

# **vPlanner 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

<u>Getting Started with vPlanner</u> .....	7
<u>1.1 User Task Flow</u> .....	8
<u>1.2 vPlanner Data Flow</u> .....	10
<u>1.3 Launching vPlanner in GUI Mode</u> .....	11
<u>1.4 vPlanner User Interface</u> .....	12
<u>1.4.1 Menu Bar</u> .....	13
<u>1.4.2 Toolbar</u> .....	15
<u>1.4.3 vPlan Editor</u> .....	16
<u>1.4.4 Tabs Pane</u> .....	16
<u>1.4.5 Summary Pane</u> .....	17
<u>1.4.6 Message Pane</u> .....	19
<u>1.5 Opening a Verification Plan</u> .....	19
<u>1.5.1 Converting a Legacy Plan to a .vPlanx Format</u> .....	20
<u>1.5.2 Opening a .vplanx File At the Time of Launching vPlanner</u> .....	20
<u>1.5.3 Opening a .vplanx File After Launching vPlanner</u> .....	21

## 2

<u>Creating a Verification Plan</u> .....	23
<u>2.1 Creating a New Verification Plan</u> .....	24
<u>2.2 Creating and Editing Sections</u> .....	26
<u>2.2.1 Creating Child Sections</u> .....	26
<u>2.2.2 Creating Sibling Sections</u> .....	28
<u>2.3 Creating Coverage Ports</u> .....	29
<u>2.4 Creating Check Ports</u> .....	30
<u>2.5 Creating Testcase Ports</u> .....	31
<u>2.6 Deleting, Copying, Moving, and Renaming vPlan Elements</u> .....	32
<u>2.6.1 Deleting vPlan Elements</u> .....	32
<u>2.6.2 Copying and Pasting vPlan Elements</u> .....	32

<u>2.6.3 Moving vPlan Elements</u>	32
<u>2.6.4 Renaming vPlan Elements</u>	33
<u>2.7 Attributes Associated with a vPlan Element</u>	34
<u>    2.7.1 Editing Attributes</u>	39
<u>2.8 Specifying a User-Defined Proof Bound</u>	41
<u>2.9 Searching and Filtering Items in the vPlan Hierarchy</u>	43
<u>2.10 Creating Perspectives</u>	55
<u>    2.10.1 Example: Creating Perspective From Selection</u>	56
<u>    2.10.2 Example: Creating Filter-based Perspective</u>	57
<u>    2.10.3 Deleting Perspectives</u>	60
<u>    2.10.4 Renaming Perspectives</u>	61
<u>2.11 CSV-Based Verification Plans</u>	61
<u>    2.11.1 Creating a CSV-Based Verification Plan</u>	61
<u>    2.11.2 Embedding or Opening a CSV-Based Verification Plan in vPlanner</u>	69
<u>2.12 Creating and Using vPlan User-defined Attributes</u>	74
<u>    2.12.1 Creating User-defined Attributes</u>	75
<u>    2.12.2 Using the User-defined Attributes</u>	78
 <u>3 Annotating and Updating Verification Plans</u>	
<u>3.1 Associating Specification Document with a Plan</u>	81
<u>    3.1.1 Manipulating the Display of Document within Spec Annotator</u>	84
<u>3.2 Creating and Modifying Annotations</u>	87
<u>    3.2.1 Creating a New vPlan Element Using Selected Text</u>	88
<u>    3.2.2 Annotating an Existing vPlan Element</u>	91
<u>    3.2.3 Marking Text as Always Ignored or Ignored Until Later Version</u>	93
<u>    3.2.4 Locating Text in the PDF Specification Document</u>	94
<u>    3.2.5 Search for a Specific Text</u>	95
<u>    3.2.6 Displaying and Modifying Annotations for a vPlan Element</u>	98
<u>    3.2.7 Finding the vPlan Elements of an Annotation</u>	99
<u>    3.2.8 Navigating Between Annotations</u>	101
<u>    3.2.9 Applying Navigation Filters</u>	101
<u>    3.2.10 Displaying Annotation Statistics</u>	104
<u>3.3 Updating the Specification Document</u>	105
<u>3.4 Rich-text Editing</u>	111

<u>3.4.1 Enabling OpenOffice in vPlanner</u>	112
<u>3.4.2 Using the Rich-text Editor</u>	113
 <b>4</b>	
<b>Reusing Verification Plans</b>	117
<u>4.1 Difference Between Embedding and Referencing</u>	117
<u>4.2 Embedding a Plan in Another Plan</u>	118
<u>4.2.1 Embedding a Verification Plan</u>	118
<u>4.2.2 Synchronizing the Plan</u>	120
<u>4.2.3 Deleting Embedded Plans</u>	122
<u>4.3 Creating a Reference to Another Plan</u>	122
<u>4.3.1 Referencing a Plan</u>	122
<u>4.3.2 Reloading Referenced Plans</u>	126
<u>4.3.3 Deleting a Reference to a Plan</u>	126
<u>4.4 Making Verification Plans Reusable</u>	126
<u>4.4.1 Using Parameters</u>	127
<u>4.4.2 Using Logical Instances</u>	130
 <b>5</b>	
<b>Creating the Coverage Model</b>	143
<u>5.1 Loading Coverage Data</u>	143
<u>5.1.1 Loading Coverage Data from Run Directory</u>	143
<u>5.1.2 Loading Coverage Data from a vsdf File</u>	145
<u>5.2 Listing Specific Items in Verification Hierarchy Tree</u>	148
<u>5.3 Unloading Coverage Data</u>	149
<u>5.4 Mapping Existing Coverage Metrics to the vPlan</u>	150
<u>5.4.1 Alternate Methods to Apply Mappings</u>	153
<u>5.4.2 Mapping Multiple Coverage Metrics to a Single vPlan Item</u>	155
<u>5.4.3 Mapping Pattern</u>	157
<u>5.5 Viewing Mappings</u>	161
<u>5.6 Deleting Mappings</u>	162
<u>5.7 Searching and Filtering Coverage Items in the Metrics Tree</u>	163
<u>5.7.1 Advanced Search and Filtering</u>	165
<u>5.8 Creating Bin Filters</u>	173
<u>5.9 Using the Functional Coverage Builder</u>	180

<u>5.9.1 Importing a Testbench</u>	180
<u>5.9.2 Creating a Scalar Functional Coverage Item</u>	182
<u>5.9.3 Customizing a Functional Coverage Item</u>	186
<u>5.9.4 Designing a Cross or Transition Item</u>	188
<u>5.9.5 Designing a Per Instance Item</u>	190
<u>5.9.6 Writing e code and Regenerating Coverage Model</u>	194
 <b>6</b>	
<u>Exporting and Publishing the Plan</u>	197
<u>6.1 Exporting a Verification Plan</u>	197
<u>6.2 Viewing the Exported Plan</u>	199
<u>6.2.1 Sample of Exported Plan (HTML Format)</u>	199
<u>6.2.2 Sample of Exported Plan (CSV Format)</u>	201
<u>6.2.3 Sample of Exported Plan (Word Format)</u>	202
<u>6.2.4 Sample of Exported Plan (Perl script Format)</u>	203
<u>Index</u>	205

---

## Getting Started with vPlanner

---

vPlanner is designed to automate much of the verification planning process. Its capabilities include:

- Capturing verification plans in a common, modular, and reusable format
  - Exporting to customizable HTML, CSV, or Microsoft Word format (Chinese, Japanese, Korean fonts supported)
  - Reusing verification plans across projects or from module to system level
- Providing traceability from functional and design specs to the verification plan
  - Easily capturing, classifying, and tracking large amounts of specification data
  - Maintaining consistency between specs and the verification plan across spec revisions
- Accelerating adoption of Metric Driven Verification (MDV) methodology
  - Providing a guided process for linking verification plan features to checks and coverage metrics
  - Automating the generation of **e** functional coverage
  - Capturing functional behaviors that must be checked in the verification environment
  - Driving the verification process from plan to coverage closure through integration with vManager

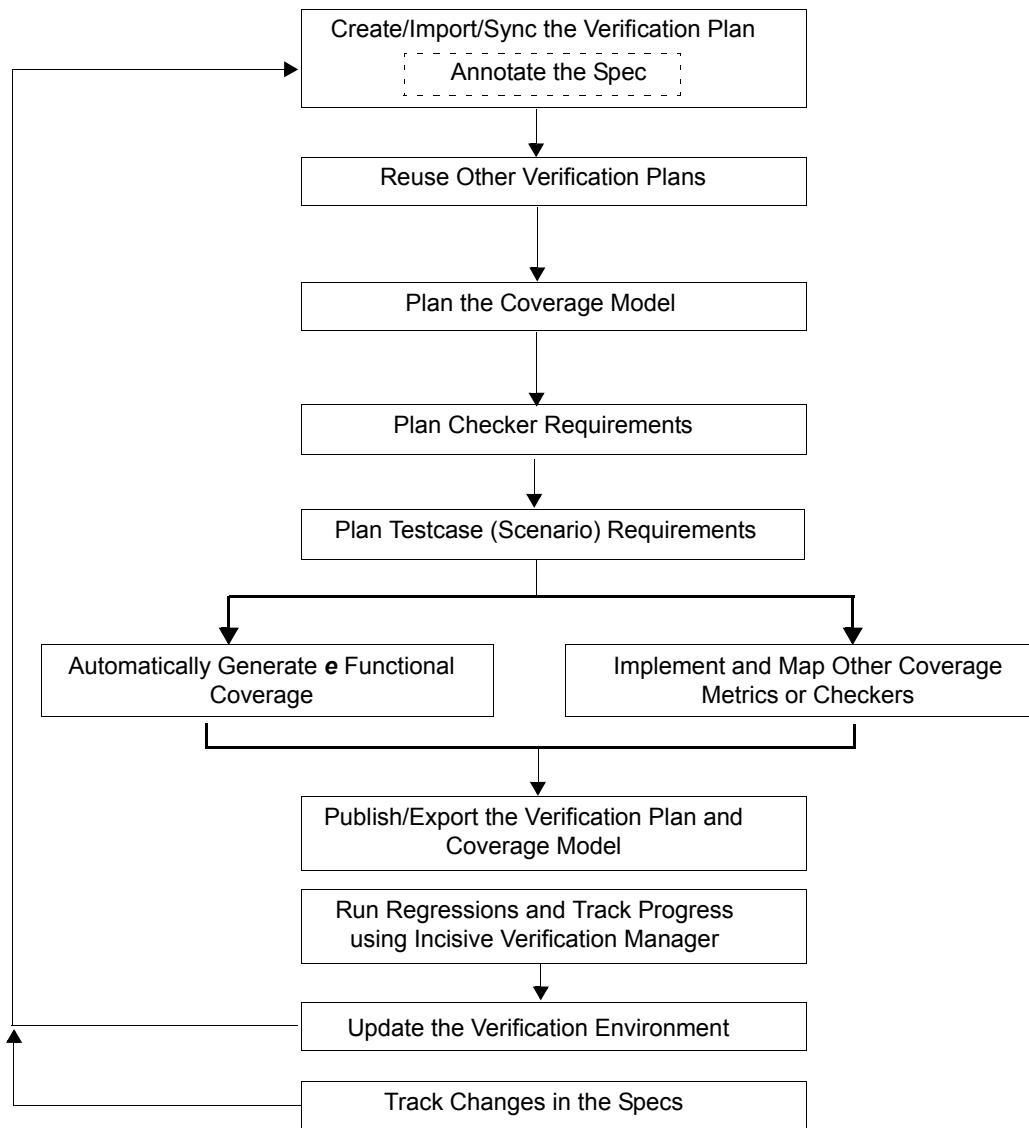
This chapter discusses the following topics:

- [User Task Flow](#)
- [vPlanner Data Flow](#)
- [Launching vPlanner in GUI Mode](#)
- [vPlanner User Interface](#)
- [Opening a Verification Plan](#)

### 1.1 User Task Flow

[Figure 1-1 on page 8](#) shows the vPlanner task flow diagram.

**Figure 1-1 vPlanner Task Flow Diagram**



The following topics describe each of these tasks in detail.

#### 1. Create a Verification Plan

When creating a verification plan, you can use either a top-down or bottom-up approach.

### Top-down Approach

The *top-down* approach is a generalized brainstorming approach that works for any product, regardless of whether written product specifications exist. In this approach, using a block diagram of the design for reference, you list two categories of features—product requirements and design requirements. In each of these categories, you list interface and core requirements. As the requirements are defined, a team member enters them into vPlanner's verification plan vPlan Editor.

### Bottom-up Approach

The bottom-up approach is a more systematic approach that focuses on the specifications for the current project. The team reviews the specifications paragraph by paragraph and identifies the features that are described. Using vPlanner, a team member selects the appropriate text in the specifications document and creates a new section or other element in the verification plan. The selected text appears in the verification plan as part of the description of the new element.

The bottom-up approach is more systematic, but it creates a verification plan whose structure reflects the functional requirements. As a result, the plan might not be easily reusable by other projects.



#### Important

For best results, use a combination of the two approaches. For example, brainstorm key features, and then annotate the specification to ensure nothing is missing from the plan.

## 2. Reuse Other Verification Plans

For complex projects, you might need a top-level plan that references a separate plan for each subcomponent. For example, in system-level verification projects, it is useful to create a system-level plan that imports the module- or block-level plans. You can use parameters to configure the referenced plans appropriately for your project.

You can also embed legacy verification plans available in .csv format.

**Note:** You can convert an old vPlan into a .vplanx format, save it as a .csv file, and then embed it. For more details on converting legacy verification plans, see [Converting a Legacy Plan to a .vPlanx Format](#) on page 20.

## 3. Plan the Coverage Model

The next task is to plan the coverage model to enable project status tracking throughout the verification process. The coverage model consists of a set of attributes that you can measure to evaluate status and quality of a specific verification activity.

### 4. Plan the Checker Requirements

You also need to identify the product behaviors that must be checked during the verification process. As with the coverage model, at the planning stage it is not necessary to decide how to implement these requirements.

### 5. (Optional) Use vPlanner to automatically generate the functional coverage model in **e**.

If the struct/unit hierarchy of the **e** testbench is defined and implemented, you can leverage the type definitions by importing the testbench into vPlanner or you can create new types in vPlanner. In any case, you can use vPlanner's Functional Coverage Builder to define scalar, cross, or transition cover items in **e**. You can then export the coverage definitions and load them on top of the testbench.

### 6. Implement other coverage metrics or checkers and map them to the plan.

Once you have defined coverage metrics or checkers in any language—including **e**, SVA, PSL, or C—you can import those definitions into vPlanner. You can also import testcases as defined in a vsif file. You can then select the planned coverage, check, or testcase ports in the verification plan and map them to appropriate metrics or checkers in the imported coverage model.

The model can be loaded either from the vManager sessions database or a standalone run directory.

### 7. Publish or export the verification plan and coverage model.

Once you have a workable verification plan, you can publish it in various formats.

### 8. Run regression tests and track their progress using vManager.

### 9. Update the verification environment and verification plan.

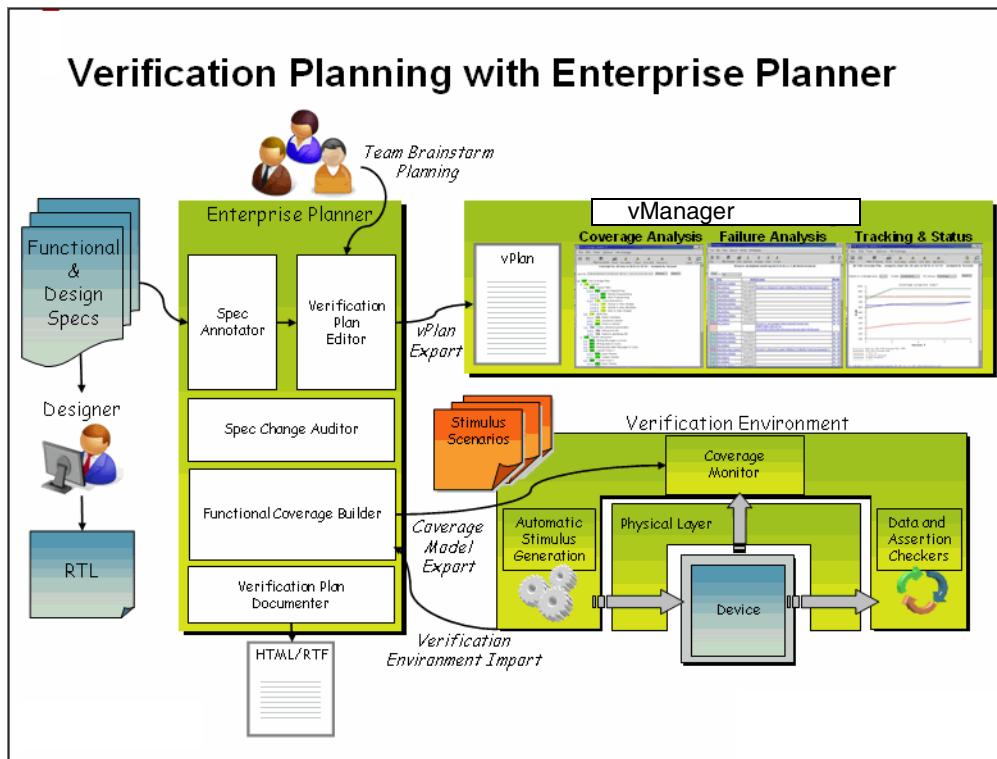
**Note:** If you update the **e** testbench or the coverage model outside of vPlanner, you must re-import it in order to keep the verification plan in sync with the coverage model and checkers.

If you have annotated the verification plan with text from the product specifications, you can use vPlanner to periodically audit the specs for changes, review each change, and update the verification plan, if necessary.

## 1.2 vPlanner Data Flow

[Figure 1-2](#) on page 11 shows the data flow between vManager, and the verification environment.

**Figure 1-2 vPlanner Data Flow**



### 1.3 Launching vPlanner in GUI Mode

To launch vPlanner, type `vplanner` at the command prompt and press `Enter`:

```
vplanner
```

You can also launch vPlanner using the following command:

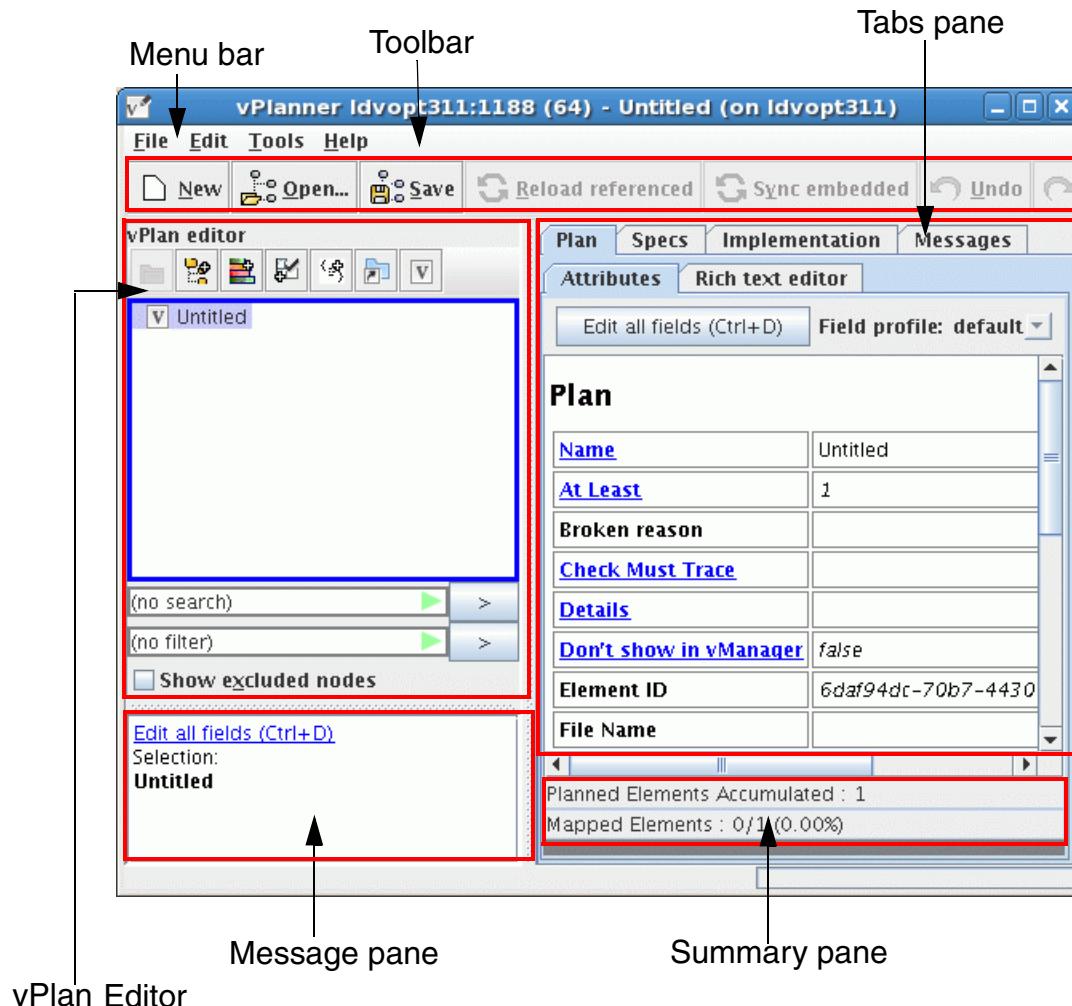
```
vmanager -planner
```

Both the above commands launch vPlanner and displays the *vPlanner* window.

**Note:** You can also launch vPlanner in local mode using the `vplanner -standalone` command.

[Figure 1-3](#) on page 12 shows the vPlanner screen.

**Figure 1-3 vPlanner User Interface**



## 1.4 vPlanner User Interface

The vPlanner window displayed in Figure 1-3 on page 12 has the following components:

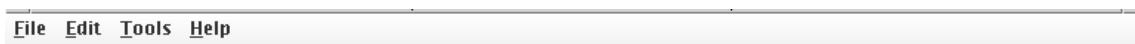
- Menu Bar
- Toolbar
- vPlan Editor
- Tabs Pane
- Summary Pane

- Message Pane

### 1.4.1 Menu Bar

The menu bar, shown in [Figure 1-4](#) on page 13, contains commands that you use to open plans, save plans, import plans, synchronize plans, and read documentation.

**Figure 1-4** Menu Bar



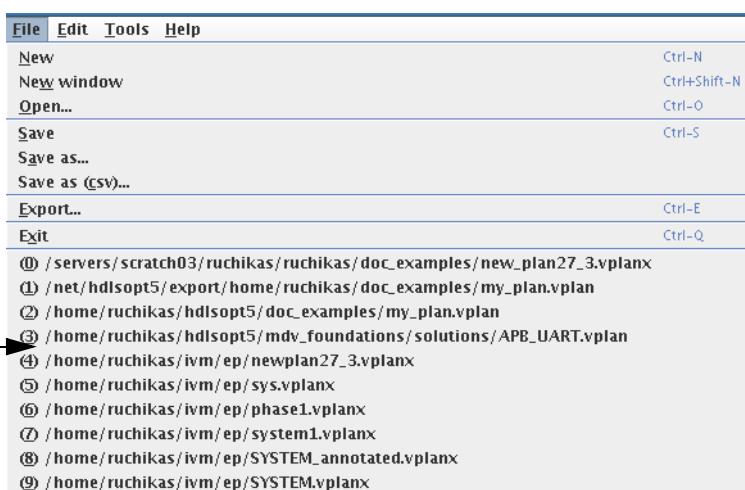
The menu bar has the following menus:

- File
- Edit
- Tools
- Help

#### 1.4.1.1 File

[Figure 1-5](#) on page 13 displays the *File* menu.

**Figure 1-5** File Menu



The *File* menu contains the commands for opening plans, saving plans, importing plans, and exiting vPlanner. You can choose the following items from the *File* menu:

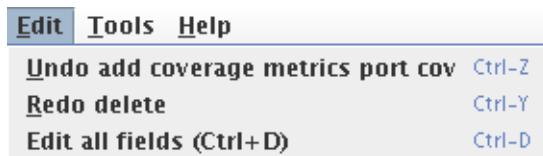
- New—To create a new plan from scratch  
For more details on creating verification plans, see [Creating a Verification Plan](#) on page 23.
- New window —To open another independent vPlanner window with a blank plan
- Open—To open an existing vPlan
- Save—To save the current plan with the same name and at the same location
- Save as —To save the current plan with a new name
- Save as (csv) —To save the current plan as a CSV file
- Export—To export the plan to HTML, CSV, or Microsoft Word format  
For more details on exporting verification plans, see [Exporting and Publishing the Plan](#) on page 197.
- Exit—To exit the tool

**Note:** The *File* menu also maintains a history of previously loaded plans (in current or previous sessions). A maximum of 10 plans are maintained and listed in the *File* menu. You can quickly load a plan by selecting it.

### 1.4.1.2 Edit

[Figure 1-6](#) on page 14 displays the *Edit* menu.

**Figure 1-6** Edit Menu



The *Edit* menu has the following options:

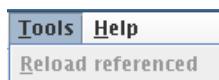
- Undo—To undo the last action performed
- Redo—To repeat the last action performed

- Edit all fields—To quickly edit all the attributes associated with the element selected in the vPlan editor. A dialog box is opened that shows all the attributes and allows you to edit their values. For more details, see [Editing Attributes](#) on page 39.

### 1.4.1.3 Tools

[Figure 1-7](#) on page 15 displays the *Tools* menu.

**Figure 1-7 Tools Menu**



The *Tools* menu has the following options:

- Reload referenced—To reload the referenced verification plan.

### 1.4.1.4 Help

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

- *vPlanner User Guide*—To display the *Incisive Verification Manager Planner User Guide*.
- *Cadence Online Support*—To open an Internet browser and launch the [www.support.cadence.com](http://www.support.cadence.com) page.
- *About vManager Planner*—To display the version and copyright information of the vPlanner.

## 1.4.2 Toolbar

[Figure 1-8](#) on page 15 displays the toolbar that appears on the *vPlanner* window.

**Figure 1-8 Toolbar**



The toolbar provides you with the following buttons:

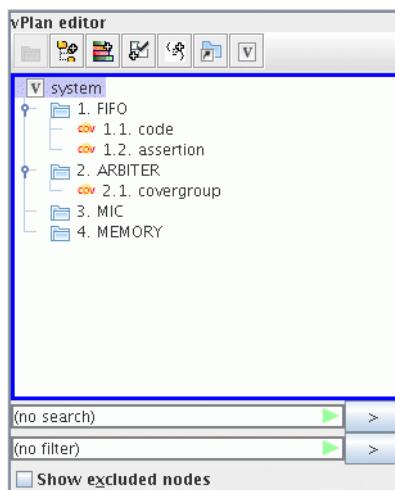
- New—To create a new verification plan

- Open—To open an existing verification plan
- Save—To save the current plan with the same name and at the same location
- Reload referenced—To reload a referenced verification plan
- Sync embedded—To synchronize the embedded verification plans
- Undo—To undo the last action performed
- Redo—To repeat the last action performed

### 1.4.3 vPlan Editor

Figure 1-9 on page 16 displays the vPlan editor that appears on the *vPlanner* window.

**Figure 1-9 vPlan Editor**



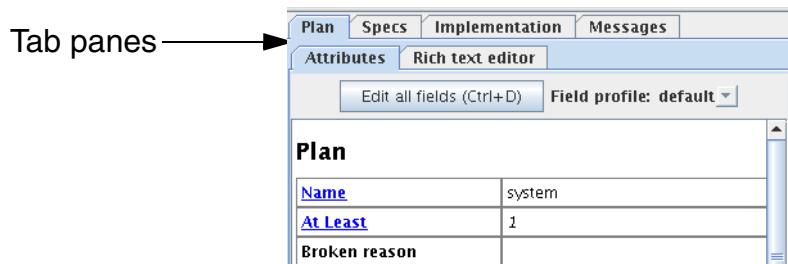
The vPlan editor lets you create, view, and modify the current hierarchy of the verification plan.

For details on creating and modifying the current verification plan hierarchy, see [Creating a Verification Plan](#) on page 23.

### 1.4.4 Tabs Pane

Figure 1-9 on page 16 displays the different tab panes that appears on the *vPlanner* window.

**Figure 1-10 vPlan Tab Panes**



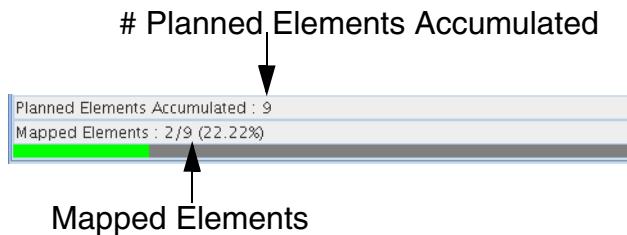
The *Tabs* pane has the following tabs:

- *Plan*—This tab has following tab pages:
  - *Attributes*—Lets you review and specify attributes for the element selected in the vPlan editor.
  - *Rich text editor*—Lets you integrate OpenOffice with vPlanner. For more details, see [Rich-text Editing](#) on page 111.
- *Specs*— This tab has a tab page named *Spec Manager* for importing specification PDF documents. It also shows a tab page for each open document where you can annotate elements in the vPlan with text from the specifications. For more details, see [Annotating and Updating Verification Plans](#) on page 81.
- *Implementation*—This tab has following tab pages:
  - *Import manager*—Lets you import and view the coverage model. For more details, see [Creating the Coverage Model](#) on page 143.
  - *Metrics*—Lets you view or map coverage metrics for a selected vPlan element. For more details, see [Creating the Coverage Model](#) on page 143.
  - *Logical instances*—Lets you view and remove logical instances defined in the verification plan. For more details, see [Using Logical Instances](#) on page 130.
  - *Coverage builder*—Lets you load a testbench, build new coverage item, and write e code and refresh coverage model. For more details, see [Using the Functional Coverage Builder](#) on page 180.
- *Messages*—Shows the summary of tasks performed, warnings, and errors in the vPlanner window.

### 1.4.5 Summary Pane

[Figure 1-11](#) on page 18 displays the *Summary* pane that appears on the vPlanner window.

**Figure 1-11 Summary Pane**



The *Summary* pane displays the value of *Planned Elements Accumulated* and *Mapped Elements* for the item selected in the vPlan editor.

- If a metrics port is selected in the vPlan editor:

- *Planned Elements Accumulated* shows any of the following values when a metrics port is selected:
    - If there are no mapping on the metric port and the *Planned Elements* attribute is not set, then the *Planned Elements Accumulated* value will be 1.
    - If there are no mapping on the metric port and the *Planned Elements* attribute is set, then the *Planned Elements Accumulated* value will be the value that is set for the *Planned Elements* attribute.
    - If mapping exists for the port, then the *Planned Elements Accumulated* value will be the maximum between the actual number of mapping and *Planned Elements* attribute value (if set).

**Note:** For details on *Planned Elements* attribute, see [Planned Elements](#) on page 38.

- *Mapped Elements* shows the following:

<Mapped elements>/<Planned elements> (Mapped elements grade)

where Mapped elements grade is the:

Percentage(%) of <Mapped elements>/<Planned elements accumulated>

**Note:** For more details on mapping elements, see [Creating the Coverage Model](#) on page 143.

- If a section or subsection is selected in the vPlan editor:

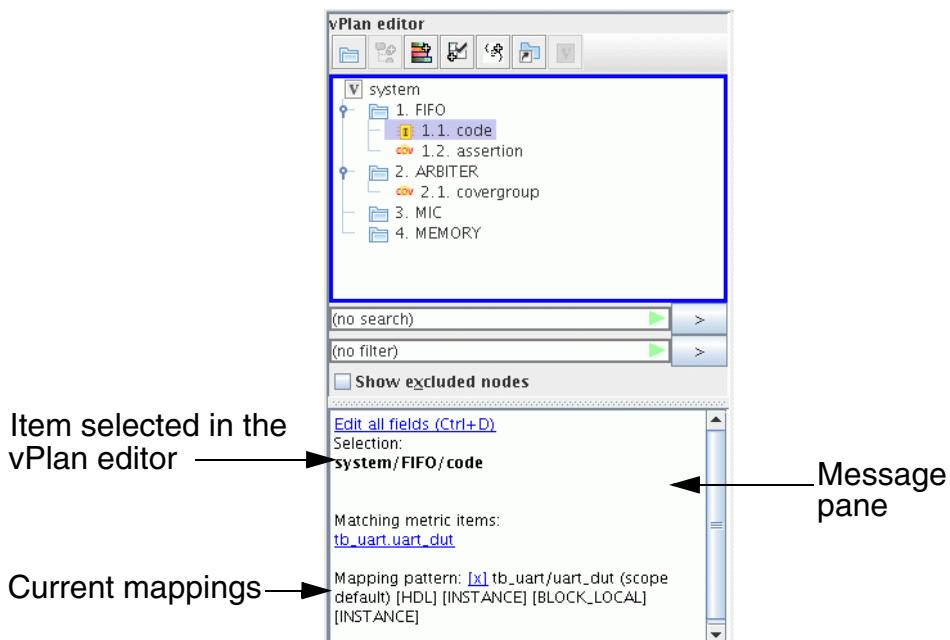
- *Planned Elements Accumulated* shows the rolled up result, which is the sum of planned elements of all the children within the section. Note that the value taken will be the maximum between the rolled up result and *Planned Elements* attribute value (if set).

