

## D

Data Source Pane [334](#)  
Debug\_Analyzer View [169](#)  
Defining Views [43](#)  
delete [458](#)  
Deleting a Configuration [331](#)  
Deleting Sessions [105](#)  
Deleting Views [45](#)  
detach panes [39](#)  
Detailed Analysis Page for vPlan

Entity [289](#)  
Details Pane [226, 337](#)  
Dialog Choices [20](#)  
Dirty children [307](#)  
Dirty Self [307](#)

## E

edit [466](#)  
Edit Attributes [223](#)  
Editing a Configuration [331](#)  
Editing Session Details [108](#)  
Editing vPlans [259](#)  
Enable or Disable Showing Section  
Numbers [280](#)  
exclude [417, 418](#)  
Excluding Sections or Metrics Ports [291](#)  
exit [405](#)  
export [435](#)  
export a context [18](#)  
Exporting sessions [91](#)  
Extending a vPlan visitor Interface [317](#)

## F

Filter Bar [223](#)  
filter table data [33](#)  
Filtering Date and Time Values [36](#)  
Filtering Enumeration Values [35](#)  
Filtering Numerical Values [34](#)  
Filtering Runs [184](#)  
Filtering Textual Values [34](#)  
First\_Failure View [166](#)  
formulas  
goal-relative grade [266](#)

## G

Generating Charts [342](#)  
Generating Metrics Report [249](#)  
Generating Runs Report [204](#)  
Generating Sessions Report [127](#)  
Generating Tracking Reports [358](#)  
Generating vPlan Reports [309](#)  
goal-relative grade [266, 267](#)  
Grouping by Multiple Attributes [182](#)  
Grouping by One Attribute [180](#)  
Grouping Runs [180](#)

## H

help [405](#)  
hide panes [41](#)  
history [405](#)

## I

import [434](#)  
import a context [17](#)  
Importing vsosf [89](#)  
inaccessible [107](#)  
Indago Debug Analyzer [209](#)  
INDAGO\_BATCH\_DEBUG [146](#)

## L

launch [441](#)  
Launching Analysis Center [133](#)  
Launching Metrics Analysis From  
Regression Center [217](#)  
Launching the Metrics Analysis From  
Analysis Center [218](#)  
Launching the Tracking Center [319](#)  
Launching vManager [383](#)  
List Tabs Pane [225](#)  
load [476](#)  
Loading Saved Refinements [302](#)  
Loading Sessions [87](#)  
Loading Verification Plans [251](#)  
Loading vPlan from Analysis Center [253](#)  
Loading vPlan from Regression  
Center [252](#)  
Looking for a Instance or Type [229](#)  
Looking for a Section or Port [283](#)  
Lookup drop-down [283](#)  
Lookup Enclosing Entity in Hierarchy [284](#)  
Lookup Enclosing Section in Hierarchy [284](#)  
Lookup Entity in Hierarchy [283](#)  
Lookup Entity in Metric Hierarchy [286](#)  
Lookup Port in Hierarchy [285](#)  
Lookup Section in Hierarchy [283](#)

## M

Manage Views [45](#)  
memory [408](#)

Message Popups [19](#)  
Messages Log [27](#)  
Metrics Summary Page [220](#)  
Metrics tab [226](#)

## N

Navigate Through Various Test Groups [156](#)

## O

Open Run Directory in Terminal [150](#)  
Opening Session Directory in Terminal [97](#)  
Organize Views [47](#)  
Organizing and Defining vPlan Views [261](#)

## R

rank command [521](#)  
rank runs [185](#)  
Ranking and Correlation Views [202](#)  
Ranking Runs [184](#)  
Refinement  
    Un-Exclude Excluded Items [297](#)  
Refining Metrics Data [238](#)  
Refining vPlan Attributes [298](#)  
Refining vPlan Items [290](#)  
Refreshing a Configuration [331](#)  
Refreshing Sessions [96](#)  
Registered Schemes [145](#)  
Regression Center [25](#)  
    Menu Bar [79](#)  
    Toolbars [80](#)  
    View area [154](#)  
Regression Center Interface [78](#)  
remove columns [29](#)  
Removing Attributes from Charts [354](#)  
Removing Charts [355](#)  
Removing Columns [29](#)  
Renaming Views [45](#)  
reorder columns [29](#)  
Reordering Columns [29](#)  
report [409](#)  
report\_config command [519](#)  
report\_metrics [483](#)  
report\_runs command [511](#)  
report\_sessions [492](#)

report\_summary command [513](#)  
report\_tests [494](#)  
report\_tracking [509](#)  
report\_vplan [501](#)  
Rerun [144](#)  
rerun [480](#)  
RESCAN\_LOGS [146](#)  
Reset Settings [22](#)  
resize columns [29](#)  
Resizing Columns [29](#)  
Revert vPlan Refinements [301](#)

## S

save [427](#)  
Save Settings [22](#)  
Saving Refinements [301](#)  
Saving Views [44](#)  
Scripts toolbar [71](#)  
Search Bar [224](#)  
search bar [31, 224](#)  
Searching Data in Tables [31](#)  
section goal, for total grade [265](#)  
server\_info [471](#)  
Setting a Default View [46](#)  
show [478](#)  
Show Snapshot Editing History [339](#)  
snapshot [470](#)  
sort data [31](#)  
Source tab [227](#)  
sourcemap [407](#)  
stop [468](#)  
Stopping Sessions [100](#)  
Summary page  
    Details pane [226](#)  
    Information tabs [225](#)  
    Verification Hierarchy pane [220](#)  
Synchronization Conflicts [271](#)

## T

Taking Snapshots [331](#)  
terminate\_connection [472](#)  
Toggle auto-hide [41](#)  
Toggle floating [39](#)  
toolbar [26](#)  
Tools & Options drop-down [221](#)  
Tracking Center [26, 319](#)  
    Menu Bar [321](#)

Toolbar [322](#)  
View Area [324](#)  
Tracking Pane [333](#)

## U

unalias [406](#)  
unexclude [426](#)  
Un-Excluding Excluded Items [297](#)  
Ungrouping Runs [184](#)  
unload [479](#)  
user-defined actions [71](#)

## V

Valid Modes for Launching vManager [383](#)  
Verification Hierarchy Pane [220](#)  
View Attributes and Logs Associated with  
the Selected Test or Runs [160](#)  
View Errors and Warnings Associated with  
Selected Test or Run [158](#)  
View Runs within the Selected Test or Test  
Group [157](#)  
Viewing Session Information [98](#)  
Visitor Interface [316](#)  
vManager Full Client [13](#)  
vManager Interface [14](#)  
vManager RCL Client [12](#)  
vManager window  
    Coverage view area [26](#)  
VMANGER\_INVOCATION\_MODE [383](#)  
vPlan Configuration Options [270](#)  
vPlan Extended Tree [278](#)  
vPlan Visitor Interface [316](#)  
vscan [483](#)