- *Mapped Elements* shows the rolled up result, which is the sum the mapped elements of all the children within the section.

### 1.4.6 Message Pane

Figure 1-12 on page 19 displays the *Message* pane that appears on the vPlanner window.

**Figure 1-12 Message Pane**



The *Message* pane displays messages about the status of the item selected in the vPlan editor.

## 1.5 Opening a Verification Plan

If you have an existing plan in a .vplan, .xml, or .sxw format, it must be converted to a .vplanx format before it can be opened in vPlanner.

If you have an existing plan in a .vplanx format, you can open it at the time of launching vPlanner or after launching vPlanner.

This section covers the following topics:

- Converting a Legacy Plan to a .vPlanx Format

- Opening a .vplanx File At the Time of Launching vPlanner
- Opening a .vplanx File After Launching vPlanner

### 1.5.1 Converting a Legacy Plan to a .vPlanx Format

To convert a legacy plan that is in a .vplan, .xml, or .sxw format to a .vplanx format:

1. Invoke vManager in batch mode.

```
vmanager -batch
```

2. Use the following CLI command to convert the plan:

```
convert -vplan <file> [-overwrite]
```

where,

- ❑ <file> is the name of the file you plan to convert. 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.

- ❑ -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.

For example, 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.

After conversion, you can open the file in vPlanner.

### 1.5.2 Opening a .vplanx File At the Time of Launching vPlanner

To open an existing verification plan at the time of launching vPlanner, use:

```
% vmanager -planner <pathname_of_verification_plan>
```

or

```
% vplanner -load_vplan <pathname_of_verification_plan>
```

For example, to open `system.vplanx` saved under `myplans` directory at the time of launching vPlanner, use:

```
% vmanager -planner myplans/system.vplanx
```

or

```
% vplanner -load_vplan myplans/system.vplanx
```

The above command launches vPlanner with `system.vplanx` loaded.

### 1.5.3 Opening a .vplanx File After Launching vPlanner

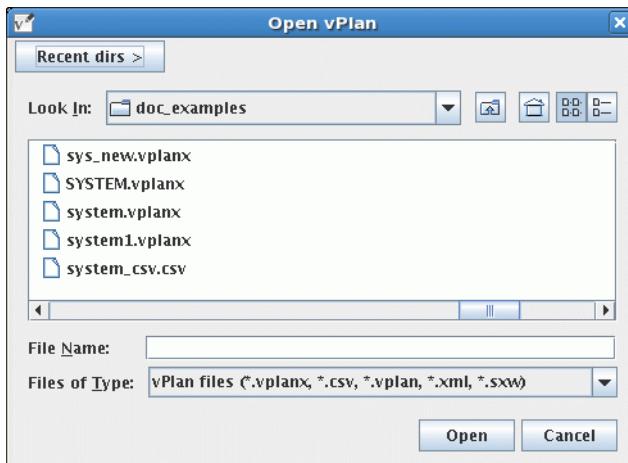
To open a verification plan after launching vPlanner:

1. Select *File* —> *Open*.

Alternatively, you can click the *Open* button on the toolbar.

The *Open vPlan* dialog box is displayed, as shown in [Figure 1-13](#) on page 21.

**Figure 1-13 Open Verification Plan**



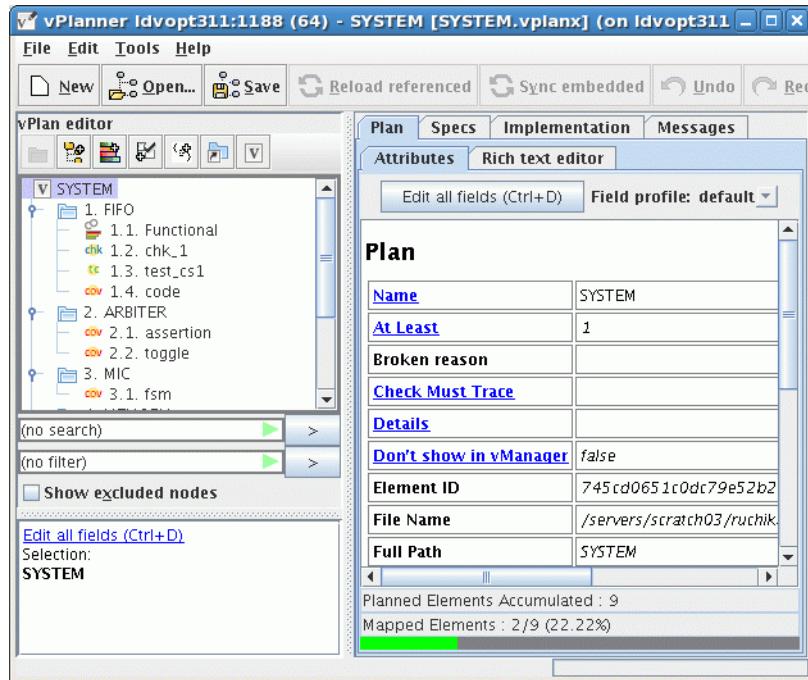
2. Navigate through the hierarchy and select the relevant verification plan. For example, to open the verification plan named `SYSTEM.vplanx` select `SYSTEM.vplanx` and click *Open*.

The selected verification plan is loaded, as shown in [Figure 1-14](#) on page 22.

## vPlanner User Guide

---

**Figure 1-14 vPlanner Window with Loaded Verification Plan**



After opening a verification plan, you can modify it, add mappings, and also export and publish the plan.

---

## Creating a Verification Plan

---

Using vPlanner, you can create a hierarchical plan that includes:

■ Sections and subsections

Each section describes a set of related features that need to be verified. Each section can include one or more subsections.

■ Metric Ports

Metric ports can be of type coverage, check, or testcase.

□ Coverage ports

It is recommended that you plan at least one coverage port for each of the main features in a section. Later, you can implement each port as a coverage metric in the verification environment. Possible coverage metrics include:

○ Functional coverage, including the implicit coverage of assertions

○ Code coverage

□ Check ports

If a main feature in a section describes a behavior to be checked, it is recommended that you plan a check port. Later you can implement each port as a checker in the verification environment. Possible check implementations include:

○ Assertion checks

○ Scoreboard checks

○ Reference model checks

□ Testcase (scenario) ports, directed tests

**Note:** Metric ports are leaf items in the verification plan hierarchy. It is not possible to create children of metric port items.

■ Perspectives

You can identify the features associated with a particular component, project team, or project milestone by selecting one or more sections and grouping them as a perspective.

This chapter covers the following topics:

- [Creating a New Verification Plan](#)
- [Creating and Editing Sections](#)
- [Creating Coverage Ports](#)
- [Creating Check Ports](#)
- [Creating Testcase Ports](#)
- [Deleting, Copying, Moving, and Renaming vPlan Elements](#)
- [Attributes Associated with a vPlan Element](#)
- [Specifying a User-Defined Proof Bound](#)
- [Searching and Filtering Items in the vPlan Hierarchy](#)
- [Creating Perspectives](#)
- [CSV-Based Verification Plans](#)
- [Creating and Using vPlan User-defined Attributes](#)

## 2.1 Creating a New Verification Plan

To create and save a new verification plan from scratch:

1. Select *File* —> *New*.

Alternatively, you can click the *New* button on the toolbar.

2. Add sections, metric ports (coverage, check, testcase), as desired.

For details on adding sections, coverage ports, check ports, and testcase ports, see subsequent topics in this chapter.

3. Select *File* —> *Save as*.

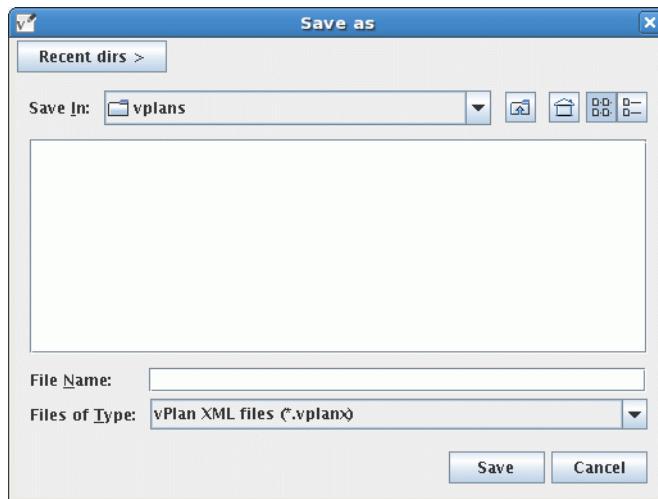
Alternatively, you can click the *Save* button on the toolbar.

**Note:** You can also save the verification plan as a CSV file. For that, select *File* —> *Save as (csv)*. You can also save the verification plan as a CSV file. For that, select *File* —> *Save as (csv)*. If you save the file as a CSV file, then the fields NAME, DEPTH,

NODE\_KIND, MAPPING\_PATTERN, element\_id, metrics\_port\_kind are added to the CSV file. For more details, on CSV based vPlans, see [CSV-Based Verification Plans](#) on page 61.

The Save as dialog box is displayed, as shown in [Figure 2-1](#) on page 25.

**Figure 2-1 Save Verification Plan**



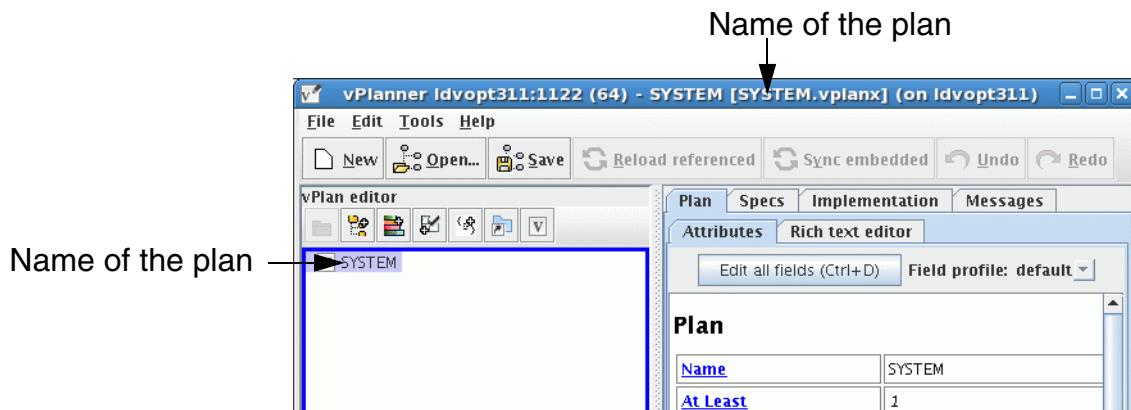
4. Browse to the directory where you want to save the plan.
5. Enter a name for the plan in the *File Name* field and click *Save*.

For example, to save the plan as SYSTEM, enter *SYSTEM* in the *File Name* field and click *Save*.

**Note:** When you save a plan, by default, the plan is stored with a .vplanx extension.

The name of the plan appears as the title of the vPlanner window, as shown in [Figure 2-2](#) on page 26.

**Figure 2-2 vPlanner Window with the Saved Verification Plan**



All verification plans, by default, are stored with a \*.vplanx extension.

## 2.2 Creating and Editing Sections

The next step in creating a verification plan is to create the section hierarchy.

### 2.2.1 Creating Child Sections

To create a new child section:

1. Select the root node of the plan or any other section.

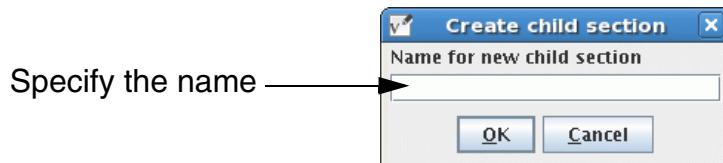
For example, to create a child section within root node SYSTEM, select SYSTEM.

2. Do one of the following:

- Click the *Child Section* button .
- Press the Insert key on the keyboard.
- Right-click on the vPlan element and select *New section as child*.

The *Create child section* dialog box appears, as shown in [Figure 2-3](#) on page 27.

**Figure 2-3 Create Child Section**



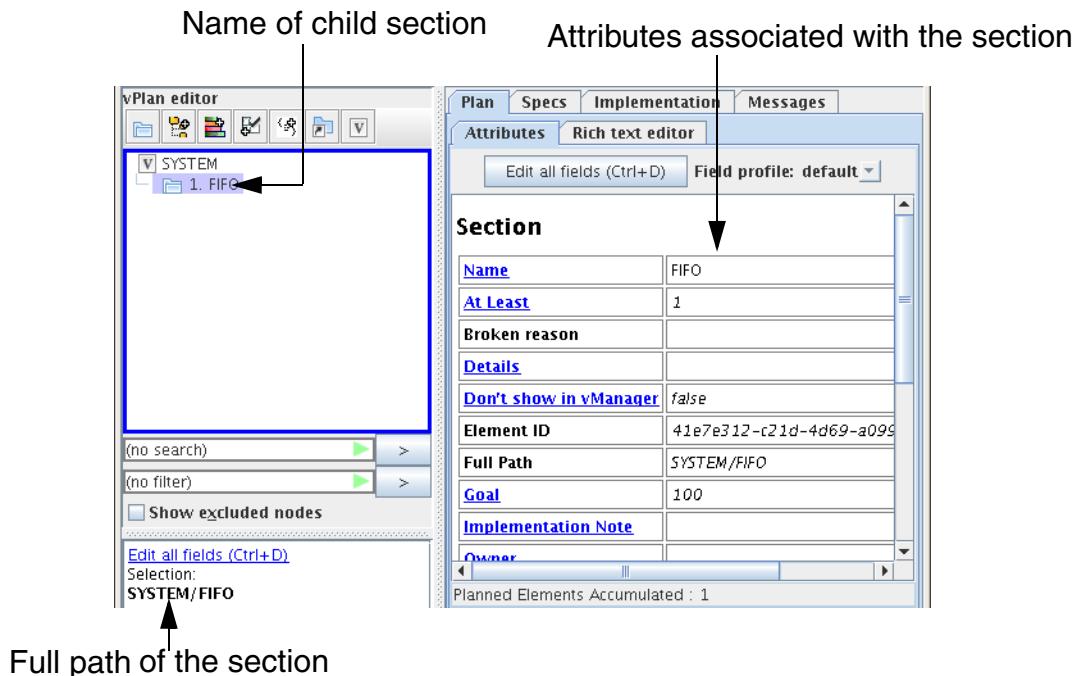
3. Type the name of the section in the field that appears and click *OK*.

For example, to create a child named *FIFO*, type *FIFO* and click *OK*.

**Note:** You can also press the *Enter* key instead of clicking on *OK*.

The new section is highlighted in the *vPlan* editor, as shown in [Figure 2-4](#) on page 27.

**Figure 2-4 vPlanner Window with the New Child Section**



The new section is highlighted in the *vPlan* editor. The attributes associated with the section are shown in the *Attributes* pane, with their default values. You can define or modify the values of these attributes. For more details on attributes associated with the section, see [Attributes Associated with a vPlan Element](#) on page 34.

Similarly, you can add more sections.

### 2.2.2 Creating Sibling Sections

To create a sibling section:

1. Select the section to which you want to add a sibling.
2. Do one of the following:

- Click the *Sibling Section* button , or press **Ctrl+B** keys.
- Right-click on the vPlan element and select *New section as sibling*.

The *Create sibling section* dialog box appears, as shown in [Figure 2-5](#) on page 28.

**Figure 2-5 Create Sibling Section**



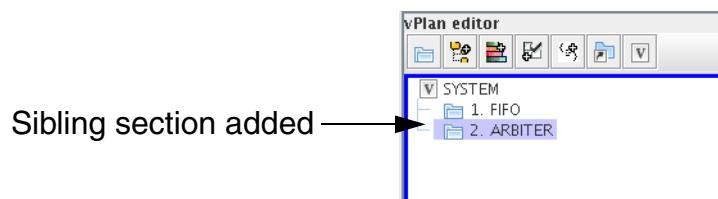
3. Type the name of the section in the field that appears and click *OK*.

For example, to create a sibling named ARBITER, type *ARBITER* and click *OK*.

**Note:** You can also press the **Enter** key instead of clicking on *OK*.

The new sibling section is added to the vPlan editor, as shown in [Figure 2-6](#) on page 28.

**Figure 2-6 vPlan Editor with the New Sibling Section**



The new section is highlighted in the vPlan editor. The attributes associated with the section are shown in the *Attributes* pane, with their default values. You can define or modify the values of these attributes. For more details on attributes associated with the section, see [Attributes Associated with a vPlan Element](#) on page 34.

Similarly, you can create more sibling sections.

## 2.3 Creating Coverage Ports

When planning the coverage model, you need to identify the features that you want to monitor. You can create a separate coverage port for each attribute of the feature you want to monitor.

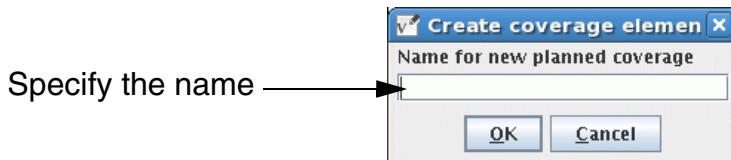
To create a new coverage port:

1. Do any of the following:

- Click the *New Coverage* button , or press **Ctrl+G** keys.
- Right-click on a section and select *New planned coverage*.

The *Create coverage element* dialog box appears, as shown in [Figure 2-7](#) on page 29.

**Figure 2-7 Create Coverage Element**

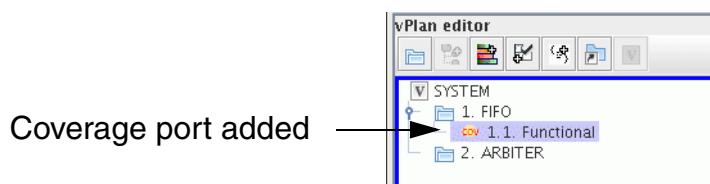


2. Type the name of the coverage port in the field that appears and click *OK*.

For example, to create a coverage port named *Functional* within the section *FIFO*, select *FIFO* and click the *New Coverage* button. Then, type *Functional* and click *OK*.

The new coverage port is created within *FIFO*, as shown in [Figure 2-8](#) on page 29.

**Figure 2-8 vPlan Editor with the New Coverage Port**



The new coverage port is highlighted in the vPlan editor. The attributes associated with the coverage port are shown in the *Attributes* pane, with their default values. You can define or modify the values of these attributes. For more details on attributes associated with coverage port, see [Attributes Associated with a vPlan Element](#) on page 34.

## 2.4 Creating Check Ports

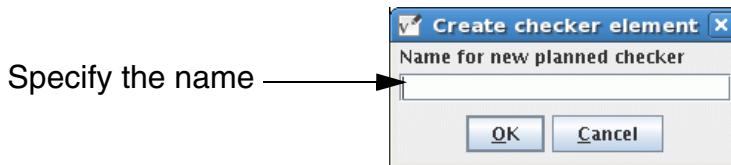
When planning the checker requirements, you need to identify the behaviors whose correctness you want to verify. You can create a separate check port for each behavior you want to check.

To create a new check port:

1. Select a section within which you want to create a check port.
2. Do any of the following:
  - Click the *New Check* button , or press **Ctrl+K** keys.
  - Right-click on a section and select *New planned checker*.

The *Create checker element* dialog appears, as shown in [Figure 2-9](#) on page 30.

**Figure 2-9 Create Checker Element**

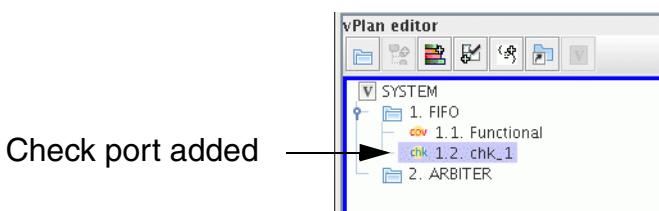


3. Type the name of the checker port in the field that appears and click *OK*.

For example, to create a checker port named `chk_1` within the section FIFO, select *FIFO* and click the *New Check* button. Then, type `chk_1` and click *OK*.

The new checker port is created within FIFO, as shown in [Figure 2-10](#) on page 30.

**Figure 2-10 vPlan Editor with the New Checker Port**



The new check port is highlighted in the vPlan editor. The attributes associated with the check port are shown in the *Attributes* pane, with their default values. You can define or modify the values of these attributes. For more details on attributes associated with check port, see [Attributes Associated with a vPlan Element](#) on page 34.

## 2.5 Creating Testcase Ports

You can create testcase (scenario) ports that are used to exercise a specific feature.

To create a new testcase port:

1. Select a section in which you want to create a testcase port.
2. Do any of the following:

- Click the *New Testcase* button , or press **Ctrl+T** keys.
- Right-click on a section and select *New planned testcase*.

The *Create testcase element* dialog box appears, as shown in [Figure 2-11](#) on page 31.

**Figure 2-11 Create Testcase Element**

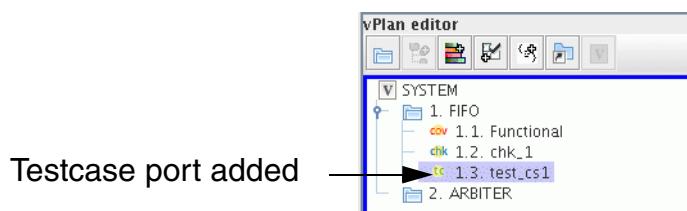


3. Type the name of the testcase port in the field that appears and click *OK*.

For example, to create a testcase port named `test_cs1` within the section FIFO, select *FIFO* and click the *New Testcase* button. Then, type `test_cs1` and click *OK*.

The new testcase port is created within FIFO, as shown in [Figure 2-12](#) on page 31.

**Figure 2-12 vPlan Editor with the New Testcase Port**



The new testcase port is highlighted in the vPlan editor. The attributes associated with the testcase port are shown in the *Attributes* pane, with their default values. You can define or modify the values of these attributes. For more details on attributes associated with testcase port, see [Attributes Associated with a vPlan Element](#) on page 34.

## 2.6 Deleting, Copying, Moving, and Renaming vPlan Elements

Using vPlanner, you can delete, move, and rename vPlan elements (sections, and metric ports).

### 2.6.1 Deleting vPlan Elements

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

- Select the vPlan element that you want to delete and press the `Del` key.
- Right-click the vPlan element that you want to delete and select *Delete*.

A confirmation dialog appears and once you confirm the deletion, it is deleted from the vPlan hierarchy.

**Note:** You can use *Undo* to restore a deleted element. You can select *Undo* from the *Edit* menu or click *Undo* in the toolbar.

**Note:** You can delete multiple vPlan elements at a time by selecting them (to select multiple vPlan elements, use `Shift+Select` or `Ctrl+Select`) in the vPlan hierarchy.

### 2.6.2 Copying and Pasting vPlan Elements

To copy and paste a vPlan element:

- Select the element that you want to copy and click `Ctrl+C` or right-click and select *Copy* (`Ctrl+C`). In case you want to do a cut and paste, click `Ctrl+X` to cut the element.
- Select the element where you want to paste it and click `Ctrl+V` or right-click and select *Paste* (`Ctrl+V`).

### 2.6.3 Moving vPlan Elements

To move a vPlan element(s):

1. Select the element(s) you want to move (to select multiple elements, use `Shift+Select` or `Ctrl+Select`).
2. Drag and drop the element(s) to the new location.

A green marker stretches to show the new location.

**Note:** You can also use the Tab key to indent the selected element, and Shift+Tab keys to unindent the selected element.

### 2.6.4 Renaming vPlan Elements

To rename a vPlan element:

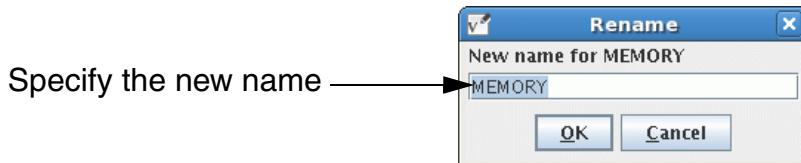
1. Do one of the following:

- Select the vPlan element you want to rename and press the F2 key on the keyboard.
- Right-click the element you want to rename and select *Rename*.

For example, to rename the section MEMORY as MIC, select *MEMORY* in the vPlan hierarchy and press the F2 key.

The *Rename* dialog appears, as shown in [Figure 2-13](#) on page 33.

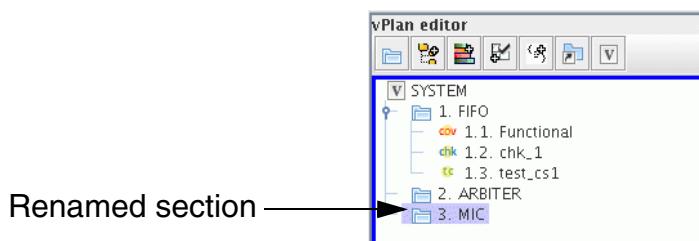
**Figure 2-13 Rename a Section**



2. Specify the new name and click *OK*. For example, type *MIC* and click *OK*.

The new name shows in the vPlan hierarchy, as shown in [Figure 2-14](#) on page 33.

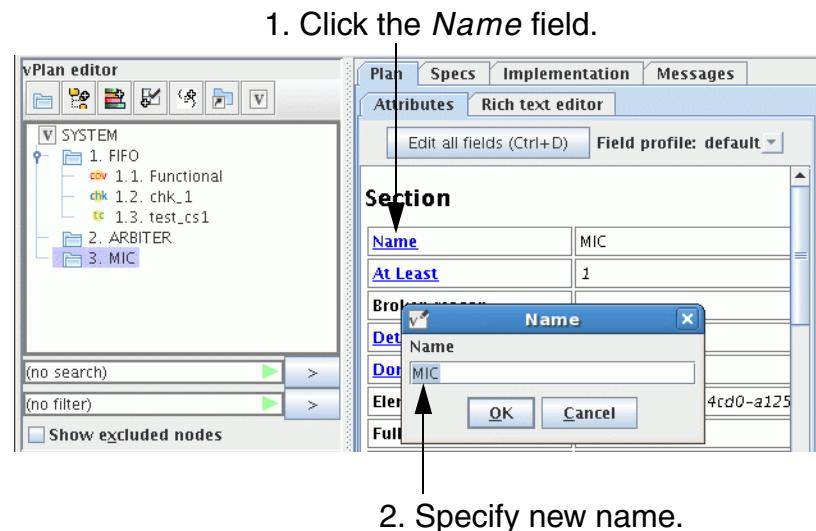
**Figure 2-14 vPlan Editor with Renamed Section**



Notice that *MEMORY* is renamed as *MIC*.

**Note:** You can also change the name of the vPlan element by clicking the *Name* field in the *Attributes* tab. You can then specify the new name in the dialog box that appears and click *OK*, as shown in [Figure 2-15](#) on page 34.

**Figure 2-15 Renaming Section**



## 2.7 Attributes Associated with a vPlan Element

Table 2-1 describes the attributes associated with a vPlan element (section, or a metric port).

**Table 2-1**

Attribute Name	Description
Name	The name of the selected item.
At least	<p>Specifies the number of hits for the selected vPlan item to be considered covered.</p> <p>Default value for this attribute is 1.</p> <p>The value specified in this field propagates to the children.</p>
Broken reason	Shows the reason for a broken item. If the selected item shows a red ! in the vPlan tree, it indicates that there is a problem with the selected item. The problem can be an error in locating referenced element, some unresolved logical instance, or matching metric items not found for a metric port. In such cases, this attribute shows the reason for a broken item.

## vPlanner User Guide

---

Attribute Name	Description
Check Must Trace	<p>Lets you specify if for the selected vPlan element, the assertion properties must have a trace in order to be marked as <i>Proved</i>.</p> <ul style="list-style-type: none"> <li>■ If enabled (<code>true</code>), the assertions must have a trace for the status to be marked as <i>Proved</i>.</li> <li>■ If disabled (<code>false</code>), the assertion status can be marked as <i>Proved</i> without a trace.</li> </ul> <p><b>Note:</b> This option impacts the <i>Formal Status Grade</i> calculation.</p>
Bin Filter	<p>Lets you specify a bin filter to exclude the specified bins from grade calculations. This attribute does not affect the way data is displayed in vPlanner. This attribute is used by vManager during vPlan grading.</p> <p>For details on specifying a bin filter, see <a href="#">Creating Bin Filters</a> on page 173.</p>
Details	<p>Lets you specify a description of the vPlan element. You can add a detailed description of the vPlan element in this field.</p> <p>By default, no value is specified for this attribute.</p>
Don't show in vManager	<p>Lets you enable or disable showing the selected item in vManager. By default, the value is set as <code>false</code>, which indicates that the selected item will be shown in vManager. To not show the selected item in vManager, set the value as <code>true</code>.</p>
Element ID	<p>Is the ID that is automatically generated by Planner (if not present). This field is automatically set by vPlanner and you cannot modify it.</p>
File name	<p>Shows the name of vPlan file along with the complete hierarchical path.</p>
Full path	<p>Shows the full path of the vPlan item selected in the vPlan hierarchy.</p> <p>You cannot edit this attribute.</p>

## vPlanner User Guide

---

Attribute Name	Description
Goal	Specifies the target goal percentage for the vPlan item selected in the vPlan hierarchy.  Target goal, by default, is set as 100.
Implementation Note	Lets you specify any notes regarding the implementation requirement for review.  By default, no value is specified for this attribute.
Metrics Port Kind	Lets you change the type of port. By default, it shows the type of port of the selected vPlan element. The valid types are COVERAGE, TESTCASE, and CHECK.
Owner	Lets you specify the name of the engineer responsible for the selected item. The value of this attribute propagates to the children.  By default, no value is specified for this attribute.
Plan Source Tool	Shows the name of the tool in which the file was created.
Priority	Specifies the priority of the section relative to other sections. For example, you can assign priority as Critical, P0, or Must Have.  The value specified in this field propagates to the children.  By default, no value is specified for this attribute, and there are no predefined values that can be used to set this attribute.
Spec Text	Shows the annotated text after you add an annotation. For more details, see <a href="#">Annotating and Updating Verification Plans</a> on page 81.
Updates	Shows the update made to the selected item after it was created. For example, you must check this field after you synchronize an embedded plan, or update an specification document to review the changes, if any.

Attribute Name	Description
Verification Aspect	<p>Lets you specify the grade (formal, simulation, or both) to be shown in vManager for the selected vPlan element. Valid values for this attribute are:</p> <ul style="list-style-type: none"> <li>■ <b>SIMULATION</b> — If the value of this attribute is <b>SIMULATION</b>, then the formal grade and status is not calculated and NA is shown in vManager during analysis.</li> <li>■ <b>FORMAL</b> — If the value of this attribute is <b>FORMAL</b>, then the simulation grade and status is not calculated and NA is shown in vManager during analysis.</li> <li>■ <b>BOTH</b> — If the value of this attribute is <b>BOTH</b>, then both simulation and formal metrics is calculated and shown in vManager during analysis.</li> </ul> <p><b>Note:</b> Default value of this attribute is <b>BOTH</b>.</p>
Verification Scope	<p>Lets you specify the verification scope to which the selected vPlan element must belong.</p> <p>By default, all vPlan elements belong to <b>default</b> scope.</p> <p>Setting verification scope is useful in some complex environments where two or more testbenches are required to completely verify the behavior of a component. For example, you might have both a module-level testbench and a system-level testbench that test different capabilities of the same component. Or, you might have different testbenches for a component, depending on whether it is functioning as a master agent or as a slave. There can also be a common case where you want to keep formal and simulation results separate.</p> <p>In these cases, you might want to track verification metrics separately for each testbench. To do this, you can define a verification scope for tests associated with a particular testbench and define the same scope for the sections of the plan that describe features that are verified by those tests.</p>

## vPlanner User Guide

---

Attribute Name	Description
Weight	<p>Specifies the weight of the item selected in the vPlan editor.</p> <p>Default value of this attribute is 1.</p> <p>If a particular item is more important than others, it should be assigned a higher weight than others so that its grade affects the overall grade more than the grades of other sections.</p>
Flag	<p>Shows the flag that is automatically set by vPlanner to reflect any updates to specs or any changes to foreign plans. The values shown in this field might be like user, pdf_delete, pdf_modified, foreign_add, foreign_move, or foreign_delete. You can query for Flag values with predefined or custom filters.</p> <p><b>Note:</b> This field is automatically set by vPlanner and you cannot modify it.</p>
Parameter condition	<p>Allows you to associate a parameter condition with the selected vPlan item. This attribute is useful if you are creating a reusable plan.</p> <p><b>Note:</b> Parameters are used to selectively exclude parts of the plan that are not applicable in a given perspective. A parameter must be defined before specifying a parameter condition. For more details, see <a href="#">Reusing Verification Plans</a> on page 117.</p>
Planned Elements	Allows you to specify the number of elements that you intend to add without actually creating the metrics port.
Mapping Pattern	<p>When you map the vPlan element with a relevant metrics, it shows the associated mapping pattern. It also allows you to add a mapping pattern for the selected vPlan element. By default, it does not show any value. This attribute applies only to coverage, testcase, and check ports.</p> <p>For more details, see <a href="#">Mapping Existing Coverage Metrics to the vPlan</a> on page 150.</p>

Attribute Name	Description
Proof Bound	<p>The <i>Proof Bound</i> attribute is specified by the user in the vPlan to qualify the bounded proof status of a property in formal environment. If the Actual Proof Bound is <math>\geq</math> Proof Bound, then the <i>Proved count</i> and <i>Passed count</i> of the formal property is shown 1 at the time of analysis.</p> <p>It is set at the port level or section level. The value specified in this field propagates to the children and affects all the assertions below it.</p> <p>Default value for this attribute is 2147483647, which is MAX_INT.</p> <p>For more details, see <a href="#">Specifying a User-Defined Proof Bound</a> on page 41.</p>

You can define or modify the values of these attributes. You can also add and use user-defined attributes. For more details, see [Creating and Using vPlan User-defined Attributes](#) on page 74.

### 2.7.1 Editing Attributes

You can define or modify attributes in any of the following ways:

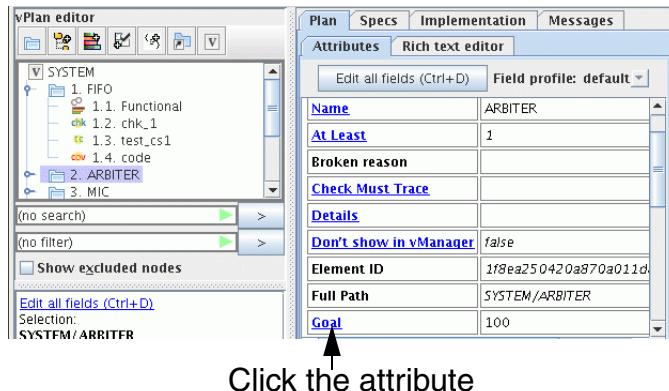
- [Modifying or Defining Individual Attributes](#)
- [Modifying or Defining All Attributes Together](#)

#### 2.7.1.1 Modifying or Defining Individual Attributes

To modify the value of an attribute:

1. In the *Attributes* tab, click on the attribute that you want to modify or define. For example, click the *Goal* attribute, as shown in [Figure 2-16](#) on page 40.

**Figure 2-16 Edit Attributes**



Click the attribute

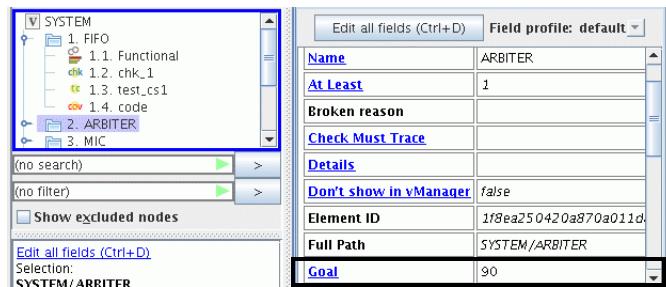
- Type the value of the attribute in the dialog box that appears and click *OK*. For example, specify 90 in the *Goal* dialog box and click *OK*, as shown in [Figure 2-17](#) on page 40.

**Figure 2-17 Edit Attributes**



The value you specify for the attribute is set, as shown in [Figure 2-18](#) on page 40.

**Figure 2-18 Value Set for Attribute Goal**



The value of the *Goal* attribute now shows as 90.

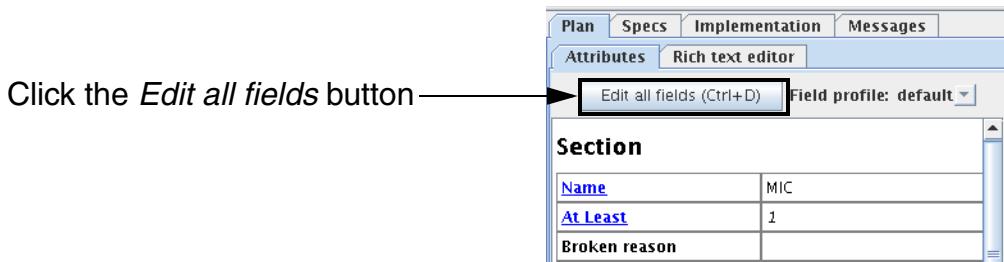
## 2.7.1.2 Modifying or Defining All Attributes Together

To modify or define all of the attributes together:

1. Do any of the following:

- ❑ In the *Attributes* tab, click the *Edit all fields* button, as shown in [Figure 2-18](#) on page 41.

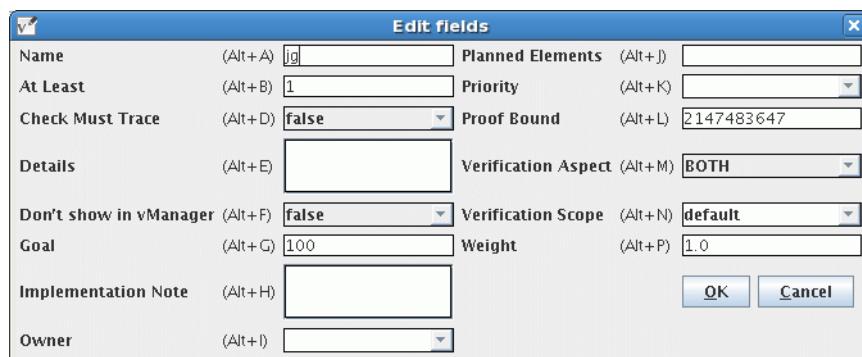
**Figure 2-19 Edit all fields**



- ❑ From the *Edit* menu, select *Edit all fields*.
- ❑ Press **Ctrl+D** keys.

This opens the *Edit fields* dialog box, as shown in [Figure 2-20](#) on page 41.

**Figure 2-20 Edit fields**



2. Modify or define the values of the required attributes and click *OK*.

**Note:** You can navigate between fields with Tab and Shift+Tab keys. You can also use the shortcut keys displayed in the dialog box to quickly navigate to that field.

The values you specify in this dialog box are set.

## 2.8 Specifying a User-Defined Proof Bound

Using the *Proof Bound* attribute of vPlan, you can define a user-defined bound which will

be used to qualify the bounded proof status of a property in formal environment.

The value of this attribute is set at the port level or section level.

The default value for this attribute is 2147483647, which is MAX\_INT.

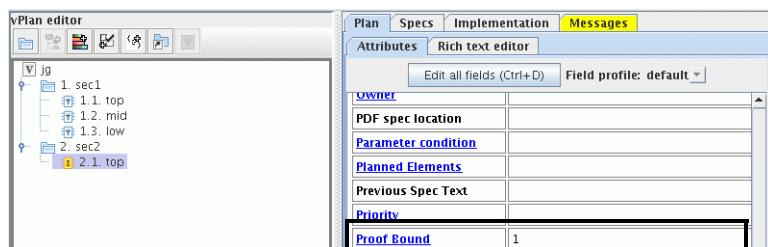
You can modify the value of the attribute, as required. The value specified in this field propagates to the children and affects all the assertions below it.

To specify a proof bound:

1. Select the port or section for which you want to define a proof bound.
2. In the *Attributes* tab, click on the attribute *Proof Bound*.
3. Type the value of the attribute in the dialog box that appears and click *OK*. For example, specify 1 and click *OK*.

The value you specify for the attribute is set, as shown in [Figure 2-21](#) on page 42.

**Figure 2-21 Value Set for Attribute Goal**



The value of the *Proof Bound* attribute now shows as 1. This value is available at the time of analysis in vManager, as shown in [Figure 2-22](#) on page 42.

**Figure 2-22 Value Set for Attribute Goal**

Name	Proof Bound	Actual	Formal	Formal	Formal	Formal
		Proof Bound	Status	Passed	Count	Proved
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
jg	2147483647	n/a	■■■■■ 0%	n/a	n/a	n/a
1. sec1	2147483647	n/a	■■■■■ 0%	n/a	n/a	n/a
2. sec2	2147483647	n/a	■■■■■ 0%	n/a	n/a	n/a
2.1. top	1	n/a	■■■■■ 0%	n/a	n/a	n/a
top	1	n/a	■■■■■ 0%	n/a	n/a	n/a
M	1	n/a	■■■■■ 0%	n/a	n/a	n/a
L	1	n/a	■■■■■ 0%	n/a	n/a	n/a
ASSERT_FAIL_TRA	1	2	■■■■■ 0%	1	0	0
ASSERT_FAIL_TRA	1	0	■■■■■ 0%	0	0	1
ASSERT_PASS_TRA	1	2	■■■■■ 0%	1	0	0
ASSERT_PASS_TRA	1	2	■■■■■ 0%	1	0	0
COVER_FAIL	1	0	n/a	0	0	0
COVER_PASS	1	0	n/a	0	0	0

Notice that the value specified for the proof bound for `top` has propagated to the children

and all the assertions below it.

During analysis in vManager, if required, you can change the value of the *Proof Bound* attribute in the *Refine each* dialog box. For more details, see the *vManager User Guide*.

A user defined proof-bound affects only the formal attributes of an assertion, as following:

*Formal Check proved Count* =

```
(assertion bound >= user specified bound && Formal Cover Finished Count >0 )?  
    (max(value retrieved directly from unicov,1)) : value retrieved directly  
    from unicov
```

*Formal Check Passed Count* =

```
(assertion bound >= user specified bound)?  
    (max(value retrieved directly from unicov,1)) : value retrieved directly  
    from unicov
```

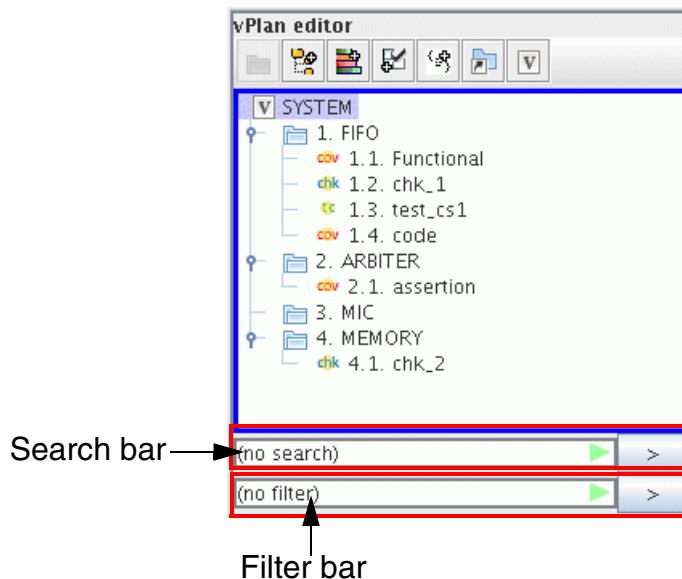
**Note:** Calculation of Formal Status, Formal Check Failed Count and Formal cover \* attributes are not impacted by user-defined bound.

If the actual proof bound of a property is  $\geq$  to the defined Proof Bound, then the *Proved count* and *Passed count* of the formal property is shown 1 at the time of analysis.

## 2.9 Searching and Filtering Items in the vPlan Hierarchy

Figure 2-23 on page 44 displays the search bar and the filter bar available at the bottom of the vPlan hierarchy.

**Figure 2-23 Search and Filter Data**



The search bar allows you specify the search criteria and once you click the green arrow button next to the text box, it highlights the items that meet the search criteria in the vPlan editor.

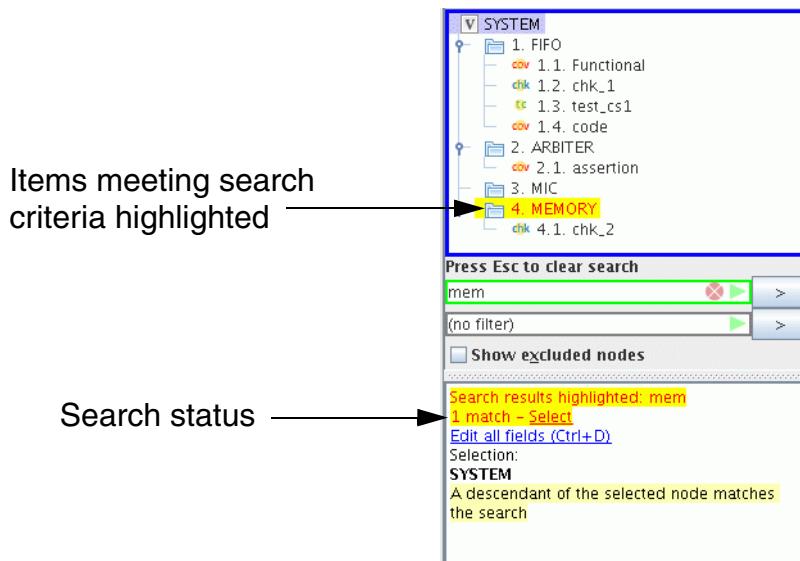
The filter bar allows you specify the filter criteria and once you click the green arrow button next to the text box, it filters the vPlan hierarchy to show only the items that meet the filter criteria. The nodes that match the filter criteria are highlighted, and the nodes that are included only because their descendants match are not highlighted.

### ***Example 1: Search the vPlan Hierarchy***

To search the vPlan hierarchy for elements that include `mem` in the name, specify `mem` in the search text box and click the green arrow button.

[Figure 2-24](#) on page 45 shows the vPlan hierarchy with items that meet search criteria highlighted.

**Figure 2-24 Search for Data**



In the vPlan hierarchy, elements that meet the search criteria are highlighted.

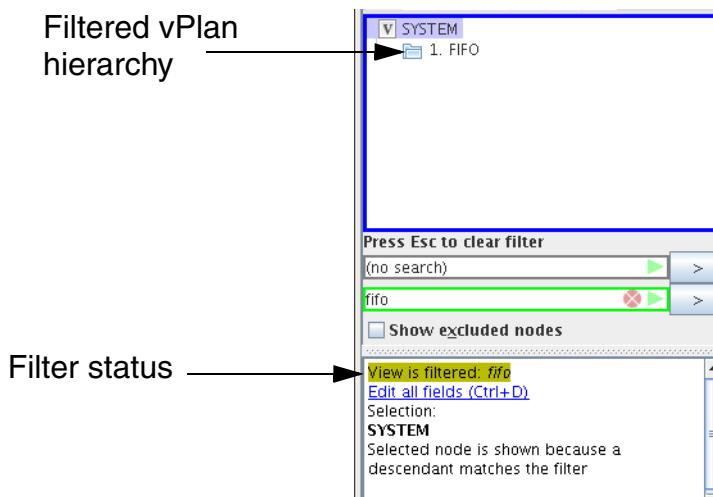
To clear the search criteria, click the cross marked red in the search text box. You can also press the `Esc` key to clear the search criteria.

### ***Example 2: Filter the vPlan Hierarchy***

To filter the vPlan hierarchy to display only the items that include `fifo`, specify `fifo` in the filter text box and click the green arrow button.

[Figure 2-25 on page 46](#) shows the filtered vPlan hierarchy.

**Figure 2-25 Filtered Data**



The vPlan hierarchy is now filtered to show only the items that meet the filter criteria.

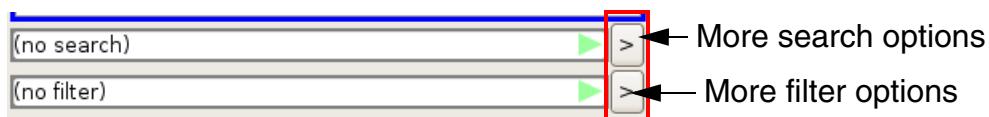
**Note:** If you are not sure why a node is highlighted or shown, check the reason in the area below the vPlan.

To clear the filter criteria, click the cross marked red in the filter text box. You can also press the `Esc` key to clear the filter criteria.

### 2.9.0.3 Advanced Search and Filtering

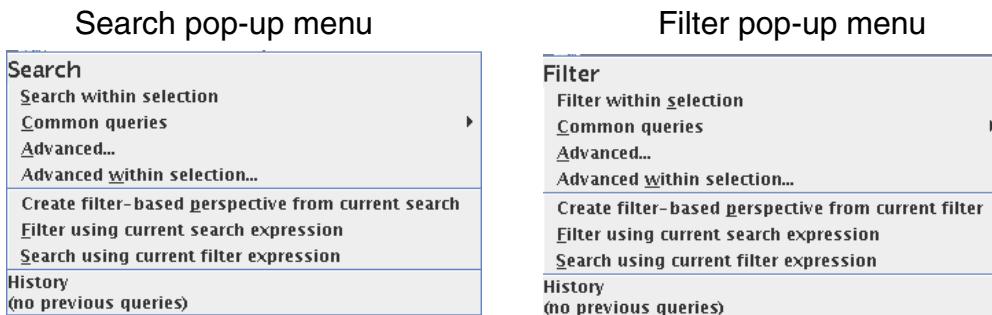
Figure 2-26 on page 46 shows the buttons that allow you to invoke more search and filtering options.

**Figure 2-26 Advanced Search and Filter Buttons**



When you click the more search options or more filter options button, a pop-up menu is displayed, as shown in Figure 2-27 on page 47.

**Figure 2-27 Advanced Search and Filtering**



You can select any of the following:

- Search within selection—To apply search only within the elements selected in the vPlan hierarchy.
- Filter within selection—To apply filter only within the elements selected in the vPlan hierarchy.
- Common queries—To quickly search or filter based on the list of pre-defined queries. See [Table 2-2](#) on page 48 for list of pre-defined queries.
- Advanced—To create your own search or filter query.
- Advanced within selection—To create your own search or filter query and apply that query only to the nodes selected in the vPlan hierarchy.
- Create filter-based perspective from the current search— To create a perspective based on the expression specified in the search text box. For more details on perspective, see [Creating Perspectives](#) on page 55.
- Create filter-based perspective from the current filter— To create a perspective based on the expression specified in the filter text box. For more details on perspective, see [Creating Perspectives](#) on page 55.
- Filter using current search expression—To apply filtering based on the expression specified in the search text box.
- Search using current filter expression—To apply search based on the expression specified in the filter text box.
- The History section maintains a history of search or filter queries ran within the current session of Planner. You can quickly run the query by selecting it. It saves time as you do not need to type the query again.

### Common Queries

This option provides a list of pre-defined queries that you can use to quickly search or filter the vPlan hierarchy tree. When you click on *Advanced* —> *Common queries*, a submenu is displayed to show the list of available queries that you can select.

Table 2-2 on page 48 lists the options within *Common queries* and their description.

**Table 2-2**

Common Queries	Description
Annotated	To search or filter the vPlan hierarchy to highlight or show annotated items.
Broken	To search or filter the vPlan hierarchy to highlight or show broken items.
User flag set	To search or filter the vPlan hierarchy to highlight or show items flagged by the user.
Spec changed	To search or filter the vPlan hierarchy to highlight or show the items where the underlying spec has changed.
Embedded plan updated	To search or filter the vPlan hierarchy to highlight or show items from embedded plans that were updated in the last synchronization.
TODO	To search or filter the vPlan hierarchy to highlight or show only the items where text TODO is found in the <i>Details</i> field.
Sections	To search or filter the vPlan hierarchy to highlight or show only the sections.
Metric ports	To search or filter the vPlan hierarchy to highlight or show only metric port items.
References	To search or filter the vPlan hierarchy to highlight or show only referenced plans.
Perspectives	To search or filter the vPlan hierarchy to highlight or show only perspectives.
Referenced elements	To search or filter the vPlan hierarchy to highlight or show only referenced items. Items within referenced plans and perspectives are considered as referenced elements.

Common Queries	Description
Mapped metric ports	To search or filter the vPlan hierarchy to highlight or show only the items that are mapped.
Unmapped metric ports	To search or filter the vPlan hierarchy to highlight or show only the items that are not mapped.
Coverage metric ports	To search or filter the vPlan hierarchy to highlight or show only the coverage metric ports.
Check metric ports	To search or filter the vPlan hierarchy to highlight or show only the check metric ports.
Testcase metric ports	To search or filter the vPlan hierarchy to highlight or show only the testcase metric ports.
No mapped inside	To search or filter the vPlan hierarchy to highlight or show sections that contain no mapped elements recursively.
No planned inside	To search or filter the vPlan hierarchy to highlight or show sections that contain no planned elements recursively.
Priority <= 1	To search or filter the vPlan hierarchy to highlight or show items where priority is less than 1.
No rich text	To show or filter the vPlan hierarchy to highlight or show items in which no rich text is defined.
Has rich text	To show or filter the vPlan hierarchy to highlight or show items in which rich text is defined.

### Advanced

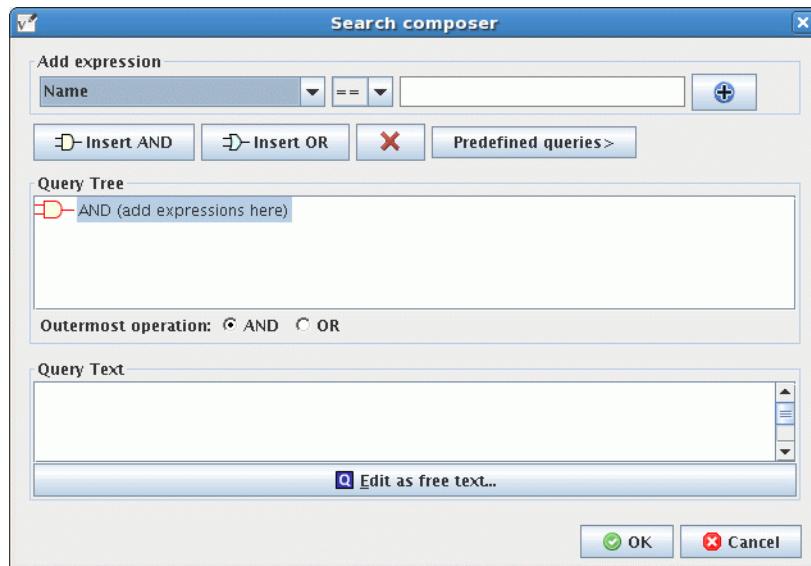
For example, you want to search the vPlan elements where owner is Bob or the priority is <= 1. For this, you have to create a custom query.

To create a custom query:

1. Click the *More search options* button and select *Advanced* from the pop-up menu.

A *Search composer* dialog box appears, as shown in [Figure 2-28](#) on page 50.

**Figure 2-28 Search composer**

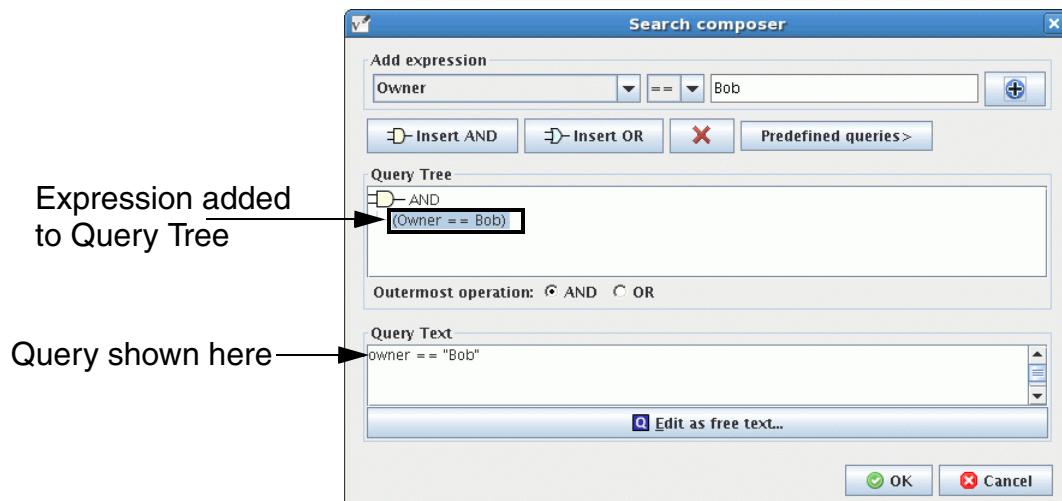


**2. In the *Add expression* section:**

- a.** Select the attribute on which the search must apply. For example, select *Owner* from the drop-down list.
- b.** Select the operator for creating the expression. For example, select *==* from the operator drop-down list.
- c.** Specify the value of the attribute for creating the expression. For example, specify *Bob* in the text box.
- d.** Click the *+* button to add the expression to the *Query Tree*.

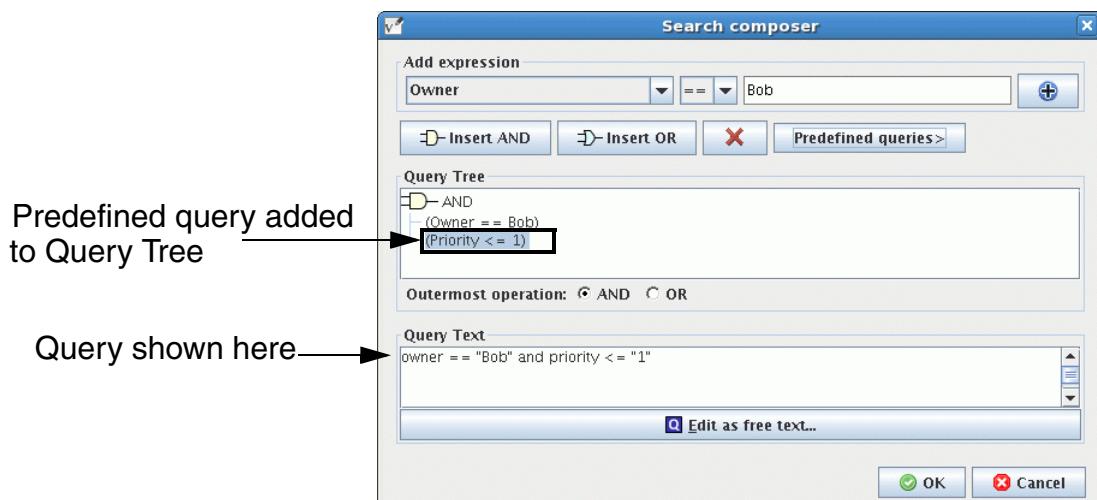
When you click the *+* button, the expression is added to the *Query Tree* and is also shown in the *Query Text* text box, as shown in [Figure 2-29](#) on page 51.

**Figure 2-29 Search Composer**



**Note:** You can also add expressions to the *Query Tree* using a list of predefined queries. The list of predefined queries is visible when you click the *Predefined queries* button. When you select a query from the list of predefined queries, it gets added to the *Query Tree*. For example, to further build the query where *Owner* is *Bob* and *priority* is  $\leq 1$ , click the *Predefined queries* button and select *Priority  $\leq 1$*  from the list. The query is added to the *Query Tree*, as shown in [Figure 2-39](#) on page 59.

**Figure 2-30 Filter-based Perspective**



While building the query, you can:

- Use the *Insert AND* button to add the AND operation in the query.

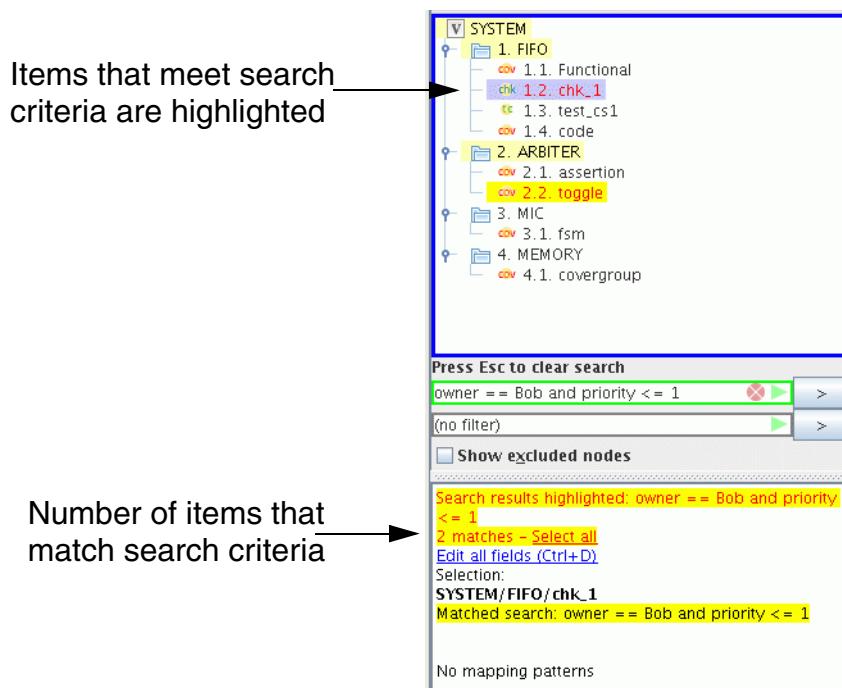
- ❑ Use the **Insert OR** button to add the OR operation in the query.
- ❑ Use the **X** button to remove the selected operation or expression from the query.
- ❑ Use the **Outermost operation** radio button to specify the outermost operation as AND or OR.

**Note:** In case you want to write or edit the query as free text, click the *Edit as free text* button. This will open a dialog box with a text box in which you can make modifications to the query, as required.

### 3. Click *OK* once you are done with building your query.

After you click OK, the Search composer dialog box is closed and the items that match the search criteria are highlighted in the verification hierarchy tree, as shown in [Figure 2-31 on page 52](#).

**Figure 2-31 Search Results**



Similarly, you can create more customized queries.

### **Advanced within selection**

The functionality of *Advanced within selection* is same as *Advanced* with a difference that in this case, the custom query applies only to the nodes that are selected in the vPlan hierarchy tree.

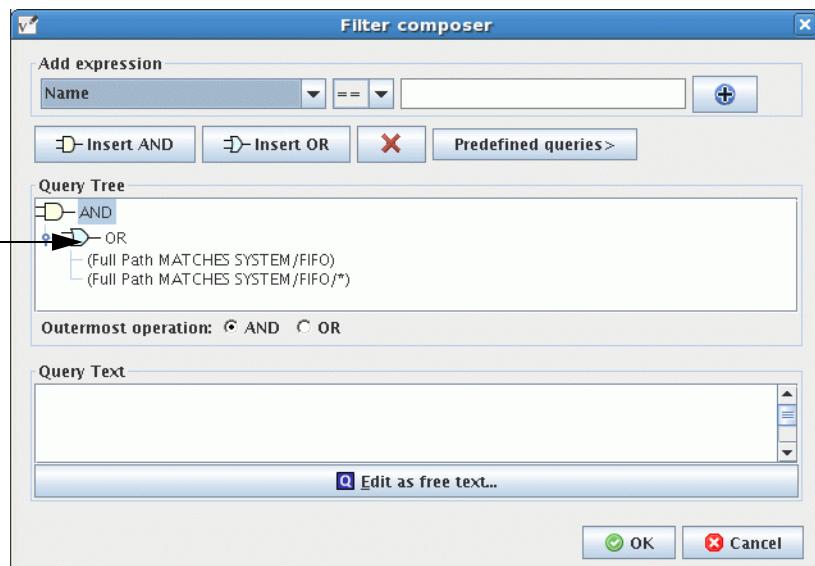
For example, you want to apply search only within FIFO and search the elements where owner is Bob. For this:

1. Select *FIFO* in the vPlan hierarchy.
2. Click the *More search options* button and select *Advanced within selection* from the pop-up menu.

A *Search composer* dialog box appears, as shown in [Figure 2-32](#) on page 53.

**Figure 2-32 Search composer (Advanced within selection)**

Automatically generated expression terms



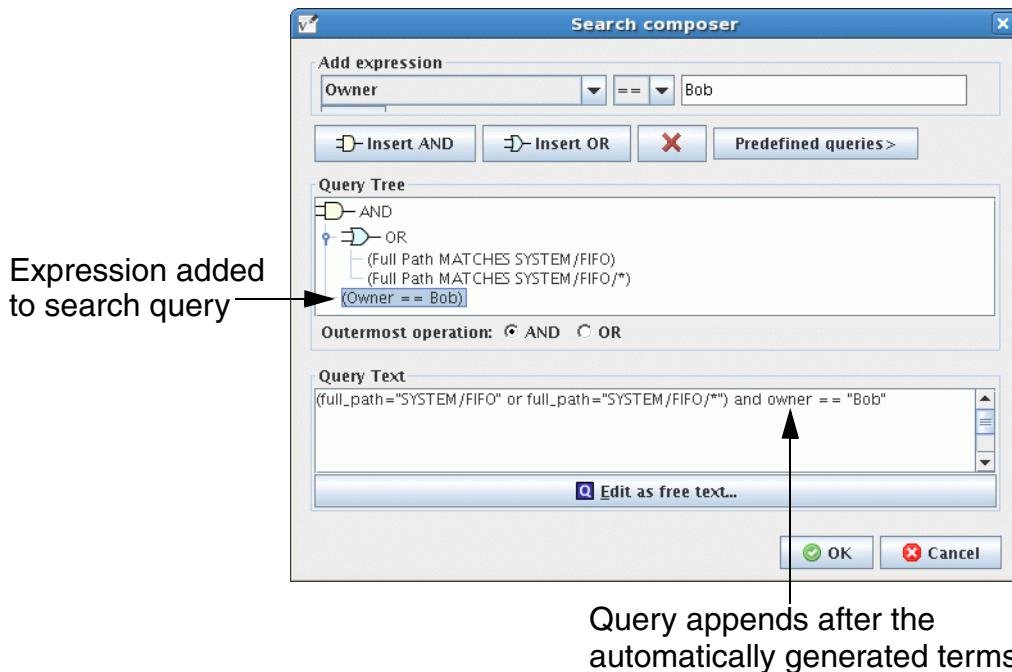
When you select *Advanced within selection*, the *Query Tree* box in the Search composer dialog box shows automatically generated terms. These terms are automatically generated by Planner based on the nodes selected in the vPlan hierarchy.

**Note:** You should not modify or tamper the automatically generated terms. Any modification to automatically generated terms will mess up the query and will not generate the required search or filter results. The query that you create using the Search composer dialog box will be appended after the automatically generated terms.

3. In the *Add expression* section:

- a. Select the attribute on which the search must apply. For example, select *Owner* from the drop-down list.
  - b. Select the operator for creating the expression. For example, select  $\text{==}$  from the operator drop-down list.
  - c. Specify the value of the attribute for creating the expression. For example, specify *Bob* in the text box.
  - d. Click the + button to add the expression to the Query Tree.
4. When you click the + button, the expression is added to the *Query Tree* and is also shown in the *Query Text* text box, as shown in [Figure 2-33](#) on page 54.

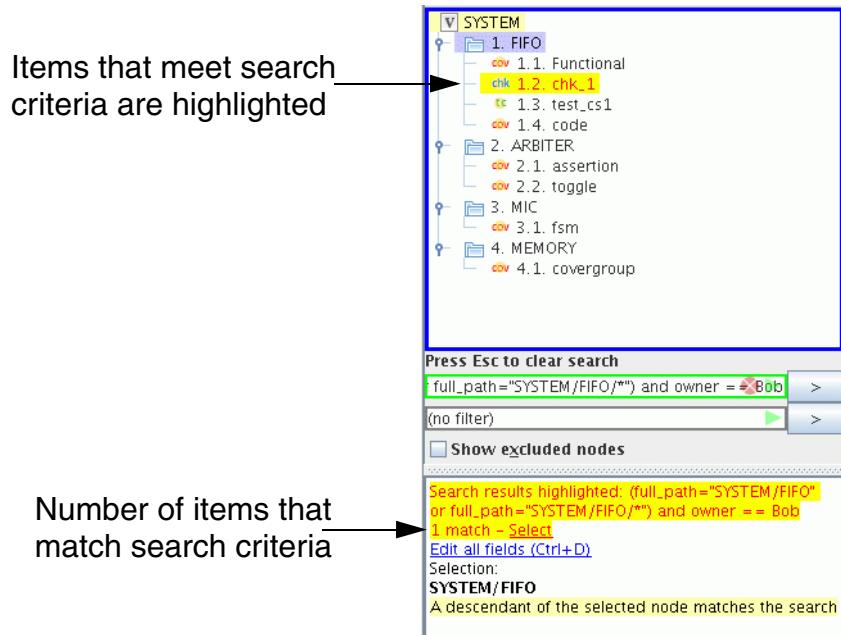
**Figure 2-33** Search composer



5. Click *OK* after completing the query.

The *Search composer* dialog box is closed and the items that match the search criteria are highlighted in the verification hierarchy tree, as shown in [Figure 2-34](#) on page 55.

**Figure 2-34 Search Results**



Similarly, you can create more customized queries.

## 2.10 Creating Perspectives

Using vPlanner, you can specify features that are associated with a particular component, project team, or project milestone by creating a perspective. For example, you can create two perspectives, Milestone 1 and Milestone 2, with exactly same features but with different coverage goals. This helps you to review the project status from different perspectives and also helps you better organize the plan.

When you select a node in the verification plan, you will find the following options to create perspectives:

- New perspective—To create a perspective that will include all the nodes from the verification plan. You can add filters later, as required.
- New perspective from selection—To create a perspective from the selected nodes in the verification plan.
- New filter-based perspective—To create a perspective that will start with the all the nodes of verification plan and while creating the perspective, you specify a query to filter the items that will appear in the perspective.

- New filter-based perspective from selection —To create a perspective that will start with the selected nodes of the verification plan and while creating the perspective, you specify a query to filter the items that will appear in the perspective.

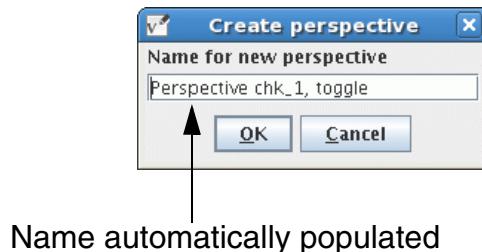
### 2.10.1 Example: Creating Perspective From Selection

To create a perspective from selected nodes:

1. Select the vPlan items that should be part of a perspective. For example, select *chk\_1* and *toggle*.
2. Right-click and select *New perspective from selection*.

The *Create perspective* dialog box appears, as shown in [Figure 2-35](#) on page 56.

**Figure 2-35 Create Perspective**

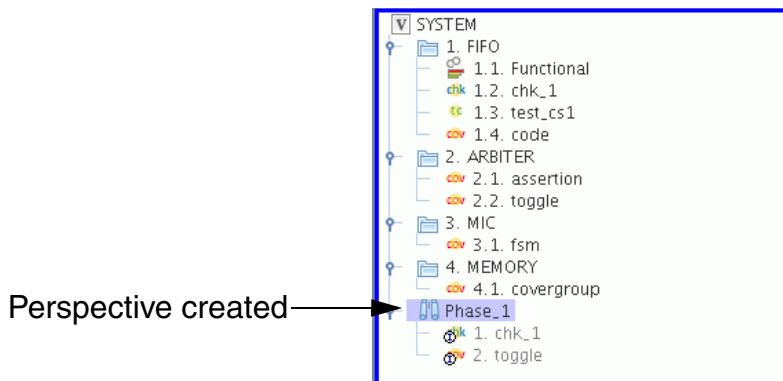


The name text box is automatically populated with a name based on your selection.

3. You can specify an alternate name for the perspective, as required. For example, specify *Phase\_1* in the text box, and click *OK*.

After you click *OK*, the perspective appears in the vPlan hierarchy, as shown in [Figure 2-36](#) on page 57.

**Figure 2-36 Perspective Created**



Notice that the elements under the perspective are disabled. You can edit the plan elements only in original location. Any change to the plan element (at its original location) is automatically reflected instantly in the perspective. A symbol  $\perp$  on an icon indicates that this section or port is actually instantiation of the original section or port in the vPlan.

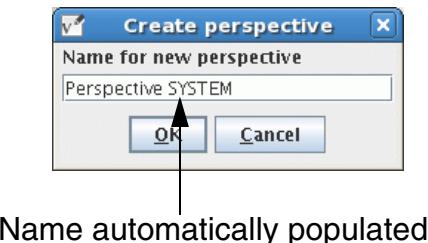
### 2.10.2 Example: Creating Filter-based Perspective

To create a filter-based perspective:

1. Select any of the vPlan items in the vPlan editor.
2. Right-click and select *New filter-based perspective*.

The *Create perspective* dialog box appears, as shown in [Figure 2-37](#) on page 57.

**Figure 2-37 Create Perspective**

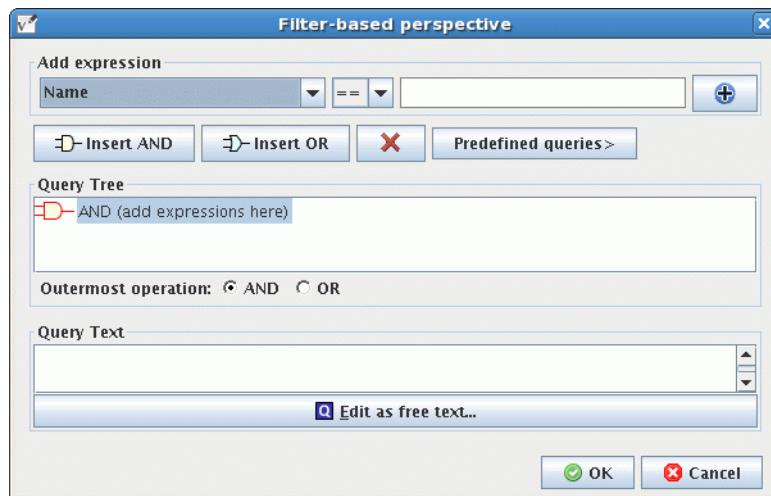


The name text box is automatically populated with a name based on your selection.

3. You can specify an alternate name for the perspective, as required. For example, specify *Phase1\_mike* in the text box, and click *OK*.

After you click *OK*, the *Filter-based perspective* dialog box appears, as shown in [Figure 2-38](#) on page 58.

**Figure 2-38 Filter-based Perspective**

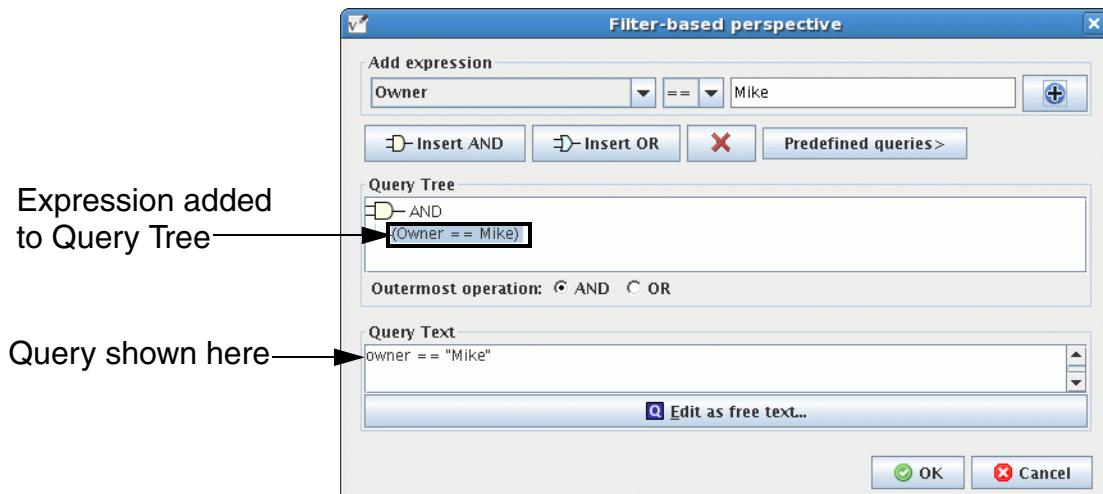


4. In the *Add expression* section:

- a. Select the attribute on which the filter must apply. For example, select *Owner* from the drop-down list.
- b. Select the operator for creating the expression. For example, select  $\text{==}$  from the operator drop-down list.
- c. Specify the value of the attribute for creating the expression. For example, specify *Mike* in the text box.
- d. Click the + button to add the expression to the Query Tree.

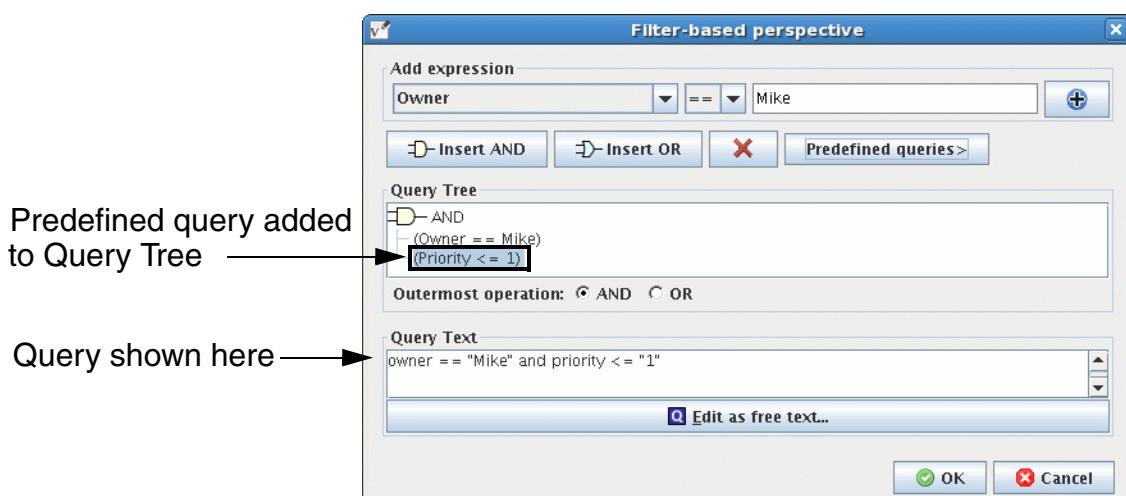
When you click the + button, the expression is added to the *Query Tree* and is also shown in the *Query Text* text box, as shown in [Figure 2-39](#) on page 59.

**Figure 2-39 Filter-based Perspective**



**Note:** You can also add expressions to the *Query Tree* using a list of predefined queries. The list of predefined queries is visible when you click the *Predefined queries* button. When you select a query from the list of predefined queries, it gets added to the *Query Tree*. For example, to further build the query where *Owner* is *Mike* and *priority* is  $\leq 1$ , click the *Predefined queries* button and select *Priority  $\leq 1$*  from the list. The query is added to the *Query Tree*, as shown in [Figure 2-39](#) on page 59.

**Figure 2-40 Filter-based Perspective**



While building the query, you can:

- ❑ Use the *Insert AND* button to add the AND operation in the query.
- ❑ Use the *Insert OR* button to add the OR operation in the query.

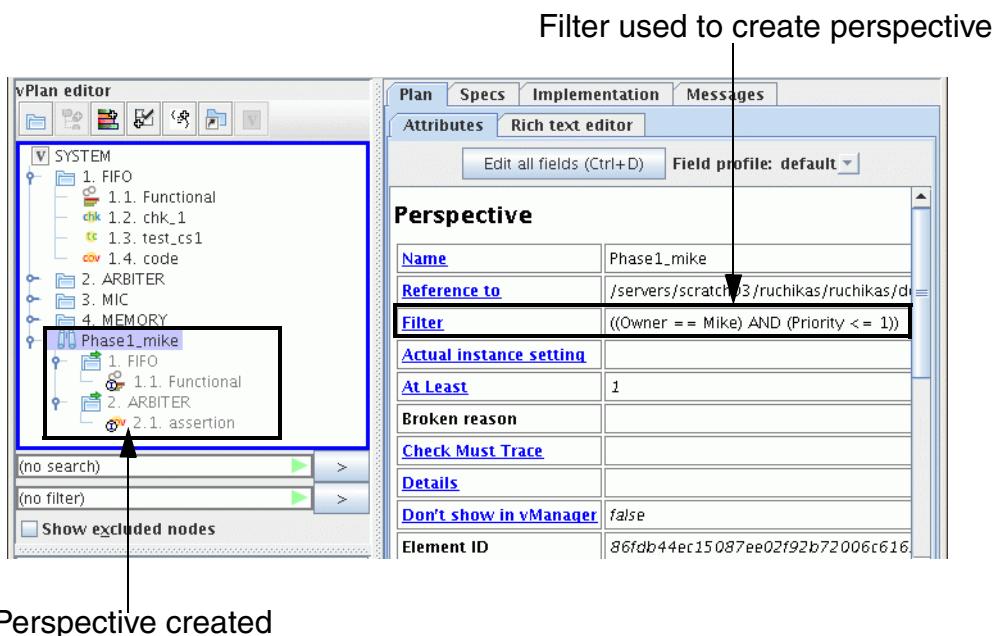
- Use the **x** button to remove the selected operation or expression from the query.
- Use the *Outermost operation* radio button to specify the outermost operation as AND or OR.

**Note:** In case you want to write or edit the query as free text, click the *Edit as free text* button. This will open a dialog box with a text box in which you can make modifications to the query, as required.

5. Click **OK** once you are done with building your query.

After you click OK, the perspective appears in the vPlan hierarchy, as shown in [Figure 2-41](#) on page 60.

**Figure 2-41 Perspective (Filter-Based) Created**



Notice that the elements under the perspective are disabled. You can edit the plan elements only in original location. Any change to the plan element (at its original location) is automatically reflected instantly in the perspective.

### 2.10.3 Deleting Perspectives

To delete a perspective, do any of the following:

- Select the perspective that you want to delete and press the **Del** key.
- Right-click the perspective that you want to delete and select *Delete*.

A confirmation dialog appears and once you confirm the deletion, it is deleted from the vPlan hierarchy.

**Note:** You can delete multiple perspectives at a time by selecting them (to select multiple sections, use Shift+Select or Ctrl+Select) in the vPlan hierarchy.

### 2.10.4 Renaming Perspectives

To rename a perspective:

1. Do one of the following:
  - Select the perspective you want to rename and press the F2 key on the keyboard.
  - Right-click the perspective you want to rename and select *Rename*.
  - Click the *Name* field in the *Attributes* tab.
2. Specify the new name and click *OK*.

## 2.11 CSV-Based Verification Plans

vPlanner allows you to import CSV-based plans created in Microsoft Excel, OpenOffice, or any other tool that can generate a CSV (comma separated value) output.

This section describes:

- [Creating a CSV-Based Verification Plan](#)
- [Embedding or Opening a CSV-Based Verification Plan in vPlanner](#)

### 2.11.1 Creating a CSV-Based Verification Plan

To create a CSV-based plan, ensure that:

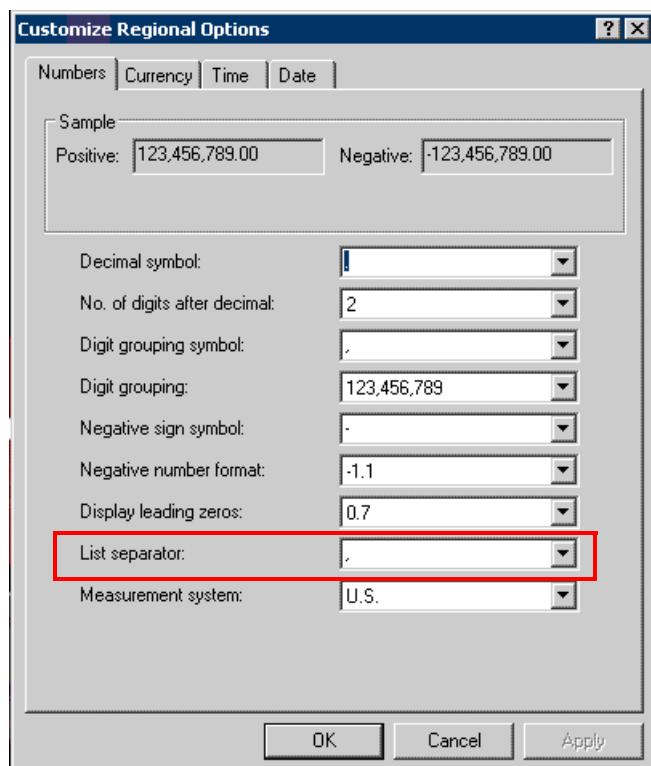
- A comma ( , ) is used as a separator.
- Correct attributes are used in the plan as column titles.

#### 2.11.1.1 Setting Comma (,) as the Separator

In some European locales, by default, the ";" is used as a separator instead of ",". If your default is ";", you can change it. You can change the default separator by configuring it in the Windows Control Panel as mentioned below:

1. Click the *Start* button in the taskbar.
2. Select *Settings* —>*Control Panel*.
3. Double-click *Regional and Language Options*.
4. The *Regional and Language Options* dialog box appears. Click *Customize* (in the case of Windows XP) or *Additional Settings* (in the case of Windows 7).
5. The *Customize Regional Options* dialog box appears, as shown in [Figure 2-42](#) on page 62. Ensure that ‘,’ is used as a List separator.

**Figure 2-42 Customize Regional Options**



**Note:** In the case of Windows 7, *Customize Format* dialog box is shown.

6. Click *OK* to close the dialog box.
7. Click *OK* to close the *Regional and Language Options* dialog box.

This will set the separator as a comma.

### 2.11.1.2 List of Attributes

In a CSV-based plan, there are two categories of attributes that can be used:

- Attributes that allow you to define the structure of the plan.
- Attributes that define other details such as owner, priority, and weight of the vPlan element.

[Table 2-3](#) on page 63 lists the attributes that help you define the structure of the plan.

**Table 2-3**

Attribute/Column Name	Description
NAME	<p>It is the only mandatory column that must exist. It specifies the name of the vPlan element or the coverage definition.</p>
DEPTH	<p>It defines the levels at which each vPlan element must show in the vPlan hierarchy. In the DEPTH column:</p> <ul style="list-style-type: none"> <li>■ 0 indicates top-level elements.</li> <li>■ 1 indicates immediate children of elements at level 0.</li> <li>■ 2 indicates items nested below items at level 1.</li> </ul> <p><b>Note:</b> Wrong DEPTH leads to errors and prevents the vPlan from loading in vPlanner. The DEPTH column is mutually exclusive with the NUMBERING column.</p>
NUMBERING	<p>This column also defines the levels at which each vPlan element must show in the vPlan hierarchy. It is mutually exclusive with the DEPTH column. In the NUMBERING column:</p> <ul style="list-style-type: none"> <li>■ 1 indicates top-level elements.</li> <li>■ 1.1 , 1.2, and so on indicates immediate children of elements at level 1.</li> <li>■ 1.1.1 1.1.2, and so on indicates items nested below items at level 1.1.</li> </ul> <p><b>Note:</b> Wrong NUMBERING leads to errors and prevents the vPlan from loading in vPlanner.</p>

Attribute/Column Name	Description
NODE_KIND	<p>It specifies the kind of node. Valid options for this column are:</p> <ul style="list-style-type: none"> <li>■ SECTION — to create an element of type section.</li> <li>■ PLANNED_COVERAGE or COV— to create an element of type coverage.</li> <li>■ PLANNED_CHECK or CHK— to create an element of type check.</li> <li>■ PLANNED_TESTCASE or TC— to create an element of type testcase.</li> </ul> <p><b>Note:</b> If no NODE_KIND is specified, then by default, SECTION is assumed as the kind of node (if the MAPPING_PATTERN is not defined). If MAPPING_PATTERN is specified, then COVERAGE is assumed.</p>
MAPPING_PATTERN	<p>It specifies the coverage definition of the vPlan element.</p> <p>If MAPPING_PATTERN is not specified, no mappings are applied to the vPlan element.</p> <p>If a MAPPING_PATTERN is specified, then it represents the coverage definition of the vPlan element and when you open or import the plan in vPlanner, these mappings are applied based on the pattern specified.</p>
ELEMENT_ID	<p>In the case of a CSV-based plan, synchronization might not work correctly until you add a column element_id in the CSV-based plan. This column must include a unique identifier for each element defined in the plan. Setting a value in the element_id column ensures that any mappings or annotations made to the vPlan element after importing the plan are not lost at the time of synchronization. For more details, see <a href="#">Embedding or Opening a CSV-Based Verification Plan in vPlanner</a> on page 69.</p>
METRICS_PORT_KIND	<p>Lets you define the type of port. The valid values are COVERAGE, TESTCASE, and CHECK.</p>

Apart from using the attributes defined in [Table 2-3](#) on page 63, you can also use any of the following attributes as columns:

- implementation\_note
- goal
- details
- owner
- verification\_scopes
- priority
- weight
- at\_least

Unrecognized columns are automatically ignored by vPlanner.

### 2.11.1.3 Sample CSV File

To ensure that the CSV-based verification plan imports or opens successfully in vPlanner, it must be in a specific format.

[Figure 2-43 on page 65](#) shows a sample verification plan in a CSV format.

**Figure 2-43 Verification Plan in CSV Format**

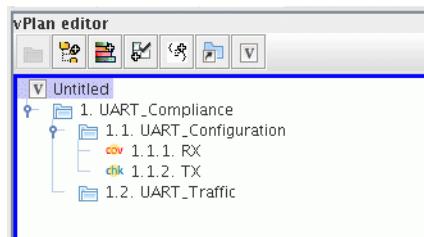
The diagram illustrates a CSV file structure for a verification plan. The first row is labeled 'Attributes' and contains three columns: A (NAME), B (DEPTH), and C (NODE\_KIND). The second row contains data: NAME 'UART\_Compliance', DEPTH '0', and NODE\_KIND 'SECTION'. The third row contains data: NAME 'UART\_Configuration', DEPTH '1', and NODE\_KIND 'COV'. The fourth row contains data: NAME 'RX', DEPTH '2', and NODE\_KIND 'CHK'. The fifth row contains data: NAME 'TX', DEPTH '2', and NODE\_KIND 'CHK'. The sixth row contains data: NAME 'UART\_Traffic', DEPTH '1', and NODE\_KIND 'CHK'. Arrows point from the labels to their respective parts: 'Attributes' points to the header row, 'vPlan elements' points to the data rows, 'Defines the hierarchy' points to the DEPTH column, and 'Defines the type of element' points to the NODE\_KIND column.

NAME	DEPTH	NODE_KIND
UART_Compliance	0	SECTION
UART_Configuration	1	COV
RX	2	CHK
TX	2	CHK
UART_Traffic	1	CHK

The first row of the CSV file must specify the attributes of the vPlan element. For example, in this case, NAME, DEPTH, and NODE\_KIND are the attributes of the vPlan elements that are shown in rows that follow.

[Figure 2-44 on page 66](#) shows how the above plan displays in vPlanner.

**Figure 2-44 Verification Plan in vPlanner**



Notice that the depth attribute has defined the levels based on the values specified in the column. *UART\_Compliance* is at top-level. It has two child sections: *UART\_Configuration* and *UART\_Traffic*. The *UART\_Configuration* section has two nested items: *RX* and *TX*.

**Note:** In this case, if NUMBERING was defined instead of DEPTH, then *UART\_Compliance* should be assigned 1. *UART\_Configuration* should be assigned 1.1 and *UART\_Traffic* must be assigned 1.2. As *UART\_Configuration* section has two nested items: *RX* should be assigned 1.1.1 and *TX* should be assigned 1.1.2.

For details on opening a CSV-based plan in vPlanner, see [Opening a CSV-based Verification Plan on page 69](#).

### 2.11.1.4 Example: Using Additional Attributes in a CSV-Based Plan

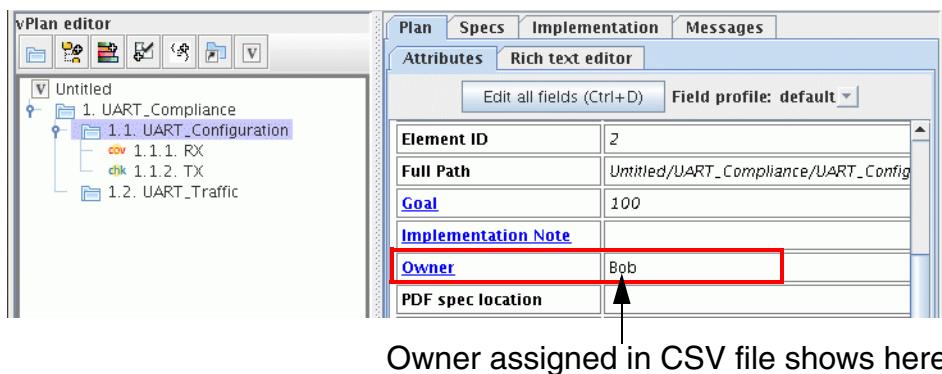
Consider the CSV file shown in [Figure 2-45 on page 66](#).

**Figure 2-45 Verification Plan in CSV Format**

	A	B	C	D
1	NAME	DEPTH	NODE_KIND	owner
2	UART_Compliance	0		
3	UART_Configuration	1	SECTION	Bob
4	RX		2 COV	
5	TX		2 CHK	
6	UART_Traffic	1		Mary

An additional attribute *owner* is added to the plan. The values assigned in this column will now show in vPlanner, as shown in [Figure 2-46 on page 67](#).

**Figure 2-46 Additional Attribute Value in vPlanner**



The section owner assigned in the *owner* column of the CSV-based plan now shows in the vPlanner.

### 2.11.1.5 Example: Using a Different Column Name

When defining attributes in a CSV-based plan, instead of using the default name of the attribute as a column name, you can choose to use a different name. For this, create a file named `csv_columns_map` and map the new name to the default column name.

The syntax for assigning a new name to the default name is:

```
<default_name>=<new_name>
```

For example, you want to use a different name for following columns:

- Use `importance` as the column name instead of default name `priority`
- Use `MyOwn` as the column name instead of default name `owner`

To do this, create a file named `csv_columns_map` and specify the following in the file:

```
priority=importance
owner=MyOwn
```

Use the new names as the column names in the CSV file, as shown in [Figure 2-47](#) on page 68.

**Figure 2-47 Verification Plan in CSV Format using Different Column Names**

New names specified as column names

A	B	C	D	E
1 NAME	DEPTH	NODE_KIND	MyOwn	importance
2 UART_Compliance	0			2
3 UART_Configuration	1	SECTION	Bob	3
4 RX	2	COV		1
5 TX	2	CHK		1
6 UART_Traffic	1		Mary	2

**Note:** The new names specified in the CSV-based plan are recognized by vPlanner only if the path where the file `csv_columns_map` is saved is included in the `VMANAGER_PATH`. The vPlanner searches for file `csv_columns_map` in the directories listed in `VMANAGER_PATH` and the first occurrence is used.

For example, the `csv_columns_map` file is stored at the following location:

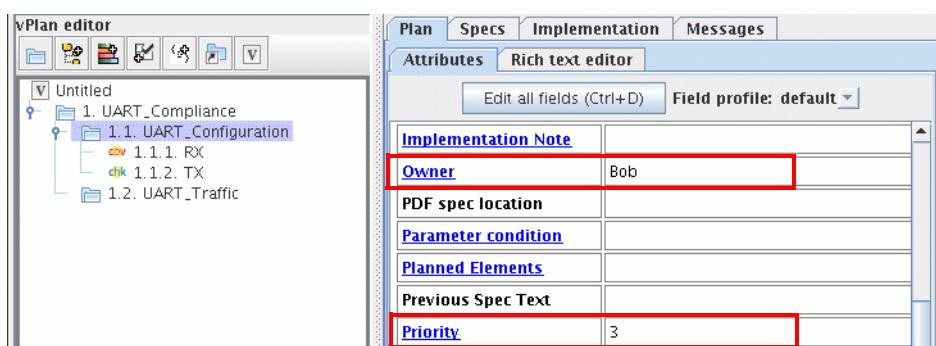
```
/home/bob/ep/
```

You must include this path in the `VMANAGER_PATH` using the following command:

```
setenv VMANAGER_PATH /home/bob/ep/
```

After the path of the `csv_columns_map` file is included in the `VMANAGER_PATH`, the new column names are recognized by vPlanner, as shown in [Figure 2-48](#) on page 68.

**Figure 2-48 New Column Names Recognized by vPlanner**



In the above figure, the *Priority* field and the *Owner* field show the values set in columns `importance` and `MyOwn`, respectively.



Features, such as parameter declarations, instantiations, and perspectives are not supported in a CSV-based plan. You can use these features only after opening or importing the CSV-based plan in vPlanner.

### 2.11.2 Embedding or Opening a CSV-Based Verification Plan in vPlanner

After creating a verification plan in CSV format, you can either open it or embed it in vPlanner.

**Note:** A plan exported in a CSV format using vPlanner cannot be opened or embedded in vPlanner.

#### 2.11.2.1 Opening a CSV-based Verification Plan

To open the CSV-based plan in vPlanner:

1. Select *File* —> *Open*.

Alternatively, you can click the *Open* button on the toolbar.

2. The *Open vPlan* dialog box is displayed. Navigate through the hierarchy and select the relevant verification plan and click *Open*.

#### 2.11.2.2 Embedding and Synchronizing a CSV-based Verification Plan

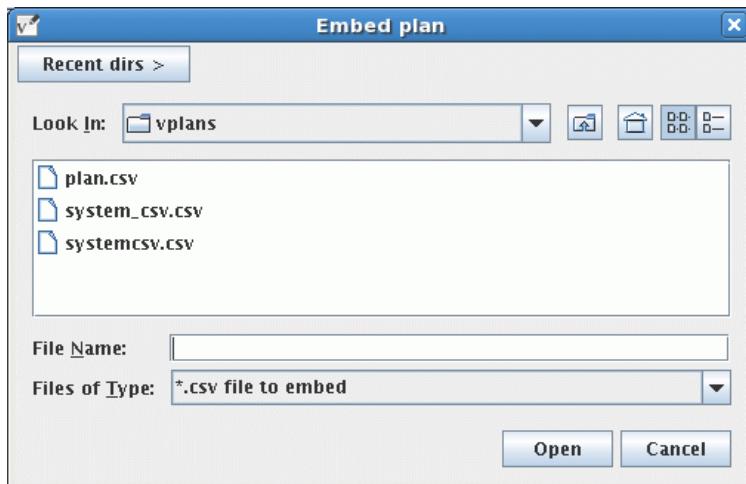
Instead of opening a CSV-based plan, you can also embed it. Embedding helps you in situations where the underlying plan is likely to change. In such cases, you can embed the plan and later synchronize it.

##### **Embedding a CSV-based Plan**

To embed the CSV-based plan in vPlanner:

1. Do any of the following:
  - Click the *Embed foreign vPlan* button , or
  - Right-click the vPlan and select *Embed foreign vPlan* from the pop-up menu.
2. The *Embed plan* dialog box appears, as shown in [Figure 2-49](#) on page 70. Select the plan that you want to embed and click *Open*.

**Figure 2-49 Embed Plan**



3. A dialog box appears, as shown in Figure 2-50 on page 70. The dialog box, by default shows the name of the plan. If you want to specify a different name for the plan, specify it in the text box, and click *OK*.

**Figure 2-50 Embed Plan**



The embedded plan is now available as a copy in the existing plan.

### **Synchronizing a CSV-based Plan**

In the case of a CSV-based plan, synchronization might not work correctly until you add a column `element_id` in the CSV-based plan. This column must include a unique identifier for each element defined in the plan. The unique identifier can be any number, character, word, or a combination of numbers and characters.

Setting a value in the `element_id` column ensures that:

- Any change, such as renaming an element, moving it, or deleting any of the elements is recognized and is available correctly after synchronization.
- Any mappings or annotations made to the vPlan element after importing the plan are not lost at the time of synchronization.

Consider the following sequence of steps:

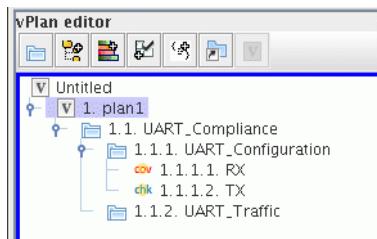
1. Create a plan in a CSV format, as shown in [Figure 2-51](#) on page 71.

**Figure 2-51 Verification Plan in CSV Format**

A	B	C	D	E
NAME	DEPTH	NODE_KIND	MyOwn	importance
1 UART_Compliance	0			2
3 UART_Configuration	1	SECTION	Bob	3
4 RX	2	COV		1
5 TX	2	CHK		1
6 UART_Traffic	1		Mary	2

2. Embed the plan in vPlanner. [Figure 2-52](#) on page 71 shows the embedded plan.

**Figure 2-52 Plan Embedded in vPlanner**

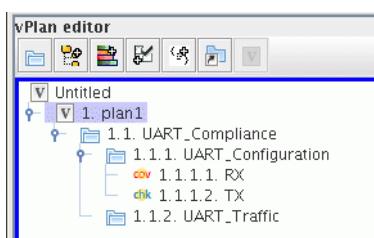


3. Rename *UART\_Configuration* as *UART\_Config* in CSV format and save it.
4. Click *Sync Embedded* button on the toolbar to make changes available in vPlanner.

**Note:** Alternatively, right-click the plan node in the vPlan editor and select *Synchronize embedded vPlan* or *Synchronize all embedded plans* from the pop-up menu.

[Figure 2-53](#) on page 71 shows the plan in vPlanner after synchronization.

**Figure 2-53 Verification Plan Shows Incorrectly After Synchronization**



Notice that *UART\_Configuration* is renamed as *UART\_Config*, but the plan incorrectly shows all items as flagged. This is misleading and the user cannot identify what exactly changed after synchronization.

To ensure that synchronization works correctly for a CSV-based plan:

1. Consider adding the `element_id` column in the CSV-based plan, as shown in [Figure 2-54 on page 72](#).

**Figure 2-54 Verification Plan with element\_id Column**

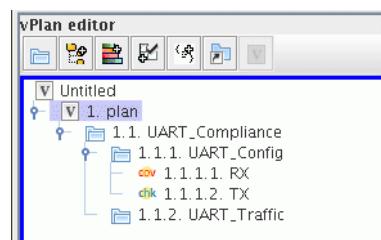


	A	B	C	D	E	F
1	NAME	DEPTH	NODE_KIND	MyOwn	importance	element_id
2	UART_Compliance	0			2	1
3	UART_Config	1	SECTION	Bob	3	2
4	RX	2	COV		1	3
5	TX	2	CHK		1	4
6	UART_Traffic	1		Mary	2	5

**Note:** The `element_id` column can contain any value and in any order. The only rule to be followed is that the value must be unique.

2. Embed the plan in vPlanner. [Figure 2-55 on page 72](#) shows the embedded plan.

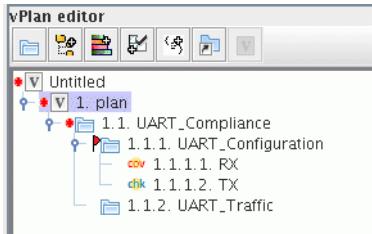
**Figure 2-55 Verification Plan Embedded in vPlanner**



3. Rename *UART\_Config* as *UART\_Configuration* in CSV format and save it.
4. Click *Sync Embedded* button on the toolbar to make changes available in vPlanner.

[Figure 2-56 on page 73](#) shows the plan in vPlanner after synchronization.

**Figure 2-56 Verification Plan Shows Correctly After Synchronization**



Notice that *UART\_Config* is now correctly renamed as *UART\_Configuration* in the plan after synchronization.

You can now click the flagged item, check what has changed, and clear the flag, as required.

**Note:** As stated earlier, the `element_id` column plays a key role in situations when after embedding the plan you add some mappings or annotations and you do not want to lose them at the time of synchronization. If the `element_id` column is not used all the mappings and annotations are lost at the time of synchronization.



When creating a CSV-based plan, the `NAME` attribute is the only mandatory attribute. You can add attributes such as, `DEPTH`, `NODE_KIND`, and so on. When you embed the CSV-based plan in the vPlanx file, syncronize the changes (if any) and save it in vPlanner, additional attributes get saved to the CSV file.

For example, consider the following CSV file:

```
NAME,DEPTH,NODE_KIND
FIFO,0,SECTION
TX,1,PLANNED_COVERAGE
RX,1,PLANNED_TESTCASE
```

Embed this file in a verification plan in vPlanner. After you embed the CSV file into another vPlan (.vplanx), you can change it either in the vPlanner or directly in the CSV file. You need to press the *Sync Embedded* button and the *Save* button each time the changes are done. When you click the *Save* button, the CSV file is saved and new attributes are also added to the CSV file as shown below:

```
NAME,DEPTH,NODE_KIND,MAPPING_PATTERN,element_id,metrics_port_kind
FIFO,0,SECTION,,e1cc6ca9-b23b-4c8a-a272-ee8b214f6ff2,
TX,1,PLANNED_COVERAGE,,dbbc9fc5-4723-4d82-852a-8fa4f3f568f9,COVERAGE
RX,1,PLANNED_COVERAGE,,50177480-22c9-4f2d-ab09-022f8f120c65,COVERAGE
```

Notice that additional attributes, such as `MAPPING_PATTERN`, `element_id`, and `metrics_port_kind` are added to the CSV file. These newly added attributes are important for the Embed Synchronization flow to work correctly.

### ***Resolving Conflicts in Embedded vPlans***

After you embed a CSV-based plan into another vPlan (.vplanx), you can change it in either vPlanner or directly in the CSV file. When you click the *Sync Embedded* button, both the CSV file and the embedded vPlan are updated by the tool. The vPlanx plan stores both -- the current state of the plan and the base version of the foreign plan (as it was after the last synchronization operation).

At the time of synchronization, if an element was modified both in the vPlanner and also in the foreign tool there is a conflict. Conflicts are resolved based on the policy defined in the user configuration. It can be any of the following:

- **BASE\_VERSION**: If 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**: If you want vPlanner to use the values from the top-level plan in the event of synchronization conflict.
- **FOREIGN\_PLAN**: If you want vPlanner to use the values from the embedded plan in the event of synchronization conflict. This is the default behavior.

## **2.12 Creating and Using vPlan User-defined Attributes**

User-defined attributes allow you to add attributes and use it for a specific purpose. For example, you might want to add a new attribute that reflects the amount of time required in order to complete the testing on a vPlan entity.

This section describes the following:

- [Creating User-defined Attributes](#)
- [Using the User-defined Attributes](#)

**Note:** Steps for importing, deleting, and exporting user-defined vPlan attributes is same as vManager user-defined session and runs attributes. For more details, see [Incisive vManager User Guide](#).

## 2.12.1 Creating User-defined Attributes

To create user-defined attributes:

1. Create a .csv file with required fields or columns. You can also copy the vm\_attributes.csv file available in the install directory (<install-dir>/vmanager/ep\_lib /vm\_attributes.csv) and modify it, as required.

**Note:** The file vm\_attributes.csv is a file that was used earlier by EMGR for creating and maintaining user-defined attributes. For description fields mentioned in vm\_attributes.csv file, see [vPlan Related Fields in vm\\_attributes.csv File](#) on page 75.

2. Add the attribute as a record in the .csv file, based on your requirements. For example, a new attribute RJP\_INT is added as a record in the .csv file, as shown in [Figure 2-57](#) on page 75.

**Figure 2-57 User-Defined Attribute Added**

	A	B	C	D	E	F	G	H
1	NAME	DISPLAY_NAME	DESCRIPTION	TYPE	CALCULATION_SCRIPT	IS_GRADE_ATTRIBUTE	VALID_CONTAINER_NAMES	CONTEXT_NAME
2	RJP_INT	RJP_INT	RJP INT Attribute	ATTRIBUTE_TYPE_INT		FALSE	section;instantiate;metricport	default

**Note:** The user-defined attribute is added as a record/row in the .csv file.



The attributes for which a CALCULATION\_SCRIPT is specified, are not available in vPlanner.

3. Save the .csv file.

### vPlan Related Fields in vm\_attributes.csv File

[Table 2-4](#) on page 76 describes the meaning of each field related to vPlan in vm\_attributes.csv file.

**Table 2-4 Description of Fields in vm\_attributes.csv**

Field Name	Description	Possible Values	Default Value
NAME	The name of the attribute.	Any string (No spaces and special characters)  It can also accept values that match this regular expression: [_a-zA-Z] [0-9_a-zA-Z]*	None
DISPLAY_NAME	The name that is displayed in vManager.	Any string	Empty
DESCRIPTION	The description of the attribute.	Any string	Empty
VALID_CONTAINER_NAMES	The containers that can hold this attribute.	One or more of the following containers separated with a semicolon:  ■ section ■ instantiate ■ metricport  The list must end with a semi-colon.	Empty

## vPlanner User Guide

---

Field Name	Description	Possible Values	Default Value
TYPE	The type of this attribute.	Any one of the following: <ul style="list-style-type: none"> <li>■ ATTRIBUTE_TYPE_INT</li> <li>■ ATTRIBUTE_TYPE_DOUBLE</li> <li>■ ATTRIBUTE_TYPE_STRING</li> <li>■ ATTRIBUTE_TYPE_ENUM</li> <li>■ ATTRIBUTE_TYPE_DATETIME</li> <li>■ ATTRIBUTE_TYPE_DURATION</li> </ul>	ATTRIBUTE_TYPE_STRING
DEFAULT_VALUE	The default value of this attribute.	Any string holding a default value	Empty
SHOULD_PROPAGATE	Whether this attribute is propagated from the parent container to the children container.	TRUE, FALSE	TRUE
VALID_VALUES	Set the valid values of an attribute	Any String <b>Note:</b> The enum values must be separated with a semi-colon (;). Example -- ATTRIBUTE_TYPE_ENUM X;Y;Z	Empty (means all values are valid)
IS_GRADE_ATTRIBUTE	Set if the attribute should be viewed as grade	TRUE, FALSE	FALSE
CALCULATION_SCRIPT	Allows you to specify the path of a vPlan visitor script file.	Path of the vPlan visitor file	Empty

### 2.12.2 Using the User-defined Attributes

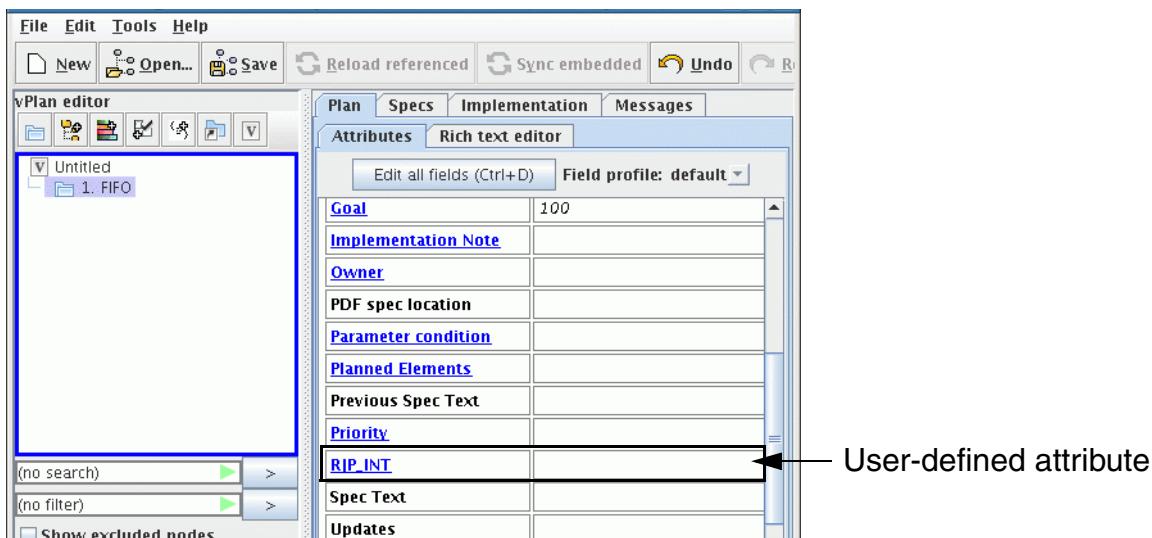
After saving the user-defined attribute in the CSV file, you can use it, as required.

To use the user-defined attributes, perform the following steps:

1. Enable showing of user-defined attributes in vPlanner. For this, do any of the following:
  - ❑ Open `vmgrprofile -edit` and import the `.csv` file that contains the user-defined attributes. (For details, see [Incisive vManager User Guide](#).)
  - ❑ Set `VMANGER_PATH` to the location that contains the `vm_attributes.csv` file that contains the user-defined attributes.
2. You can now launch vPlanner and use the attribute, as required. Launch vPlanner.
3. Create a node.

You will now see the user-defined attribute along with the list of other attributes in the *Attributes* pane, as shown in [Figure 2-58](#) on page 78.

**Figure 2-58 User-Defined Attribute Available in vPlanner**



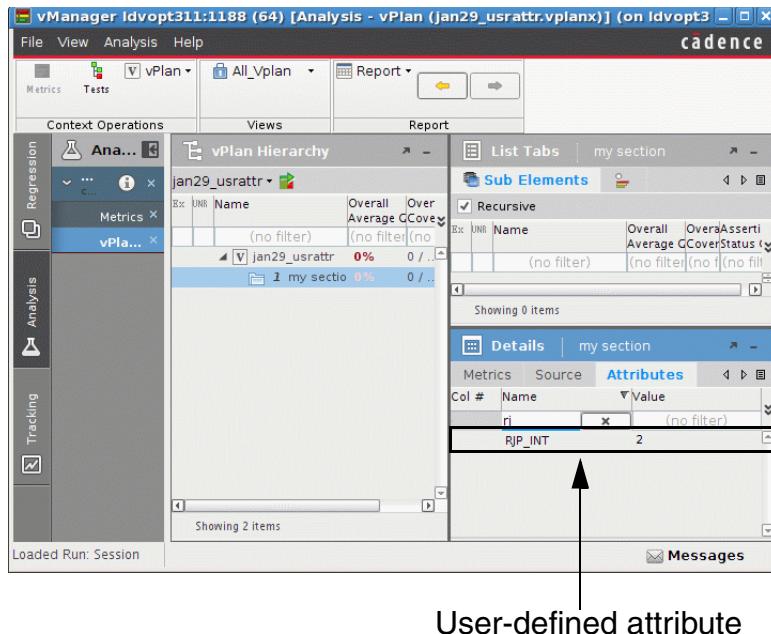
The user-defined attribute is also available in the list of predefined attributes. You can set the value of the attribute, create more metric ports in the plan, add perspectives, apply mappings, and so on.

The user-defined attributes are also available in vManager when you load the vPlan in vManager.

## vPlanner User Guide

Figure 2-59 on page 79 shows the user-defined attribute in vManager.

**Figure 2-59 User-Defined Attribute Shown in vManager**



For more details, see [Incisive vManager User Guide](#).

## vPlanner User Guide

---

---

## Annotating and Updating Verification Plans

---

Using vPlanner, you can associate specification documents with the verification plans and create annotations. In addition, you can also compose and add rich-text data to vPlan elements.

This chapter discusses the following topics:

- [Associating Specification Document with a Plan](#)
- [Creating and Modifying Annotations](#)
- [Updating the Specification Document](#)
- [Rich-text Editing](#)

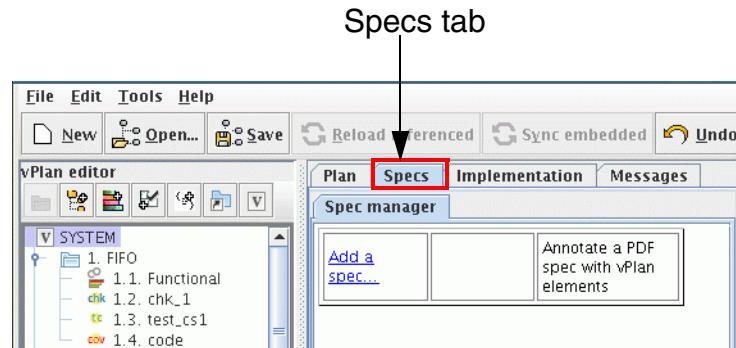
### 3.1 **Associating Specification Document with a Plan**

If product specification documents are available in a PDF format, or can be ported to a PDF format, you can highlight text from one or more documents, and add this text as an annotation to elements in a verification plan.

To associate a specification document with a verification plan:

1. Select the *Specs* tab, as shown in [Figure 3-1](#) on page 82.

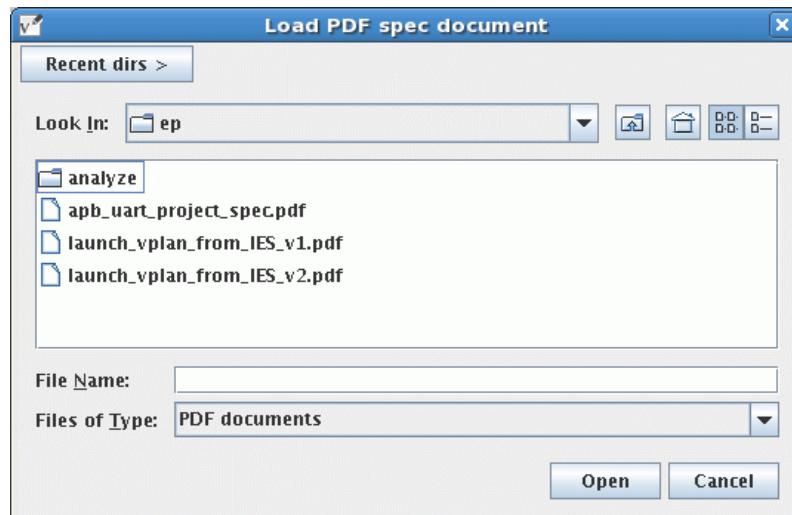
**Figure 3-1 Specs Tab**



2. Click the *Add a spec...* link.

The *Load PDF Spec document* dialog appears, as shown in [Figure 3-2](#) on page 82.

**Figure 3-2 Load PDF Spec Document**



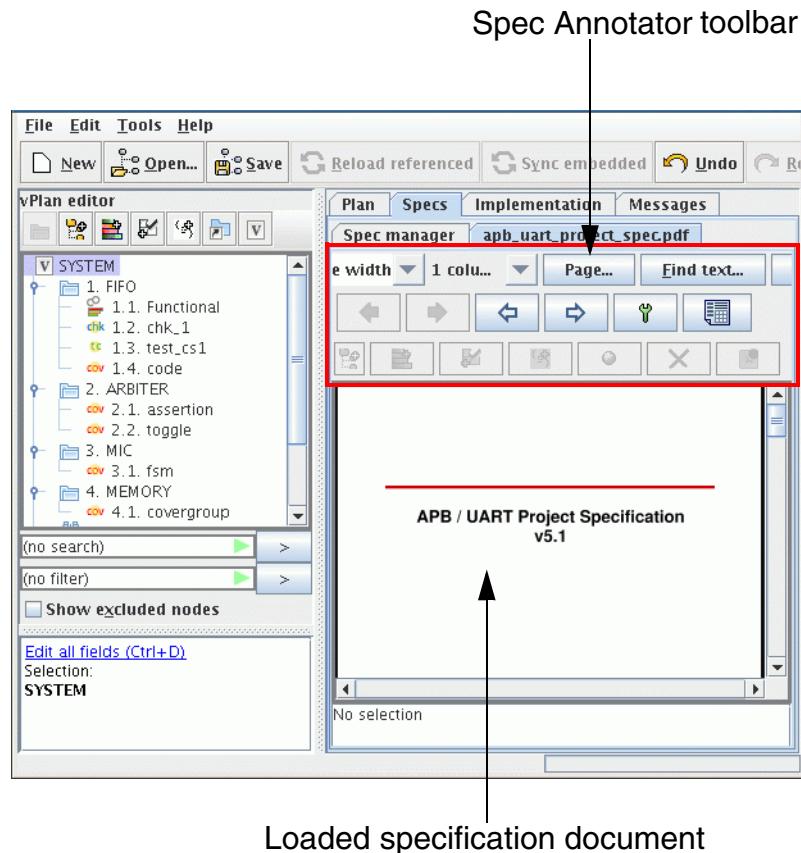
3. Locate and select the file that you want to associate with the plan.

**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.

4. Click *Open*.

Once a document is associated with the plan, it appears in the *Specs* tab, as shown in [Figure 3-3](#) on page 83.

**Figure 3-3 Loaded Specification Document**



You can associate any number of documents with a plan or planned element. In addition, you can link a single document to any number of plans or planned elements.



**Important**  
The specification document that you plan to associate must be in PDF format. If you have a document in another format such as Microsoft Word or Adobe FrameMaker, you can generate a PDF copy and annotate the copy.

### Using Password Protected PDFs

If the file that you planning to use is password-protected, then a message is shown to indicate that you first need to install a third party library.

To can get this library and to use it, perform the following steps:

1. Download `bcpprov-ext-jdk15*.jar` from the *Provider* column on this page:  
[http://www.bouncycastle.org/latest\\_releases.html](http://www.bouncycastle.org/latest_releases.html)

2. Set the environment variable VMANAGER\_EXTRA\_JAR\_PATH and point it to the location of the downloaded jar file.

```
setenv VMANAGER_EXTRA_JAR_PATH <location_of_bcprov-ext-jdk15*.jar>
```

**Note:** <location\_of\_bcprov-ext-jdk15\*.jar> should be the complete hierarchical path with file name.

3. Restart vPlanner.

```
vplanner
```

After performing the above steps (related to setting up of jar file), vPlanner will prompt for the password when opening a password-protected PDF file.

### 3.1.1 Manipulating the Display of Document within Spec Annotator

By default, the loaded specification document is horizontally scaled in the Spec Annotator area. Based on your requirements, you can:

- Scale the Document
- Set the Number of Columns to Display

#### 3.1.1.1 Scale the Document

To scale the document vertically, horizontally, or based on a specific percentage, click the *Fit page width* drop-down, as shown in [Figure 3-4](#) on page 84.

**Figure 3-4 Scale the Document**

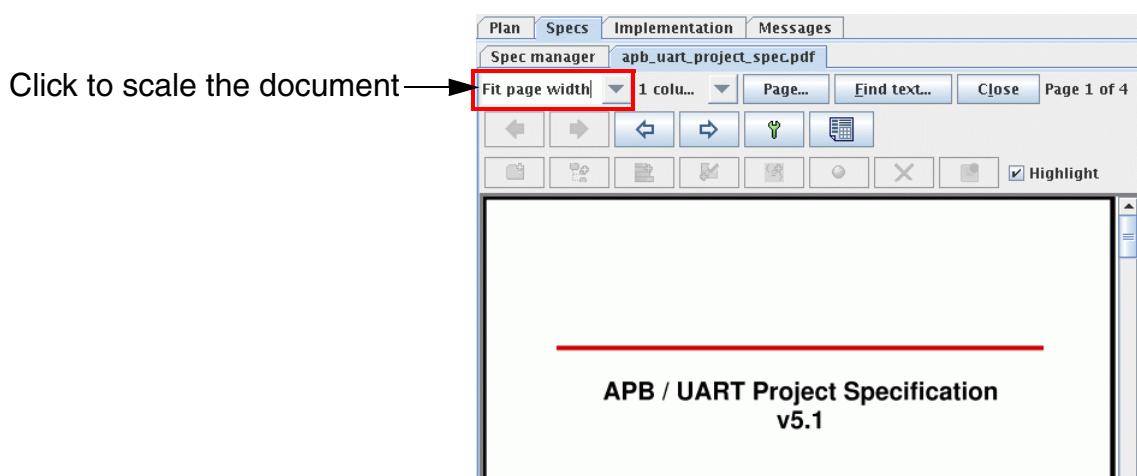
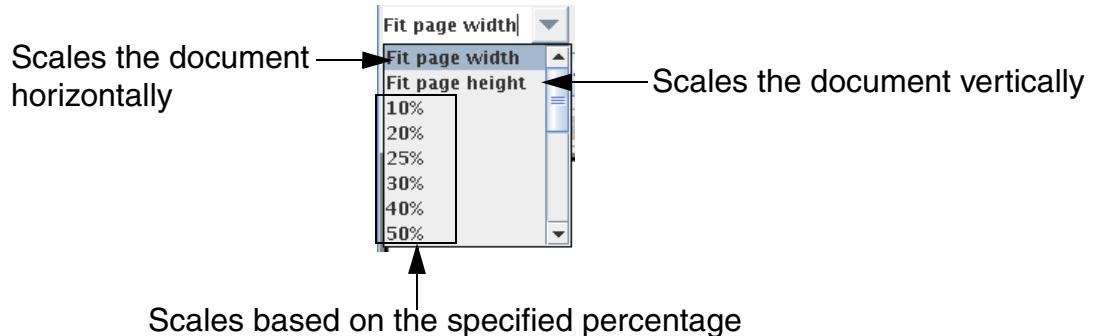


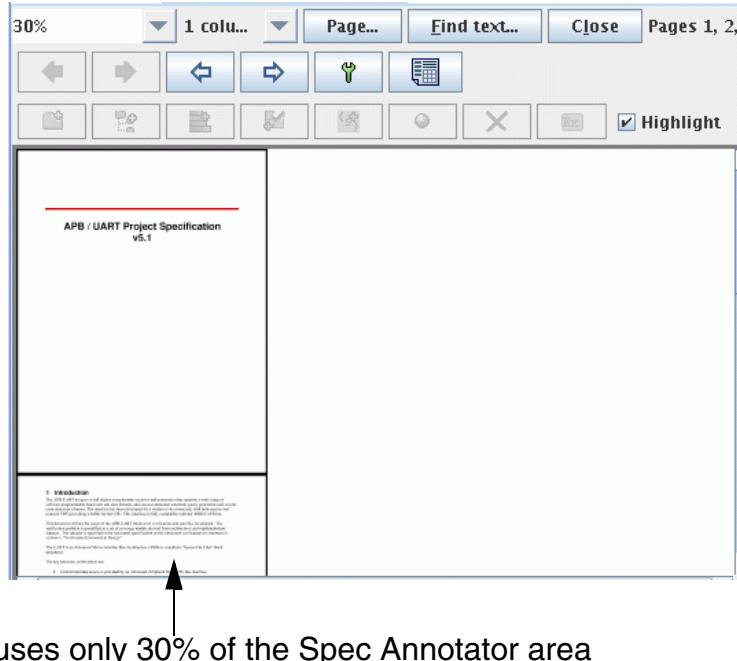
Figure 3-5 on page 85 shows the options available in the drop-down list.

**Figure 3-5 Scale the Document (drop-down list)**



You can make a selection based on your requirement. For example, select 30% from the drop-down list. The document now uses only 30% of the Spec Annotator area, as shown in Figure 3-6 on page 85.

**Figure 3-6 Document Scaled to 30%**



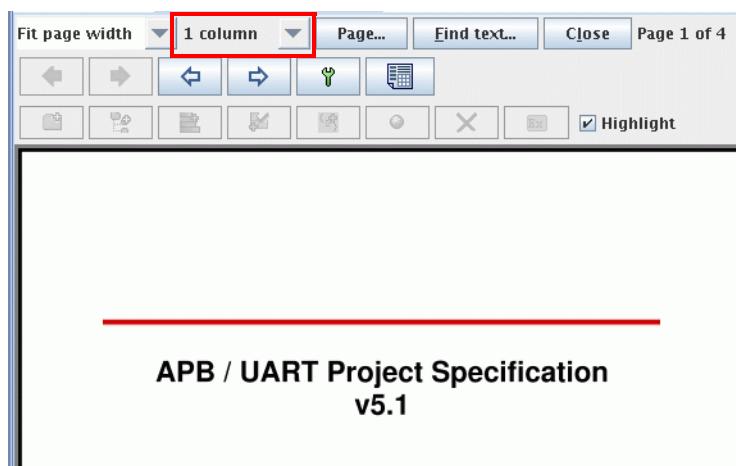
The document uses the Spec Annotator area based on your selection.

### 3.1.1.2 Set the Number of Columns to Display

By default, the loaded specification document is shown as a single column. You can modify the column settings to show multiple pages side-by-side in the Spec Annotator area.

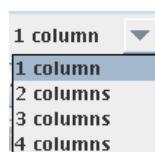
To show multiple pages side-by-side, click the *1 column* drop-down, as shown in [Figure 3-7](#) on page 86.

**Figure 3-7 Set Number of Columns**



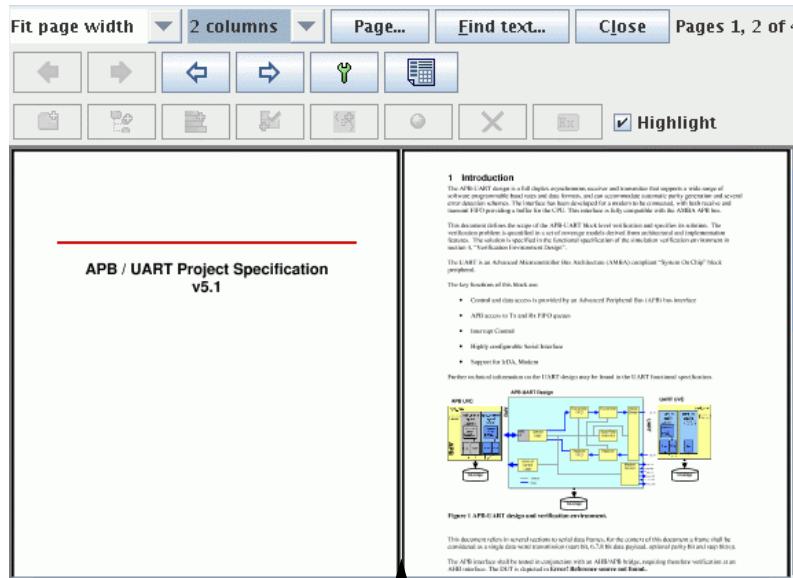
[Figure 3-8](#) on page 86 shows the options available in the drop-down list.

**Figure 3-8 Number of Columns (drop-down list)**



You can make a selection based on your requirement. For example, select *2 columns* from the drop-down list. The document now shows as two-columns in the Spec Annotator area, as shown in [Figure 3-9](#) on page 87.

**Figure 3-9 Document Shows as 2-columns**



Document shows as 2-column

Showing multiple pages side-by-side helps you quickly navigate through the entire document.

## 3.2 Creating and Modifying Annotations

After associating a specification document with the plan, you can create annotations and modify them, as required. This section covers the following topics:

- [Creating a New vPlan Element Using Selected Text](#)
- [Annotating an Existing vPlan Element](#)
- [Locating Text in the PDF Specification Document](#)
- [Displaying and Modifying Annotations for a vPlan Element](#)
- [Finding the vPlan Elements of an Annotation](#)
- [Navigating Between Annotations](#)
- [Displaying Annotation Statistics](#)

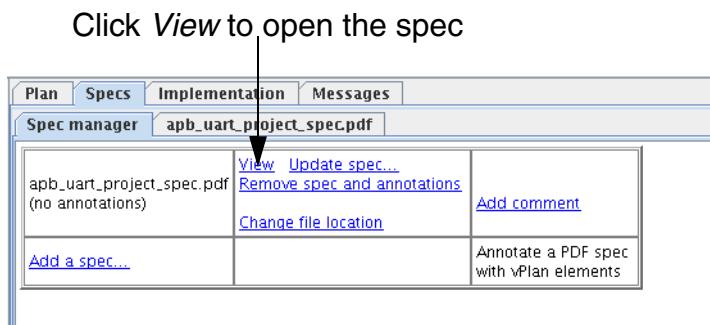
### 3.2.1 Creating a New vPlan Element Using Selected Text

To create a new vPlan element using the selected text from a spec, perform the following steps:

1. In the *Specs—Spec manager* tab, select the document whose text you want to use as annotations. Each document associated with the plan has its own tab.

Figure 3-10 on page 88 shows the list of loaded specs.

**Figure 3-10 Specs—Spec manager Tab**



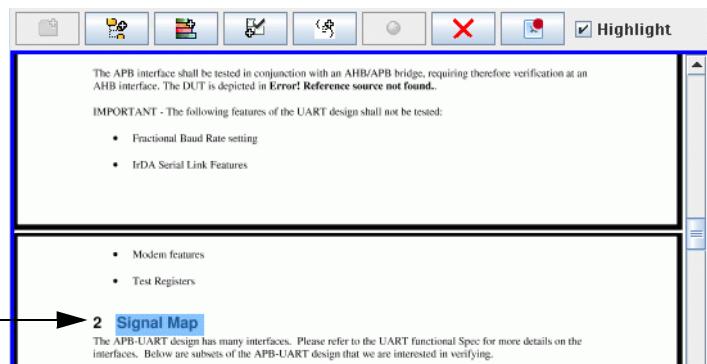
For example, to add annotations from the `apb_uart_project_spec.pdf` file, click *View* in the column next the specification file.

**Note:** Alternatively, you can click the `apb_uart_project_spec.pdf` tab page (as this file is already open).

2. The Spec Annotator area shows the selected spec. Browse through the document and locate the text that you want to use as an annotation.
3. Using the mouse, select the text that you want to use. For example, select the text *Signal Map*.

Figure 3-11 on page 89 shows the text *Signal Map* selected.

**Figure 3-11 Select the Text for Annotation**



4. Click any of the following buttons in the toolbar to create a new vPlan element, as required.

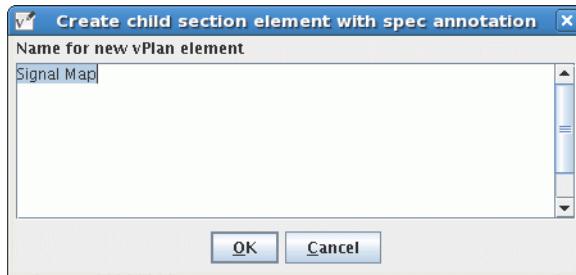
	Create a new vPlan section and map it to a new annotation created for the selected text. Add the section as a sibling of the current selection in the vPlan.
	Create a new vPlan section and map it to a new annotation created for the selected text. Add the section as a child of the current selection in the vPlan.
	Create a new planned coverage element and map it to a new annotation created for the selected text.
	Create a new planned check element and map it to a new annotation created for the selected text.
	Create a new planned testcase element and map it to a new annotation created for the selected text.

**Note:** Alternatively, you can right-click and select the required option from the pop-up menu.

For example, to add the selected text as a new child section of MEMORY, right-click and select *New section (child)*. Ensure that MEMORY is selected in the vPlan editor.

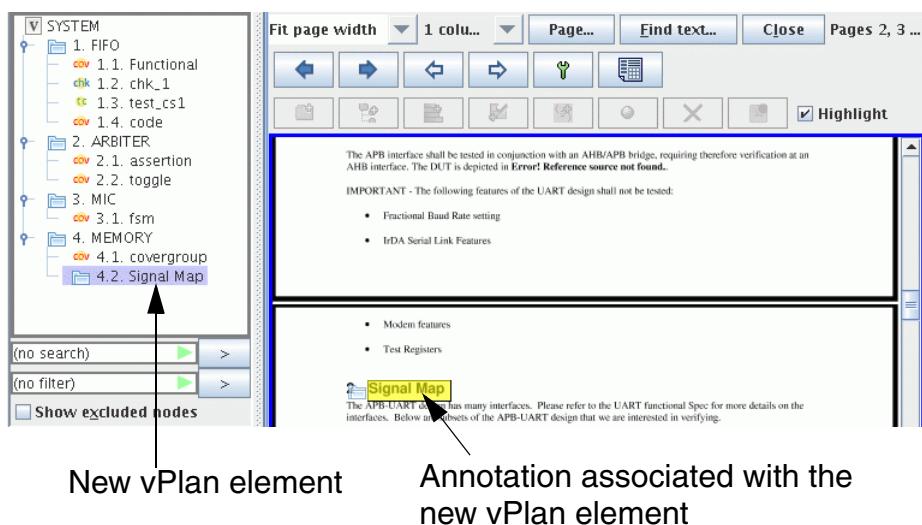
5. The *Create child section element with spec annotation* dialog box appears, as shown in [Figure 3-12](#) on page 90. By default, the selected text appears in the text box. You can choose to keep the same name of the section, or specify a different name in the text box. To keep the same name, click *OK*.

**Figure 3-12 Specify the Name of the vPlan Element**



A new child section is added to vPlan element *MEMORY* with Annotated text *Signal Map*, as shown in [Figure 3-13](#) on page 90.

**Figure 3-13 New vPlan Element and Annotation**



In the above figure, notice that:

- The annotated text is highlighted in yellow.

**Note:** vPlanner uses different colors to show annotations. This helps you easily distinguish between different types of vPlan elements. The following highlighting scheme is used by vPlanner:

- ❑ Sections are highlighted in yellow.
- ❑ Coverage is highlighted in green.
- ❑ Checks are highlighted in red.

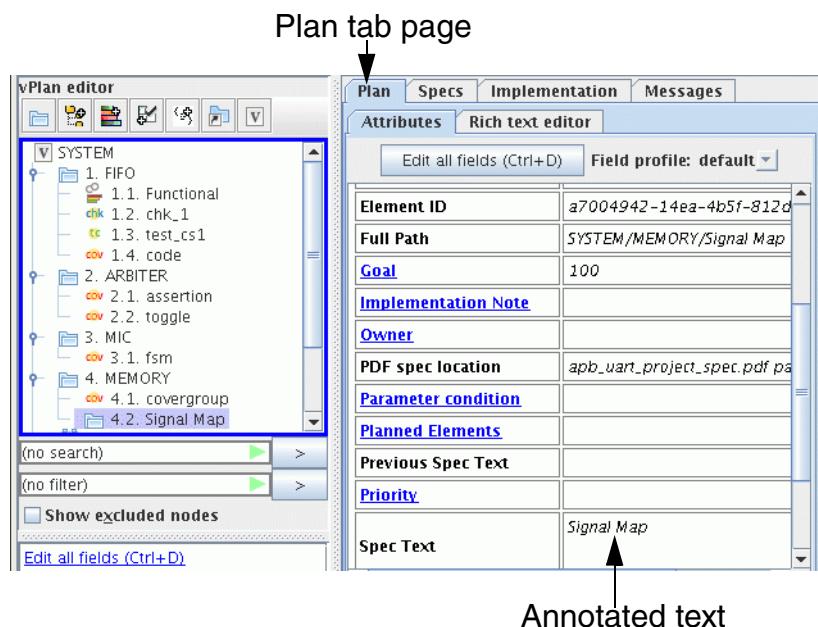
- Testcases are highlighted in blue.

In case you do not want the annotation to be highlighted, deselect the *Highlight* check box in the Spec Annotator toolbar.

- A small icon appears at the beginning of the annotated text.

When you add an annotation, the *Spec Text* attribute in the *Plan* tab is populated with the annotated text, as shown in [Figure 3-14](#) on page 91.

**Figure 3-14 Annotated Text**



The *Spec Text* row shows the annotated text.

### 3.2.2 Annotating an Existing vPlan Element

To annotate an existing vPlan element with text from a spec:

1. In the vPlan editor, select the element you want to annotate.

**Note:** You must select a single editable section or a single coverage/check/testcase element in the vPlan.

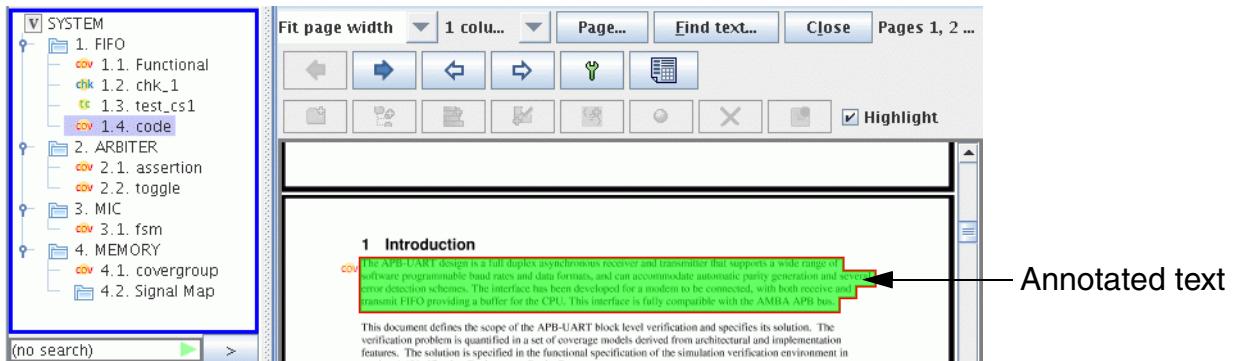
2. In the Spec Annotator area, select the text that you want to add as an annotation.

To quickly locate a specific page or specific text, use the Go to page or Search capability. See [Locating Text in the PDF Specification Document](#) for more information.

3. Right-click and select *Map to existing vPlan element*.

The annotation is highlighted, as shown in [Figure 3-15](#) on page 92.

**Figure 3-15 Annotated Text**



The annotation also shows in the *Plan* tab in the *Spec Text* row of the attributes table, as shown in [Figure 3-16](#) on page 92.

**Figure 3-16 Annotated Text in Attributes Table**

Plan tab page

This screenshot shows the vPlan editor interface. On the left is a tree view of a project structure under 'SYSTEM'. The right side has tabs for 'Plan', 'Specs', 'Implementation', and 'Messages'. The 'Plan' tab is selected. Below the tabs is a table with rows for 'Attributes' and 'Rich text editor'. The 'Rich text editor' row contains fields for 'PDF spec location' (set to 'apb\_uart\_project\_spec.pdf page 1'), 'Parameter condition' (empty), 'Planned Elements' (empty), 'Previous Spec Text' (empty), and 'Priority' (empty). The 'Spec Text' field contains the same annotated text as in Figure 3-15. An arrow points from the text 'Annotated text' to the highlighted area in the 'Spec Text' field. The table also includes a 'Field profile: default' dropdown and a 'Edit all fields (Ctrl+D)' link.

**Note:** You can annotate a single vPlan element more than once. If an element is annotated more than once, the *Spec Text* row displays each annotated passage as a separate paragraph.

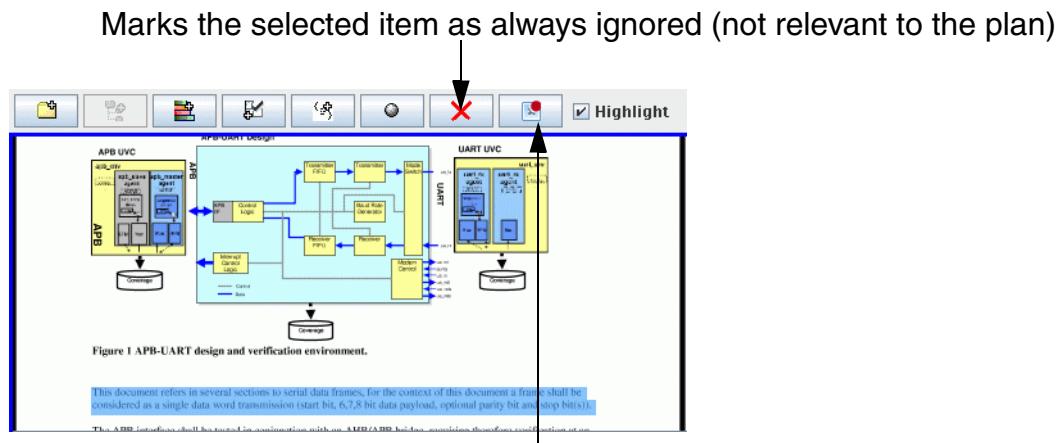
### 3.2.3 Marking Text as Always Ignored or Ignored Until Later Version

While annotating a specification document, you can mark certain text as:

- Always Ignored, which indicates that the selected text is irrelevant to the plan
- Ignored until later version, which indicates that the selected text is not relevant at this time but might be reviewed later

[Figure 3-19](#) on page 95 shows the relevant button in the Spec Annotator toolbar that can be used to mark the items as ignored or excluded.

**Figure 3-17** Marking Text Ignored or Excluded



To mark certain text as *always ignored* or *ignore until later version*:

1. Select the text you want to mark.
2. Click the required button, in the Spec Annotator toolbar.

This will mark the text, as required.

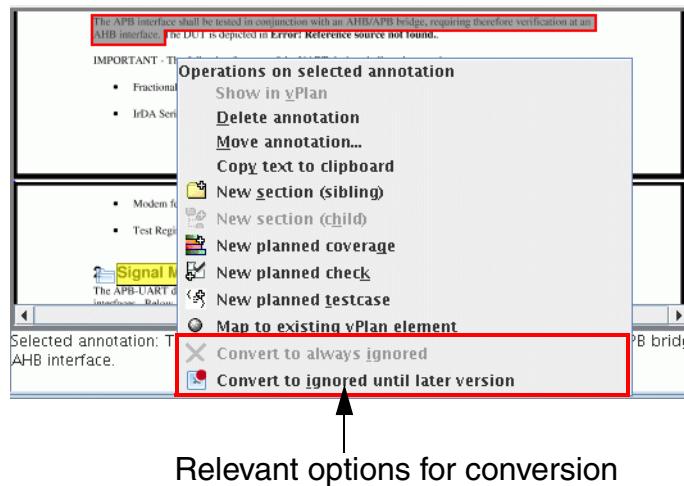
After you mark a text, if required, you can map this marked text to any of the vPlan items. For details on mapping text to a vPlan item, see [Creating a New vPlan Element Using Selected Text](#) on page 88 and [Annotating an Existing vPlan Element](#) on page 91.

You can also do following conversions:

- Convert a text marked *Always ignored* to *Ignored until later version*
- Convert a text marked *Ignored until later version* to *Always ignored*

To convert an already marked text, right-click the marked text, and select the relevant option from the pop-up menu, as shown in [Figure 3-19](#) on page 95:

**Figure 3-18 Conversion Options**



### 3.2.4 Locating Text in the PDF Specification Document

To locate text in a PDF document, you can:

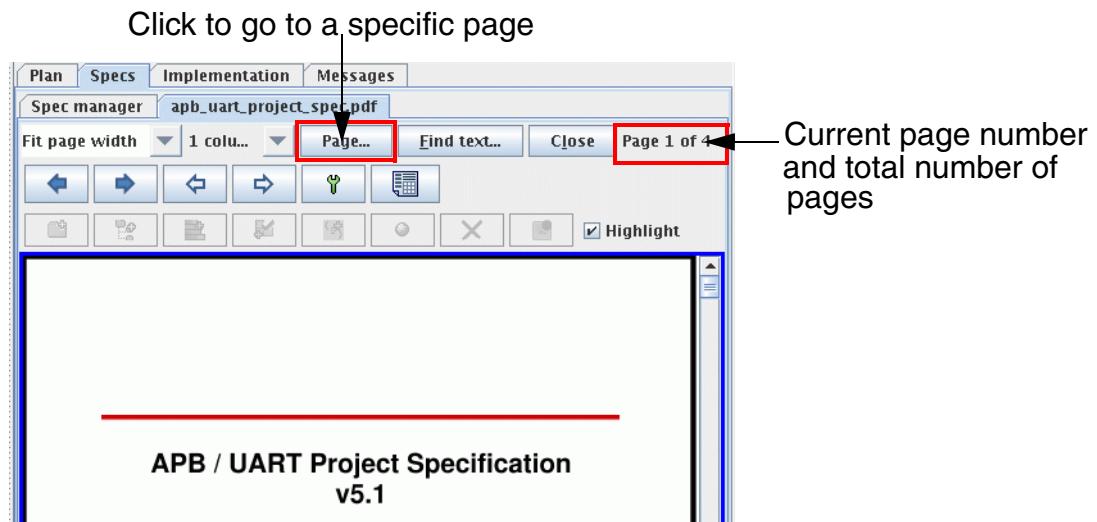
- [Go Directly to a Specific Page](#)
- [Search for a Specific Text](#)

#### 3.2.4.1 Go Directly to a Specific Page

To go to a specific page in the specification document:

1. Click the *Page* button, as shown in [Figure 3-19](#) on page 95:

**Figure 3-19 Go To a Specific Page**



- When you click the *Page* button, the *Go to page* dialog box appears, as shown in [Figure 3-20](#) on page 95. Specify the page number you want to go to, and click *OK*. For example, specify 3 and click *OK*.

**Figure 3-20 Specify Page Number**



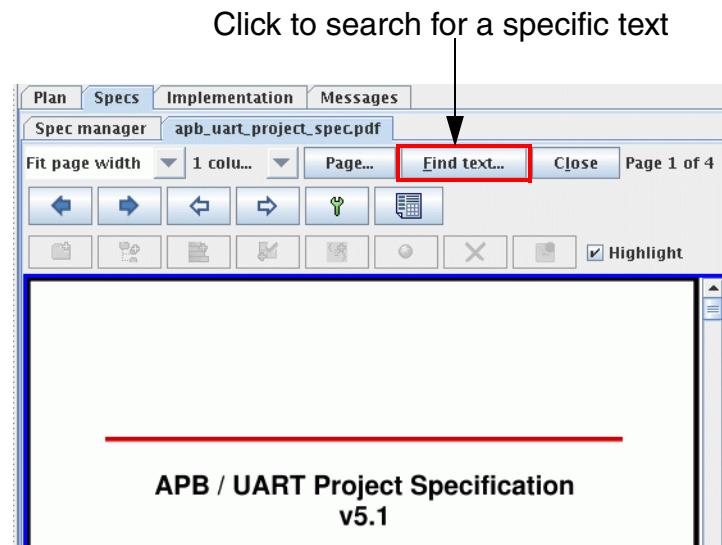
After you click *OK*, you reach the specified page immediately.

### 3.2.5 Search for a Specific Text

To search for a specific text in the specification document:

- Click the *Find text* button, as shown in [Figure 3-21](#) on page 96:

**Figure 3-21 Search for Specific Text**



2. When you click the *Find text* button, the *Search document* dialog box appears, as shown in [Figure 3-22](#) on page 96. Specify the text you want to search, and click *Run search*. For example, specify *configuration* as the search text, and click *Run search*.

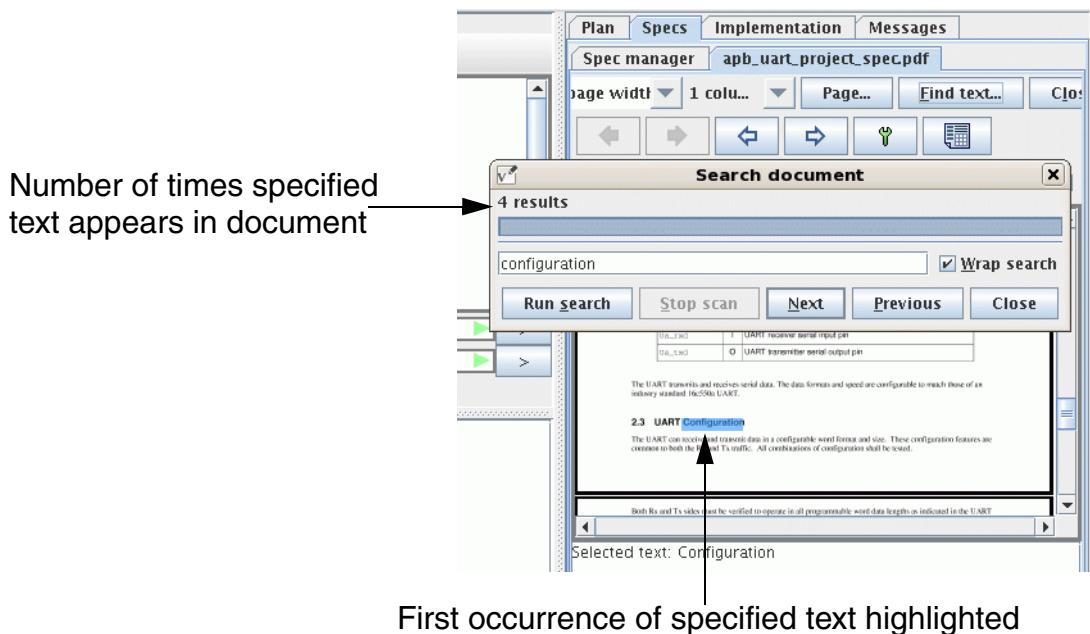
**Figure 3-22 Specify Search Text**



The box refreshes to show the number of hits. In addition, the first occurrence of specified text string is highlighted, as shown in [Figure 3-23](#) on page 97.

**Note:** The search is case insensitive and also ignores whitespace.

**Figure 3-23 Search Results**

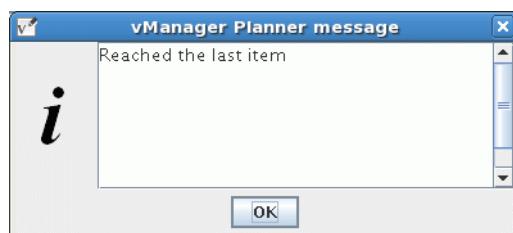


You can navigate through the search results by clicking *Next* or *Previous* buttons.

While navigating through the search result, after you reach the last occurrence of the matched text and click the *Next* button, either of the following happens:

- You are again taken to the first match. This happens if the *Wrap search* check box is selected.
- A message box, as shown in Figure 3-24 on page 97 is displayed to indicate that the search has reached the last item. This happens if *Wrap search* check box is not selected.

**Figure 3-24 Reached the Last Item**



### 3.2.6 Displaying and Modifying Annotations for a vPlan Element

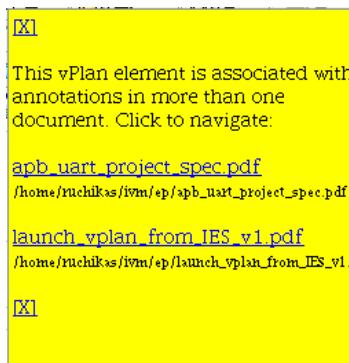
To display the annotations for a selected vPlan element:

1. Select the element in the vPlan editor.
2. Right-click and select *Show PDF annotation(s)* or click **Ctrl+P**.

If the selected vPlan element has a single annotation, the Spec Annotator area loads the corresponding PDF and highlights the annotated text.

If the selected vPlan element has more than one annotation, a yellow pop-up appears, as shown in [Figure 3-25](#) on page 98. It displays the list of annotations for the selected element.

**Figure 3-25 Multiple Annotations with Selected Element**

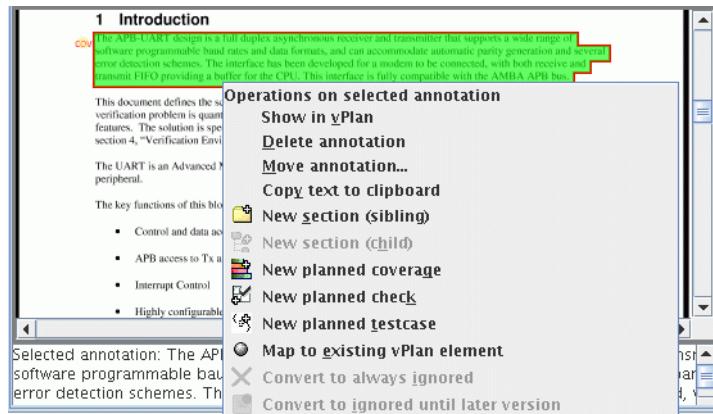


As you place the cursor over any of the links, the corresponding annotation is shown in the Spec Annotator area. When you click the hyperlink, the corresponding annotation is selected in the Spec Annotator area.

You can modify or delete the annotation by right-clicking on the annotation, and selecting the operation you want to perform.

[Figure 3-26](#) on page 99 shows the pop-up menu that appears when you right-click an annotation.

**Figure 3-26 Pop-up Menu**



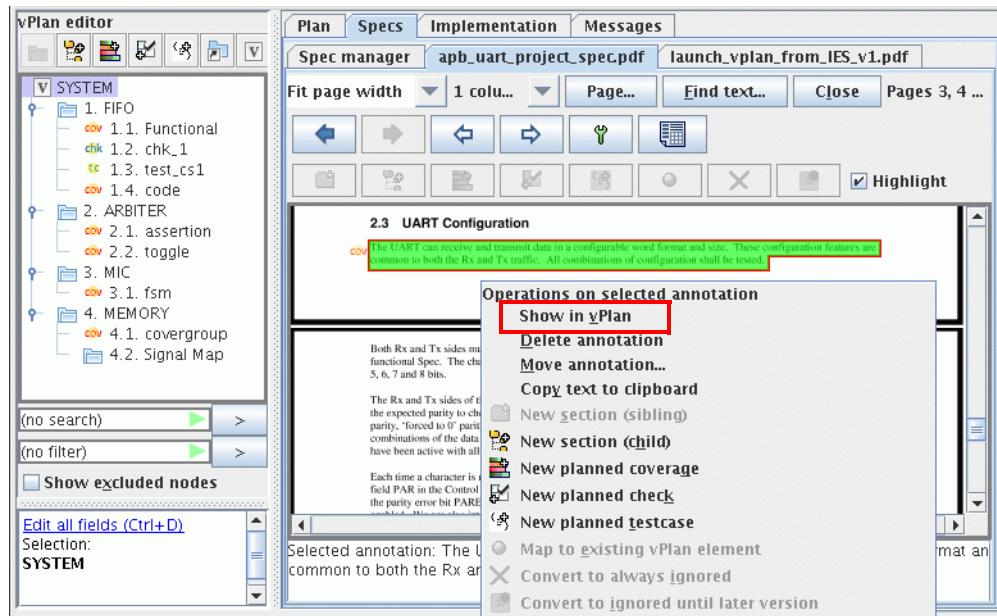
From the pop-up menu, you can select the operation that you want to perform.

### 3.2.7 Finding the vPlan Elements of an Annotation

To find the vPlan elements associated with an annotation:

1. Select the annotation.
2. Right-click the annotation and select *Show in vPlan*, as shown in [Figure 3-27](#) on page 100.

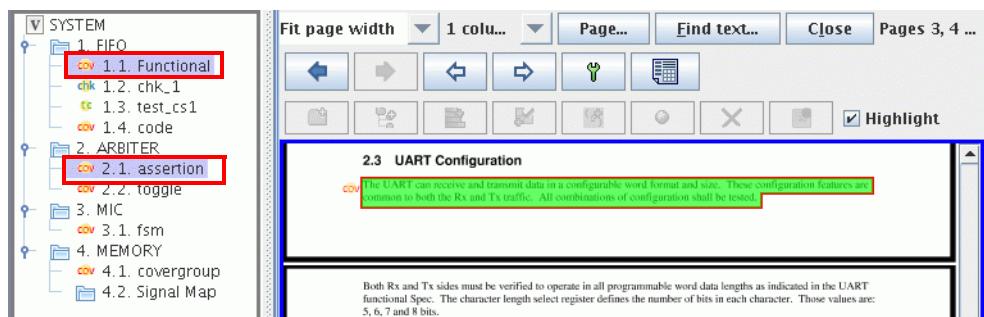
**Figure 3-27 Finding vPlan Elements of an Annotation**



This will select the corresponding vPlan element in the vPlan editor. If a particular annotation is associated with more than one vPlan elements, then all of the corresponding elements are selected simultaneously in the vPlan editor.

For example, in this case, the annotation was associated with two elements. After you select *Show in vPlan*, both the elements are selected in the vPlan editor, as shown in [Figure 3-28](#) on page 100.

**Figure 3-28 Finding vPlan Elements of an Annotation**

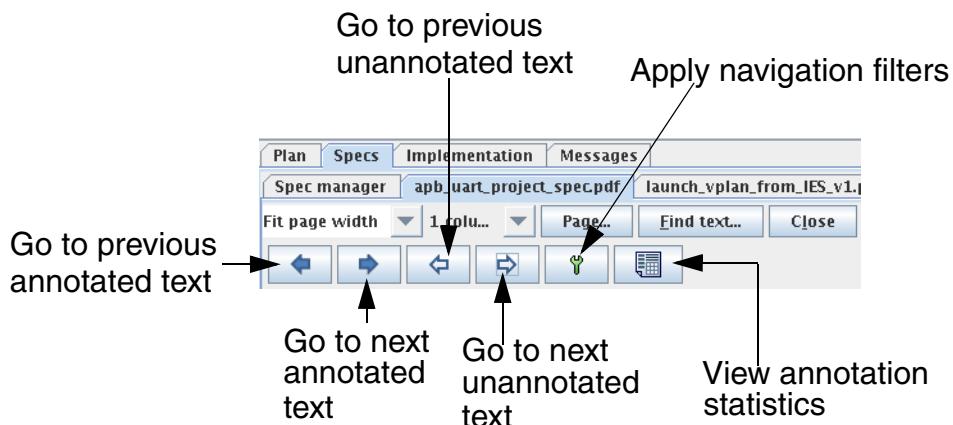


### 3.2.8 Navigating Between Annotations

After adding annotations, you can navigate between annotations and also apply filters to refine your navigation. You can also locate unannotated regions in the specification document.

[Figure 3-29 on page 101](#) shows the navigation toolbar.

**Figure 3-29 Navigation Toolbar**

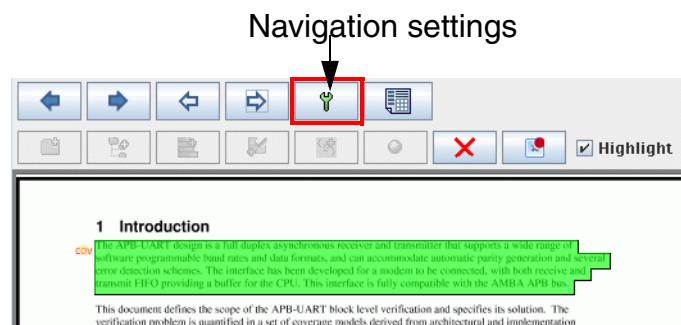


Using the navigation buttons, you can quickly navigate between annotations or unannotated regions.

### 3.2.9 Applying Navigation Filters

You can also apply filters to limit the navigation to only certain annotation types. To apply a navigation filter, click the *Navigation Settings* button, as shown in [Figure 3-30 on page 101](#).

**Figure 3-30 Apply Navigation Filter**



When you click the *Navigation Settings* button, the *Navigation settings* dialog box appears, as shown in [Figure 3-31](#) on page 102.

**Figure 3-31 Navigation Settings**



The *Navigation settings* dialog box shows the filters that can be applied. By default, all filters are selected, which indicates that when you click the navigation buttons, you are navigated to the next annotated text irrespective of the type of annotation.

Annotations can be of type:

- Section—To navigate between items mapped to vPlan sections.
- Coverage—To navigate between items mapped to vPlan coverage items.
- Checks—To navigate between items mapped to vPlan check items.
- Testcases—To navigate between items mapped to vPlan testcase items.
- Always ignored—To navigate between items marked always ignored.
- Ignored until later releases—To navigate between items marked ignored until later releases.
- Unmapped to vPlan — If a vPlan element was annotated and later that element was deleted from the vPlan, then the annotation corresponding to that vPlan item is highlighted in gray. The *Unmapped to vPlan* option is used to navigate between such items.
- Updated text—To navigate between annotations that include the text that has changed during the time of updating the specification document. For more details on updating the specification document, see [Updating the Specification Document](#) on page 105.

The type is decided at the time of applying annotation. In case you want to enable navigation only between a specific type of annotation, select the check box corresponding to that type and deselect all others.

For example, to enable navigation between annotations of type section:

1. Select only the *Sections* check box and deselect other check boxes, as shown in [Figure 3-32](#) on page 103.

**Figure 3-32 Navigation Settings**

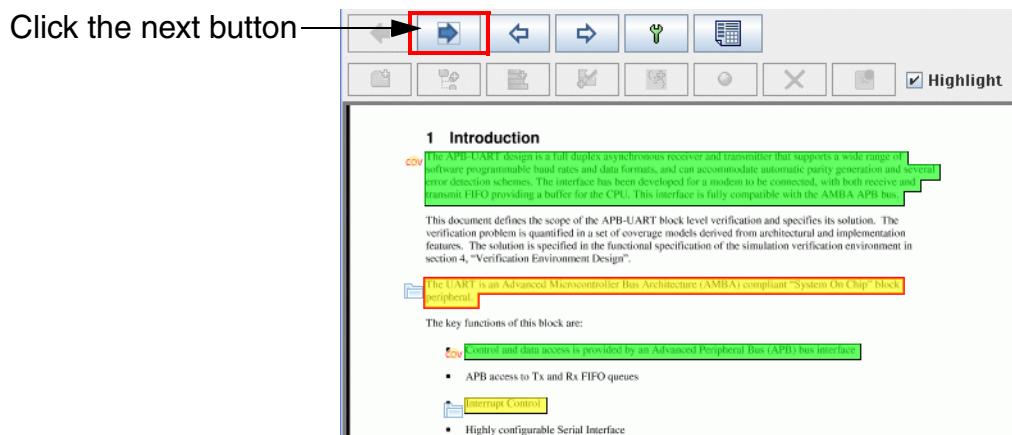


2. Click *Close* after applying the filter settings.

After applying the filter settings, when you click the navigation buttons, you are directly navigated to the annotation specified in the Navigation settings dialog box.

3. Click the *Next* navigation button in the toolbar, as shown in [Figure 3-33](#) on page 103.

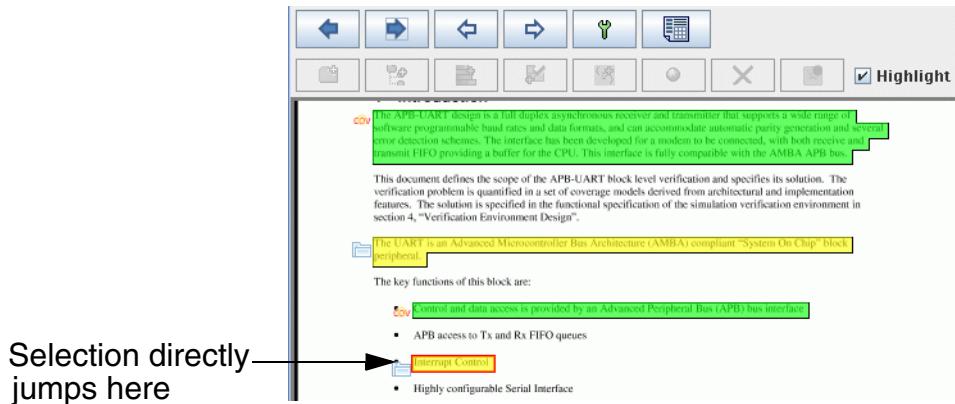
**Figure 3-33 Navigation Settings Applied**



The first instance of annotation of type section is selected. The red outline around the annotation indicates selection.

- Again click the *Next* button. You are now directly navigated to the next instance of navigation of type section, as shown in [Figure 3-34](#) on page 104.

**Figure 3-34 Navigation Settings Applied**

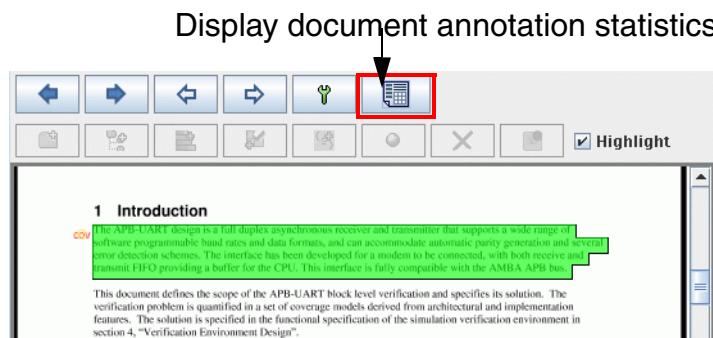


As the navigation filter is applied, the *Next* button navigates to the next available annotation of type *Section*. Notice that the annotation highlighted in green in the above figure is skipped as it is not of type *Section*.

### 3.2.10 Displaying Annotation Statistics

To display the annotation statistics of the current vPlan, click the *Display document annotation statistics* button, as shown in [Figure 3-35](#) on page 104.

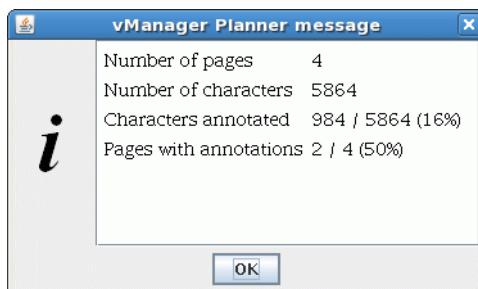
**Figure 3-35 Display Document Annotation Statistics**



When you click the *Display document annotation statistics* button, the annotation statistics is calculated and shown in a message box, as shown in [Figure 3-36](#) on page 105.

**Note:** Statistics calculation might be time-consuming in the case of large specification documents.

**Figure 3-36 Annotation Statistics**



The following information is displayed in the message box:

- Number of pages: Shows the total number of pages in the document.
- Number of characters: Shows the total number of characters in the document.
- Characters annotated: Shows the total number of characters annotated and the total number of characters in the document. It also shows the percentage of characters annotated in the document.
- Pages with annotations: Shows the number of pages that contain at least one annotation and the total number of pages in the document. It also displays the percentage of pages containing at least one annotation and percentage of pages annotated.

The annotation statistics information is helpful when you have a goal for annotating the document. By looking at the statistics, you can measure your target and make further changes.

### 3.3 Updating the Specification Document

If there are changes in the specification document after the verification plan is created and annotated, you can analyze the changes using vPlanner and update the vPlan accordingly.

vPlanner automatically identifies the changes made to the specification document. You can review the changes and decide whether it is necessary to update the verification plan with the new specification document or not.

**Note:** vPlanner does not identify changes in the specs outside of the text that has been selected as an annotation.

Updating the specification document involves following two steps:

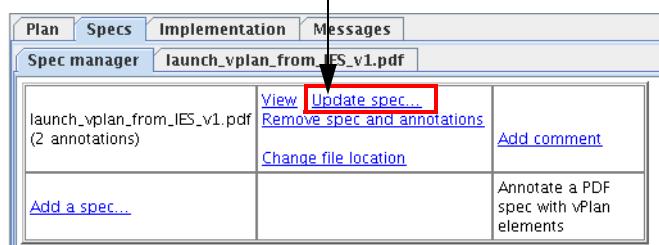
- Detecting Changes in the New Specification Document
- Migrating Annotations to the New Specification Document

To update specification document and to migrate annotations from an earlier version of the specification document to the updated specification document:

1. In the *Specs—Spec manager* tab, click the *Update Spec* link, as shown in [Figure 3-37](#) on page 106.

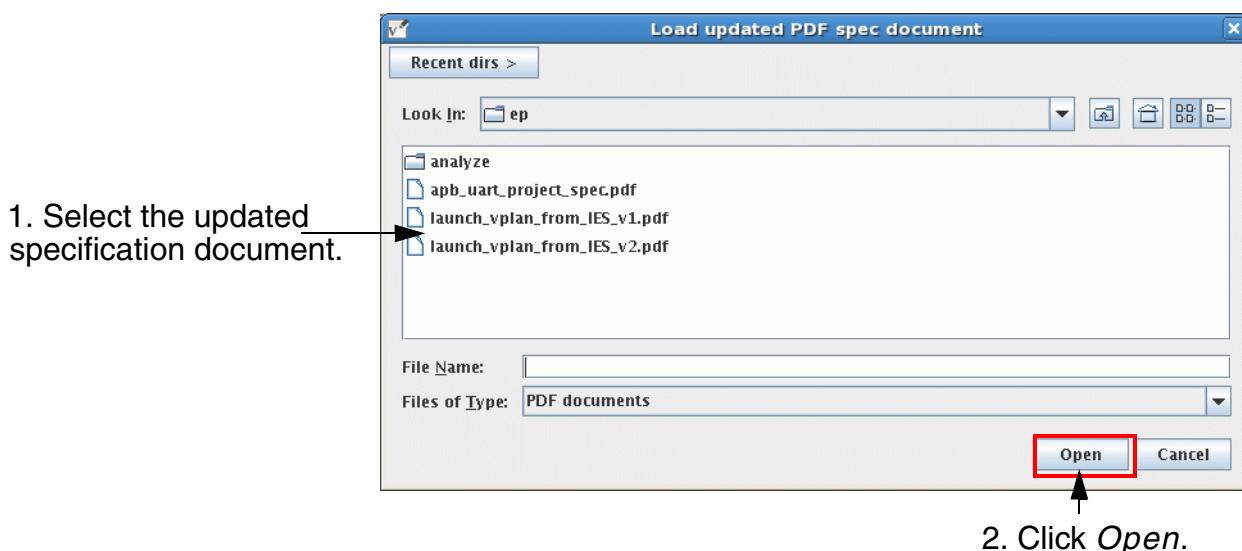
**Figure 3-37 Specs—Spec manager Tab**

Click *Update Spec* to update the specification document



2. The *Load updated PDF spec document* dialog box is displayed, as shown in [Figure 3-38](#) on page 106. Select the updated specification document and click *Open*.

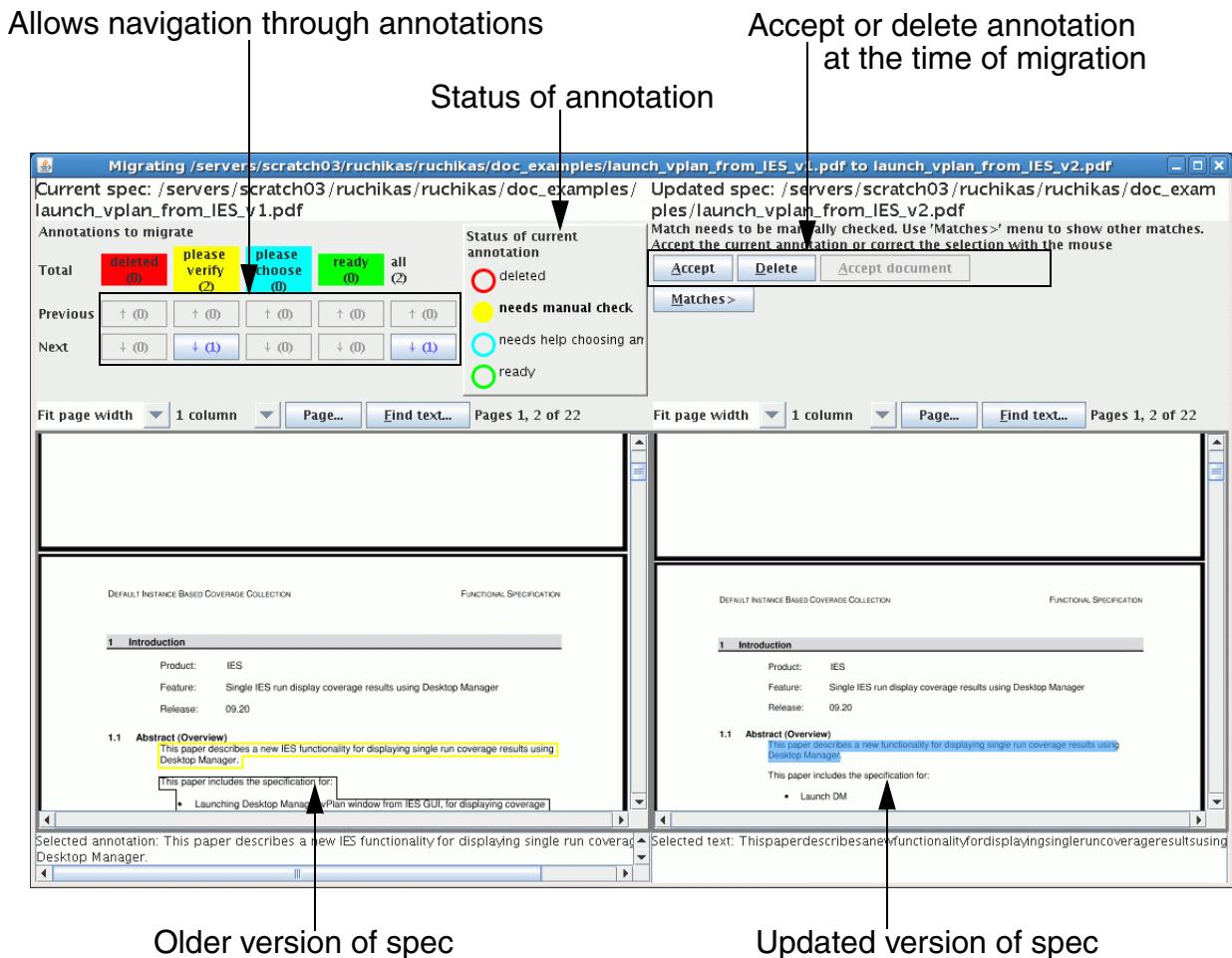
**Figure 3-38 Load Updated PDF Spec Document**



The *Migrating Annotations* window opens, as shown in Figure 3-39 on page 107.

**Note:** Analyzing the spec for changes might be time-consuming for large documents.

**Figure 3-39 Migrating Annotations**



The left pane shows the current (older) version of the spec. The right pane shows the updated (new) version of the spec.

In the left pane, the current annotation is selected and the color of the selection box is based on the type of match. The selection box can be of any of the following colors:

- Red—indicates that the annotation has been manually marked as deleted
- Yellow—indicates that potential matching text was found, but human confirmation is required

## vPlanner User Guide

- Green—indicates that matching text was found with high confidence and manual inspection is not required

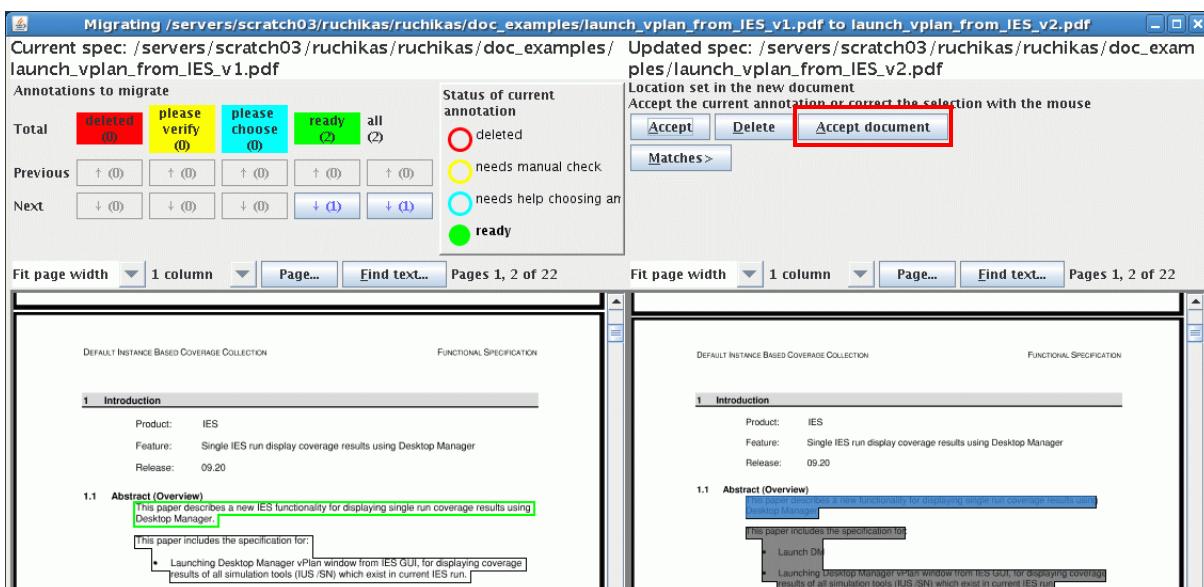
If a match corresponding to the selected annotation exists in the updated spec, it is highlighted blue in the right pane.

### 3. Do one of the following:

- Click *Accept* to confirm that the annotation should be migrated to the automatically highlighted location in the updated spec.
- Manually highlight a different area of the spec and click *Accept* to migrate the annotation to that location.
- Click *Matches* to show all potential matching text in the right pane that corresponds to the current annotation in the left pane.
- Click *Delete* if you determine that currently selected annotation no longer exists in the updated spec.

### 4. Navigate through all the annotations and Accept or Delete, as required. After reviewing all the annotations, click *Accept Document*, as shown in [Figure 3-40](#) on page 108 to transfer all annotations from the older specification document to the new updated specification document.

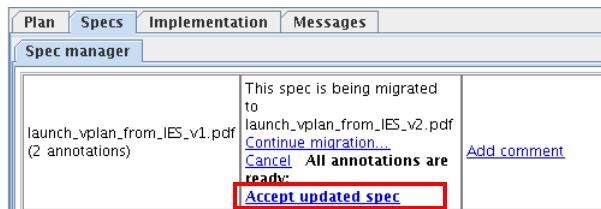
**Figure 3-40 Migrate Annotations**



**Note:** Alternatively, instead of clicking *Accept Document* in the Migration window, you can close the spec migration window and click the *Accept updated spec* link in the

*Specs—Spec manager tab page, as shown in [Figure 3-41](#) on page 109.*

**Figure 3-41 Accept Updates**



**Note:** If at any point you want to cancel the migration, you can close the migration window and click *Cancel* in the Spec manager tab page.

5. The *Accepting spec migration* confirmation box appears, as shown in [Figure 3-42](#) on page 109. Click *Yes* to confirm transfer of annotations.

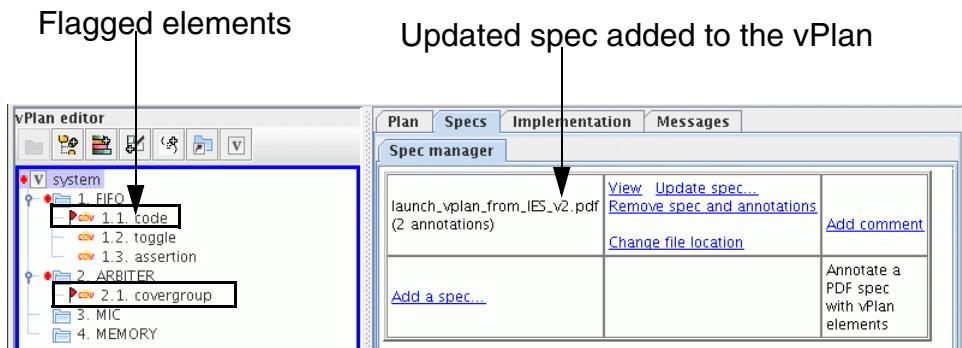
**Figure 3-42 Confirm Migration**



After you confirm, the updated specification document is added to the verification plan, and annotations are transferred to the updated specification document. In addition, the older version of the specification document is removed from the verification plan.

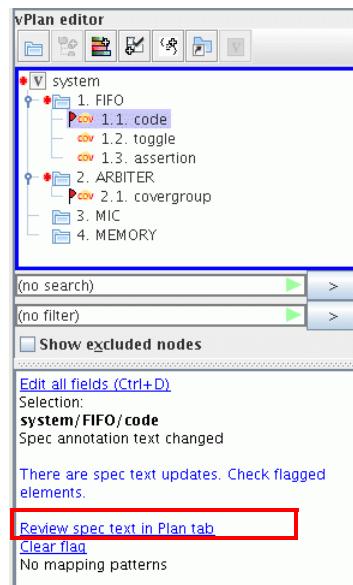
6. The vPlan elements that are impacted by the spec update are shown flagged in the vPlan editor, as shown in [Figure 3-43](#) on page 110. Select the vPlan element that is affected by the spec update.

**Figure 3-43 Spec Updated**



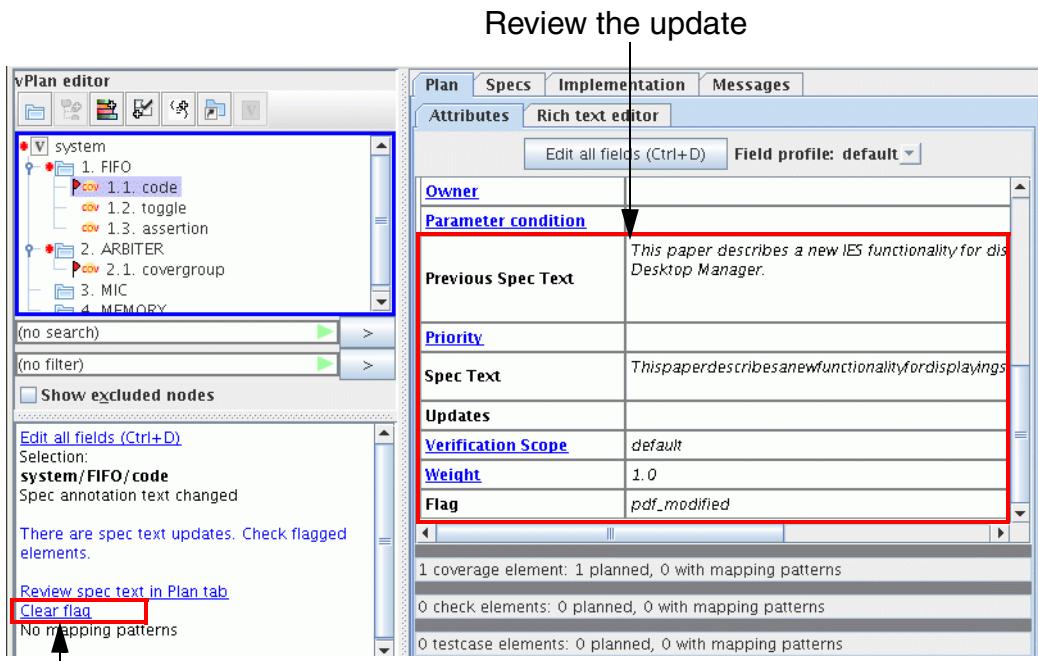
7. Select a flagged item to review the impact on it during migration.
8. Select *Review spec text in Plan tab* link, as shown in [Figure 3-44](#) on page 110.

**Figure 3-44 Review Updates**



This opens the Plan tab page in the right pane, as shown in [Figure 3-45](#) on page 111.

**Figure 3-45 Clear Flags**



Click to clear the flag corresponding to the vPlan element

You can review the changes in specification document using *Spec Text* and *Previous PDF spec text* fields in the *Attributes* table.

9. Click the *Clear Flag* link to remove the flag from the selected vPlan element.

Similarly, you can navigate through other flagged items and clear the flags, as required.

## 3.4 Rich-text Editing

In addition to associating specification documents with vPlan elements, you can also add rich-text data with each vPlan element within vPlanner. When you save the verification plan, this rich-text data is saved as a part of verification plan.

vPlanner uses OpenOffice 3.2 or higher to allow adding rich-text data to vPlan elements.

**Note:** By default, OpenOffice is not installed with INCISIVE. You can download it from betascape (<ftp://ftp.betascape.cadence.com>).

### 3.4.1 Enabling OpenOffice in vPlanner

To set up OpenOffice for vPlanner, perform the following steps:

1. Connect to the betascape ftp:
- 2.

```
ftp -n betascape.cadence.com
```

3. When prompted for username, specify vmfiles\_download.
4. Specify the password as VMEV4rzz.
5. Execute the following commands:

```
cd openoffice_for_ep  
bin  
prompt  
mget ep_oo03.20-p001lnx86.t.Z  
bye
```

6. Extract kit in base directory of the installation (<INSTALL\_DIR>/) using:

```
gtar -xzvf ep_oo03.20-p001lnx86.t.Z
```

**Note:** To extract the kit to the install location, you must have the write access to the install area. In case you do not have write access to the install area, you can extract the kit to some other stable location. You must then set the OO\_DIR environment variable to the absolute path of the location where the kit is extracted.

Once you extract OpenOffice kit at the specified location, OpenOffice is integrated into vPlanner and is available for adding text, as required.

Figure 3-46 on page 113 shows the *Rich text editor* tab page after OpenOffice is configured.

**Figure 3-46 Rich Text Editor Tab Page (After Configuring OpenOffice)**



You can begin editing in by clicking *Edit now*.

### 3.4.1.1 Using Your Own Version of OpenOffice

In case you do not want to use the Cadence OpenOffice kit, you can provide your own version of OpenOffice. For this:

1. Install OO 3.2 or higher.
2. Set `OO_DIR` environment variable to the root directory of OO installation.

Under `$OO_DIR` vPlanner should be able to find several files. The following test should succeed.

```
find $OO_DIR | grep "program/swriter"  
find $OO_DIR -name officebean.jar  
find $OO_DIR -name unoloader.jar
```

**Note:** It is important that all of the above files are related to the same OO installation, which should be of version 3.2 or higher.

### 3.4.2 Using the Rich-text Editor

After configuring OpenOffice, you can add rich-text to different vPlan elements as required.

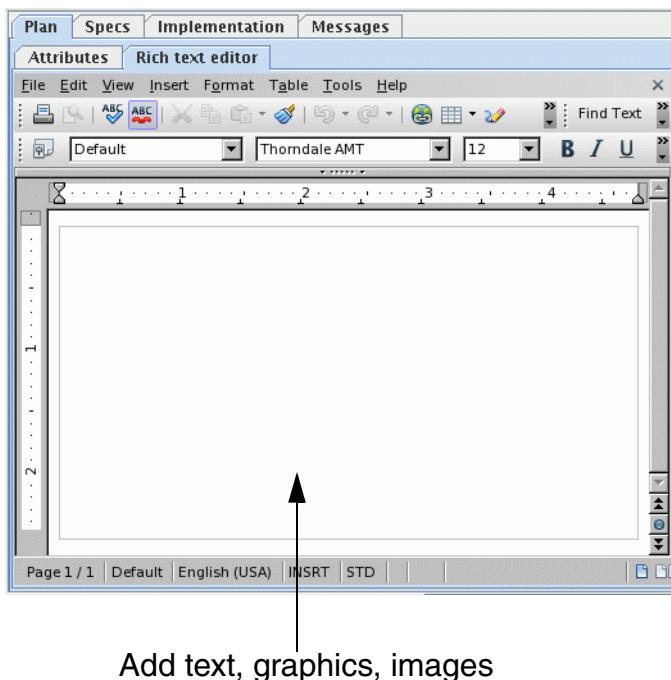
To add rich-text to a vPlan element:

1. Select the vPlan element to which you want to add text.
2. From the *Plan* tab, select *Rich text editor* tab page.

**3. Click *Edit now*.**

OpenOffice Writer open in the *Rich text editor* tab page, as shown in [Figure 3-47](#) on page 114.

**Figure 3-47 Rich Text Editor Tab Page (OpenOffice Writer)**



**4. You can add text, create tables, paste formatted text from other applications, import pictures, draw figures, and change fonts, as required.**

Rich text added to a vPlan element is saved in an encoded form within the vPlan file. This rich text is available when you reopen the verification plan.

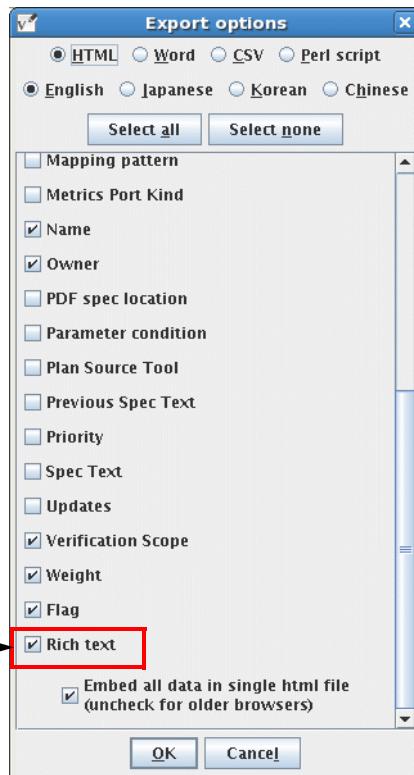
### **3.4.2.1 Exporting a Verification Plan that Includes Rich-text Content**

Exporting a verification plan is useful if you want to make the verification plan available for viewing to others who might not have vPlanner installed.

You can export a verification plan in HTML, CSV, or Microsoft Word format. For details on exporting a verification plan, see [Exporting and Publishing the Plan](#) on page 197.

If the verification plan includes rich-text content, you need to ensure that the *Rich text* check box is selected in *Export options* dialog box, as shown in [Figure 3-48](#) on page 115.

**Figure 3-48 Export Options**



**Note:** Below the *Rich text* check box, there is an additional check box, *Embed all data in single HTML file*. This check box is enabled only when you are exporting the plan in an HTML format.

- If the *Embed all data in single HTML file* check box is selected, only a single export file is created. This single file includes rich-text content (both text and images).
- If the *Embed all data in single HTML file* check box is not selected, an additional directory is created with the same name as the name of the export file. This additional directory includes subdirectories to save images used in the *Rich text editor* tab page.

It is important to deselect this option in the case of older browsers, such as Internet Explorer 7 or below, and Netscape Navigator 4.7 or below. This is because in older browsers, image data could not be embedded in the HTML file (using the "data:" URL scheme).

If you deselect "*Embed all data in single HTML file*", then when you share the export file with the user for viewing, ensure that you also share the additional folders created at the time of export.

## vPlanner User Guide

---

**Note:** The *Export options* dialog box is displayed when you select *Export* from the *File* menu. For more details on exporting a plan, see [Exporting and Publishing the Plan](#) on page 197.

---

## Reusing Verification Plans

---

There are several common scenarios where it is useful to reuse existing verification plans. For example:

- Your verification environment includes a verification IP component, and that IP component has an associated plan.
- Your verification environment is complex or the project team is geographically dispersed, making it appropriate for ownership of various components of the environment (and their associated plans) to be divided among the team members.
- You are reusing a legacy verification environment and that environment has an associated verification plan.

Using vPlanner, you can create a top-level plan and then either embed plans into it or create references to plans based on your requirements. This chapter covers the following topics:

- [Difference Between Embedding and Referencing](#)
- [Embedding a Plan in Another Plan](#)
- [Creating a Reference to Another Plan](#)
- [Making Verification Plans Reusable](#)

### 4.1 Difference Between Embedding and Referencing

[Table 4-1](#) on page 118 describes the difference between embedding a plan and referencing a plan.

**Table 4-1**

Embedding a Plan	Referencing a Plan
Embedding one plan into another plan copies all of the data from the embedded plan into the top-level plan. After embedding, the embedded plan is available as a section in the top-level plan and you can make changes to it, as required.	Referencing (Instantiating) a plan into another plan creates a “live link” from the top-level plan to the referenced plan. After referencing, the referenced plan is available for viewing only. You cannot make changes to the referenced plan from the top-level plan.
Embedding is usually appropriate when you are reusing a legacy verification environment. In this case, legacy changes in the original plan will not impact the top-level plan.  Embedding is also useful if your plan structure is defined in an external program, such as a spreadsheet or requirements tracking tool.	Referencing is appropriate if the referenced plan is maintained by another group and should only be modified by that group. In this case, the owner of the referenced plan can continue to make changes independent of the top-level plan.

## 4.2 Embedding a Plan in Another Plan

Embedding one plan into another plan copies all of the data from the embedded plan into the top-level plan. After embedding, the embedded plan is available as a section in the top-level plan and you can make changes to it, as required. Embedding is usually appropriate when you are reusing a legacy verification environment.

**Note:** In case changes are made to the embedded plan (outside the top-level plan), then the changes are not automatically propagated to the top-level plan. You need to synchronize the plan to make the changes available in the top-level plan.

This section discusses:

- [Embedding a Verification Plan](#)
- [Synchronizing the Plan](#)

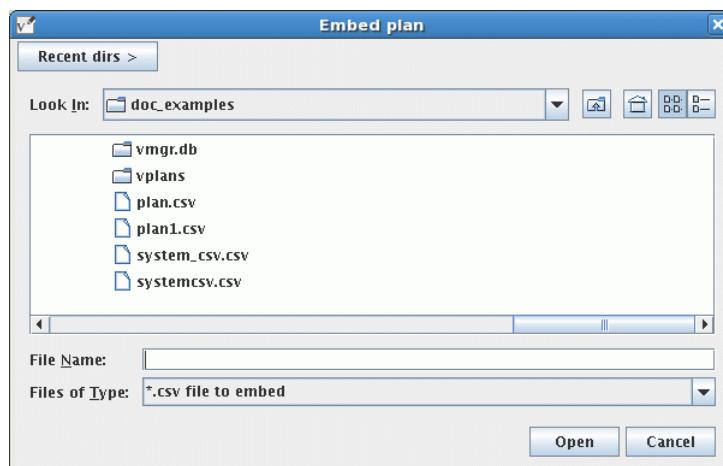
### 4.2.1 Embedding a Verification Plan

**Note:** You can embed only a CSV based plan within another verification plan.

To embed a plan within another plan:

1. Do any of the following:
  - ❑ Click the *Embed foreign vPlan* button above the vPlan tree, or
  - ❑ Right-click the vPlan and select *Embed foreign vPlan* from the pop-up menu.
2. The *Embed plan* dialog box appears, as shown in [Figure 4-1](#) on page 119. Select the plan that you want to embed and click *Open*. For example, to import `plan1.csv` in the existing plan, select `plan1.csv` from the list and click *Open*.

**Figure 4-1 Embed Plan**



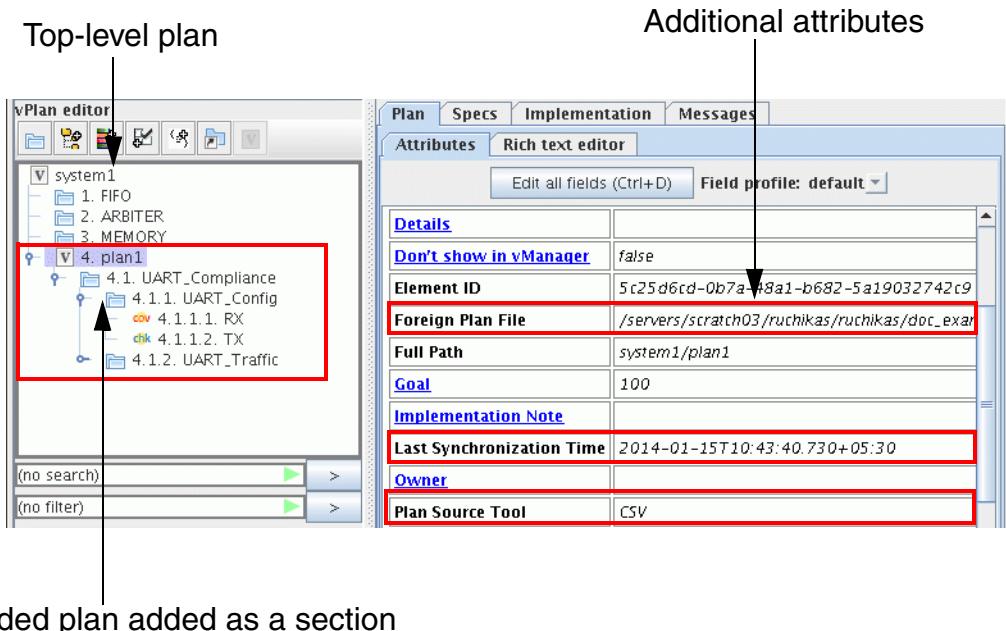
3. A dialog box appears, as shown in [Figure 4-2](#) on page 119. The dialog box, by default shows the name of the plan. If you want to specify a different name for the plan, specify it in the text box, and click *OK*.

**Figure 4-2 Embed Plan**



This will copy the contents of the `plan1.csv` to the already opened plan (top-level plan) `system1.vplanx`, as shown in [Figure 4-3](#) on page 120.

**Figure 4-3 vPlan Editor with the Embedded Plan**



The embedded plan is added as a section in the top-level plan. The *Attributes* tab page shows following attributes that are created when you embed a plan.

- Foreign Plan File—Shows the hierarchical path of the plan that is embedded.
- Last Synchronization Time—Shows the time when the embedded plan was last synchronized. For details on synchronization, see “[Synchronizing the Plan](#)” on page 120.
- Plan Source Tool —Shows the name of the tool in which the embedded plan was created.

After embedding, you can modify the plan as desired.

**Note:** You can embed any number of plans within a plan.

### 4.2.2 Synchronizing the Plan

After embedding a verification plan, if changes are made to the original plan (outside the top-level plan) and you want to make those changes available in the top-level plan, you must synchronize the plan. vPlanner allows you to:

- [Synchronize All the Embedded Plans Together](#)
- [Synchronize Specific Plans](#)

### 4.2.2.1 Synchronize All the Embedded Plans Together

To synchronize all of the embedded plans together, do any of the following:

- Right-click in the top-level plan in the vPlan editor and select *Synchronize all embedded plans*.
- Click the *Sync embedded* button on the toolbar.

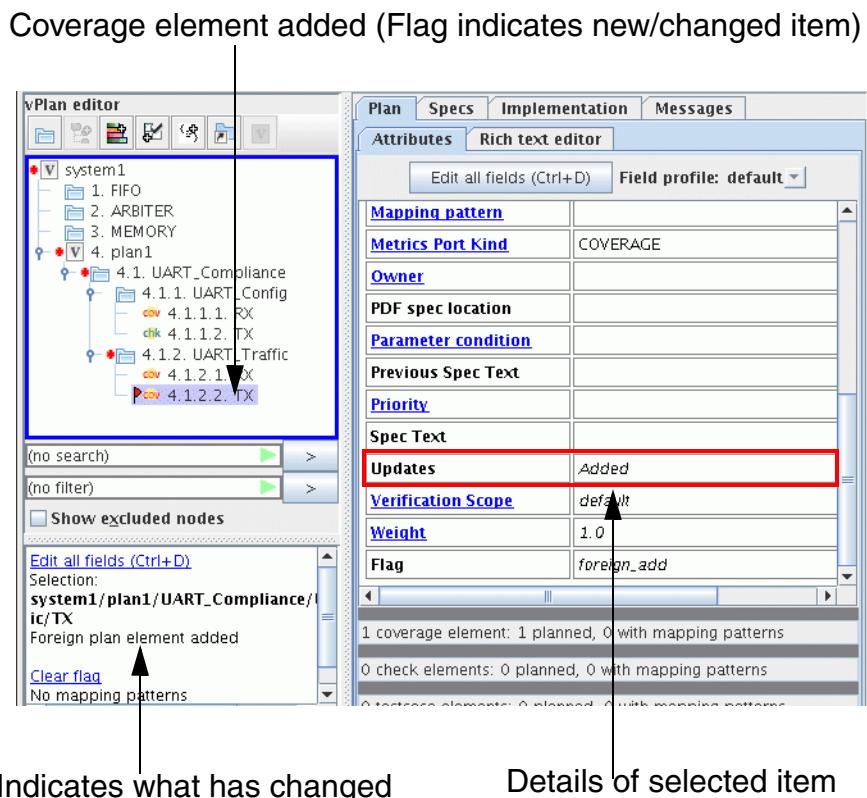
This will synchronize all of the embedded plans.

For example, earlier you imported `plan1.csv` to a top-level plan `system1.vplanx`. Later a few changes, such as a coverage element `TX` was added to one of the sections `UART_Traffic`. You can now consider synchronizing the plan.

To synchronize the plan, click the *Sync embedded* button on the toolbar.

[Figure 4-4 on page 121](#) shows the vPlan hierarchy after synchronization.

**Figure 4-4 vPlan Editor After Synchronization**



After you synchronize a plan, the vPlan hierarchy shows the new or changed elements as flagged items. When you select a flagged item, the message pane (below the vPlan hierarchy) shows what has changed for the selected item. To view a detailed description of the change, check the *Updates* and *Flag* fields in Attributes table in the *Plan* tab page.

### 4.2.2.2 Synchronize Specific Plans

To synchronize specific foreign plans:

1. Select the root node of the embedded plan in the vPlan hierarchy.
2. Right-click and select *Synchronize embedded vPlan*.

This will synchronize only the selected plan.

### 4.2.3 Deleting Embedded Plans

To delete an embedded plan, you can do any of the following:

- Select the embedded plan that you want to delete and press the `Del` key.
- Right-click the embedded plan that you want to delete and select *Delete*.

A confirmation dialog appears and once you confirm the deletion, it is deleted from the vPlan hierarchy.

## 4.3 Creating a Reference to Another Plan

Referencing (Instantiating) a plan into another plan creates a “live link” from the top-level plan to the referenced plan. This approach is recommended if the referenced plan is owned by another group or if it might be modified and you need to maintain consistency. For example, in system-level verification projects, it is useful to create a system-level plan that instantiates the module- or block-level plans. This allows the owners of the lower-level plans to continue to make changes in their plans.

An additional advantage of referencing a plan is that you can easily configure the referenced plan to make it suitable for your environment.

### 4.3.1 Referencing a Plan

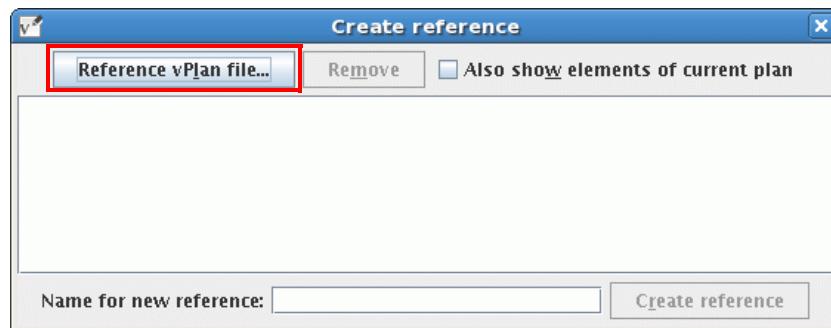
To reference a plan in the current verification plan:

1. Click the *Create Reference* button in the vPlan editor toolbar.

**Note:** Alternatively, you can right-click the top-level plan and select *Reference a vPlan* from the pop-up menu.

The *Create reference* dialog box appears, as shown in [Figure 4-5](#) on page 123.

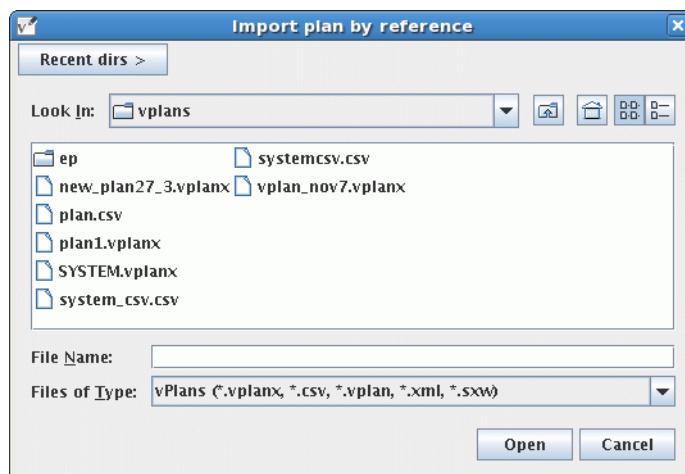
**Figure 4-5 Create reference**



2. Click the *Reference vPlan file* button.

The *Import plan by reference* file chooser box appears, as shown in [Figure 4-6](#) on page 123.

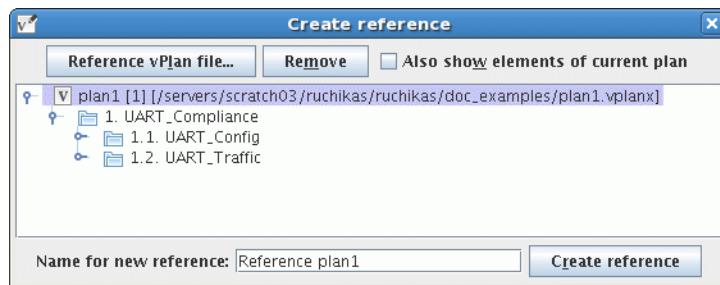
**Figure 4-6 Import plan by reference**



3. Navigate through the hierarchy and select the relevant verification plan that must be included as a reference, and click *Open*. For example, select `plan1.vplanx` and click *Open*.

The selected plan shows in the *Create reference* dialog box, as shown in [Figure 4-7](#) on page 124.

**Figure 4-7 Create reference**



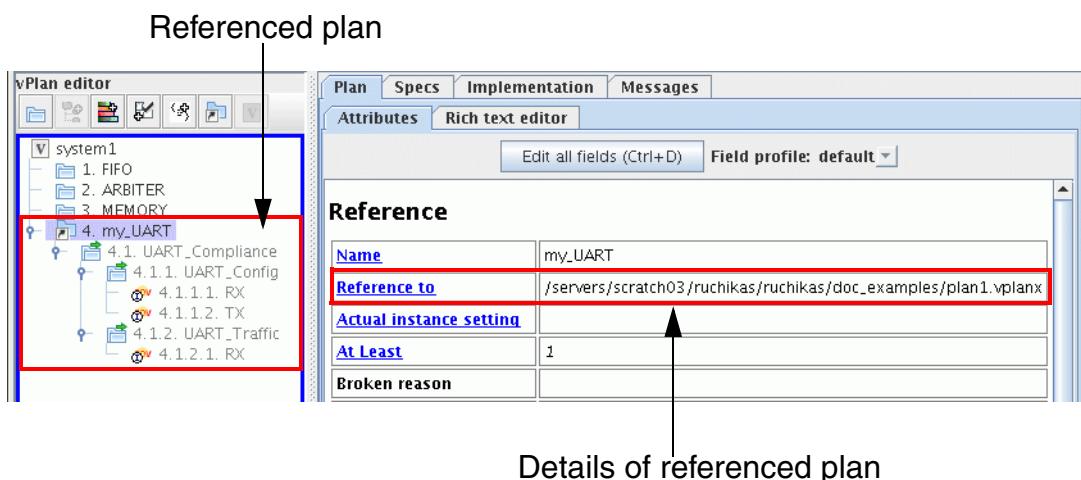
By default, a name is shown for the referenced plan in the text box. You can specify a different name for the referenced plan in the *Name for new reference* text box.

- For example, specify `my_UART` in the text box and click the *Create reference* button.

When you click the *Create reference* button, a reference to the plan is created and the referenced plan shows in the vPlan editor, as shown in [Figure 4-8](#) on page 124.

**Note:** You can also reference an element from within the current plan. To make elements of the current plan available for referencing, select the check box *Also show elements of current plan*. This will show the current plan in the *Create reference* dialog box.

**Figure 4-8 Referenced Plan in vPlan Hierarchy**



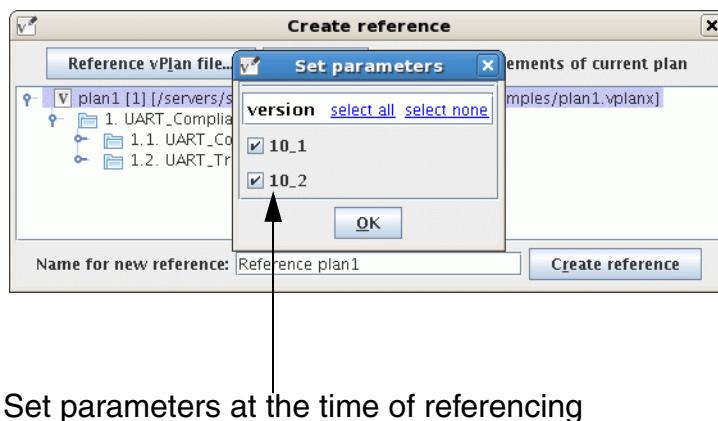
The name you specified in the *Create reference* dialog box shows in the vPlan editor. In addition, the attribute *Reference to* in the *Attributes* table, shows the location of the referenced plan.

**Note:** The referenced plan nodes are shown in gray to indicate that they cannot be edited directly. The referenced plan can be edited only by opening it directly.

### Important

If the plan being referenced is parameterized, then when you click the *Create reference* button, an additional dialog box, as shown in [Figure 4-9](#) on page 125 appears.

**Figure 4-9 Create reference -- Set Parameters**

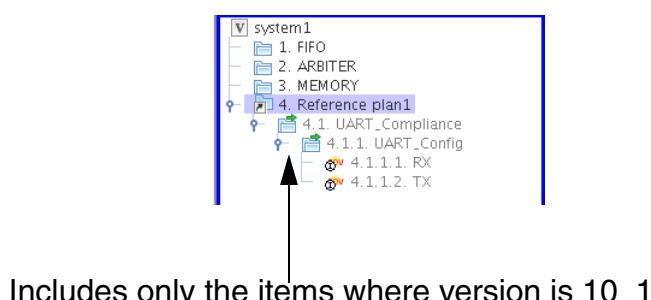


Set parameters at the time of referencing

In the *Set parameters* dialog box, you can select parameter values to configure the new reference. Based on your selection, the reference is created. For example, you want to create a reference such that only the items with version 10\_1 are included. For this, ensure that only 10\_1 is selected in the *Set parameters* dialog box and click *OK*.

After you click *OK*, a reference to the plan is created and the referenced plan is shown in the vPlan editor, as shown in [Figure 4-10](#) on page 125.

**Figure 4-10 Create reference**



Includes only the items where version is 10\_1

Notice that the referenced plan shows only the items matching the parameter setting made in the *Set parameters* dialog box. In this case, only `UART_Config` section is shown because it is the only section from the referenced plan with a parameter condition of `10_1`. The section `UART_Traffic` is not shown because its parameter condition is `10_2`.

For details on creating a parameterized plan, see “[Using Parameters](#)” on page 127.

### 4.3.2 Reloading Referenced Plans

After creating a reference to a plan from the top-level plan, you can reload the referenced plan in order to see the latest changes to the referenced plan in the top-level plan.

To reload a referenced plan, do any of the following:

- Select *Tools* —> *Reload referenced*.
- Click the *Reload referenced* button on the toolbar.

This will reload all of the referenced plans.

### 4.3.3 Deleting a Reference to a Plan

To delete a reference to a plan, you can do any of the following:

- Select the referenced plan that you want to delete and press the `Del` key.
- Right-click the referenced plan that you want to delete and select *Delete*.

A confirmation dialog appears, and once you confirm the deletion, the reference deleted from the *vPlan* hierarchy.

**Note:** Even after deleting all references to a plan, it will still be listed in the *Create reference* dialog purely for convenience. You can remove it using the *Remove* button. A plan is only truly part of the main *vPlan* if it is referenced somewhere in the main hierarchy.

## 4.4 Making Verification Plans Reusable

You can make a verification plan reusable in the following ways:

- [Using Parameters](#)
- [Using Logical Instances](#)

### 4.4.1 Using Parameters

Parameters help you to maintain one verification plan for multiple perspectives. For example, you can define parameters in a verification plan and use them to configure parts of the plan for different versions of the product.

Use of parameters is extremely useful when you have a top-level plan that references a separate plan for each subcomponent. For example, in system-level verification projects, it is useful to create a system-level plan that references the module- or block-level plans. You can use parameters to configure the referenced plans appropriately for your project.

To enable reuse of plans using parameters, you must:

- [Declare Parameters](#)
- [Associate Parameters with Appropriate Sections or Coverage Items](#)

#### 4.4.1.1 Declare Parameters

To declare a set of parameters:

1. Right-click anywhere in the vPlan hierarchy and select *Create a parameter* from the pop-up menu.

The *Create parameter* dialog box appears, as shown in [Figure 4-11](#) on page 127.

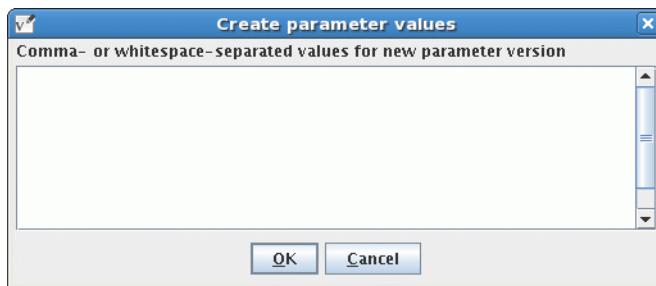
**Figure 4-11 Create Parameter**



2. Type the name of the parameter and click *OK*. For example, specify the name as *version* and click *OK*.

The *Create parameter values* dialog box appears, as shown in [Figure 4-12](#) on page 128.

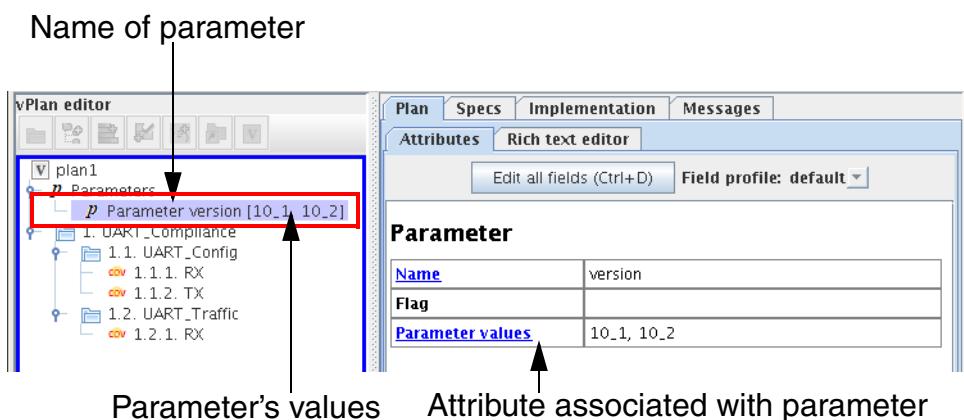
**Figure 4-12 Create Parameter Values**



3. Specify the valid values for the parameter and click *OK*. For example, to specify the value for `version` as `10_1` and `10_2`, specify `10_1, 10_2` in the text box and click *OK*.

The parameter is created and is highlighted in the *vPlan* editor, as shown in [Figure 4-13](#) on page 128.

**Figure 4-13 vPlanner Window with the New Parameter**



Notice that a parameter named `version` is created and it has two valid values, `10_1` and `10_2`.

### 4.4.1.2 Associate Parameters with Appropriate Sections or Coverage Items

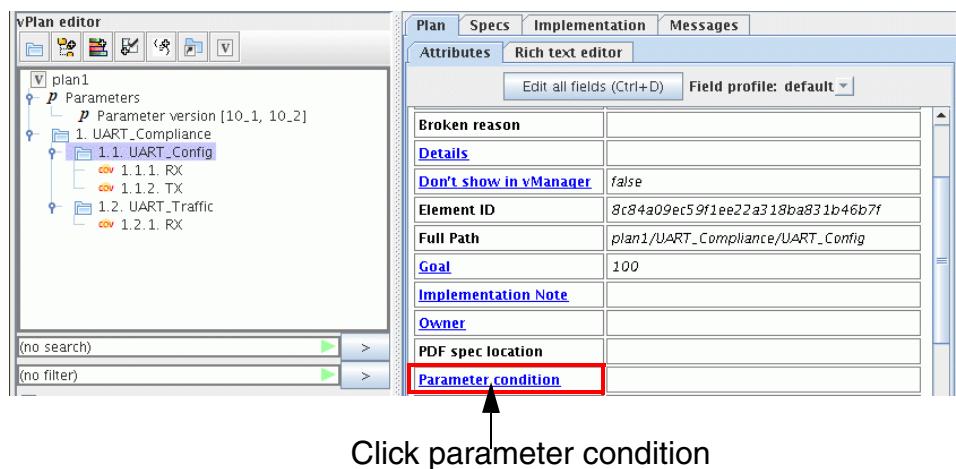
After declaring a set of parameters, you can associate values with appropriate sections or coverage items in the verification plan.

For example, consider the plan shown in [Figure 4-13](#) on page 128. The plan has two sections, `UART_Config` and `UART_Traffic`. You can consider associating parameter `version` value `10_1` with `UART_Config` and parameter `version` value `10_2` with `UART_Traffic`.

To associate parameter values with appropriate sections or coverage items:

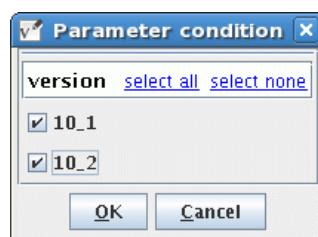
1. Select the section with which you want to associate the parameter. For example, select *UART\_Config*.
2. Click *Parameter condition* in the Attributes table on the Plan page. [Figure 4-14](#) on page 129 shows the *Parameter condition* attribute.

**Figure 4-14 Parameter Condition on the Plan Page**



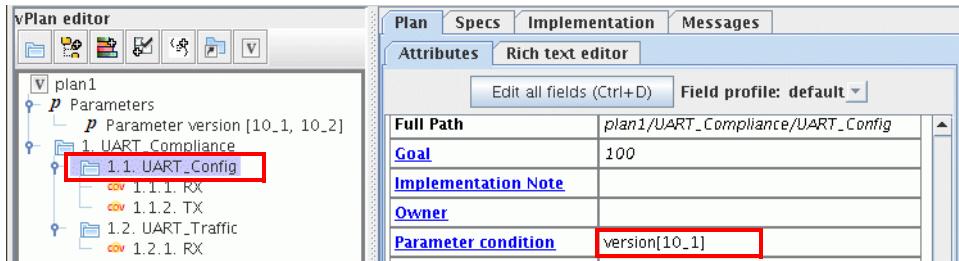
3. The *Parameter condition* dialog box appears, as shown in [Figure 4-15](#) on page 129. By default, all values show selected. Deselect the values you do not want to associate with the selected element and click *OK*. For example, deselect *10\_2* and click *OK*.

**Figure 4-15 Parameter Condition**



The parameter's value is now associated with the selected vPlan section, as shown in [Figure 4-16](#) on page 130.

**Figure 4-16 Parameter Associated with vPlan Section**



Notice that *version 10\_1* is now associated with *UART\_Config*. Similarly, you can associate more values with other vPlan sections and elements.

**Note:** You can associate any number of parameters with a section or vPlan element.

Declaring parameters makes the plan reusable. For information on reusing a parameterized plan, see [“Referencing a Plan”](#) on page 122.

### 4.4.2 Using Logical Instances

Plans that are designed to be reused must use logical instance pathnames instead of absolute pathnames to map vPlan elements to the coverage or check ports. Using logical instance pathnames allows a consumer of the reusable plan to instantiate the verification component anywhere in the testbench without breaking the mappings.

For example, if a UART compliance plan defined a coverage port with the following absolute path, a consumer of the plan would have to either ensure that the actual instance path of the instantiated agent in the testbench matched this path exactly or modify the plan to reflect the actual instance path.

```
sys.cdn_uart.cdn_uart_if.tx_agent.monitor.mon_frame_done.databit_type
```

However, if the UART compliance plan identifies the term **UART\_AGENT** as a logical instance and defines the coverage port as follows, the consumer of the plan can easily reuse the plan:

```
$ (UART_AGENT).monitor.mon_frame_done.databit_type
```

You can define your own logical instances and use them in actual instance paths. For example, if your plan has two instances of the UART agent with the following actual pathnames, you can define a logical instance **UART\_APB** with the actual path

`sys.cdn_uart_apb`. This will improve readability and reduce the possibility of typos:

```
sys.cdn_uart_apb.cdn_uart_if.tx_agent
```

```
sys.cdn_uart_apb.cdn_uart_if.rx_agent
```

Creating a logical instance allows consumers of your plan to focus on the component, independent of the design or test structure around it.

To enable reuse of plans using logical instances, you must:

- [Create Logical Instances](#)
- [Mapping Metrics Relative to Logical Instances](#)

### 4.4.2.1 Create Logical Instances

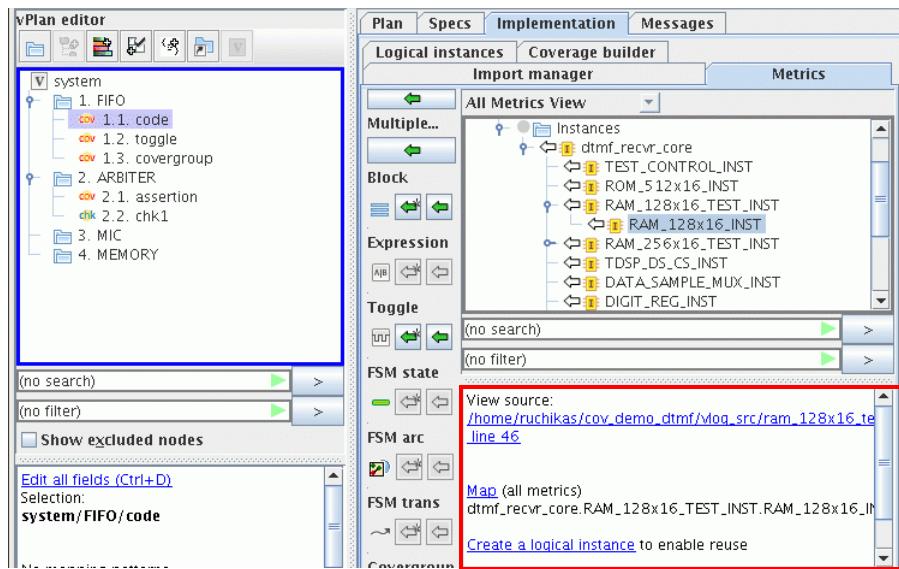
Consumers of a reusable plan most probably have multiple testbenches with different hierarchies. Consumers must be able to instantiate verification components anywhere in their testbench hierarchy without breaking the vPlan mappings. To enable this, you must use logical instance pathnames instead of absolute pathnames in the vPlan mappings. Then, once the testbench hierarchy is defined, the consumer can map logical instances to an actual instance paths.

**Note:** Logical instances cannot be created within the type-based part of the metrics hierarchy.

To create a logical instance:

1. Open the verification plan.
2. Load the coverage model. See [Loading Coverage Data](#) on page 143.
3. Select the part of the model where you want to create a logical instance. For example, select *RAM\_128x16\_INST*, as shown in [Figure 4-17](#) on page 132.

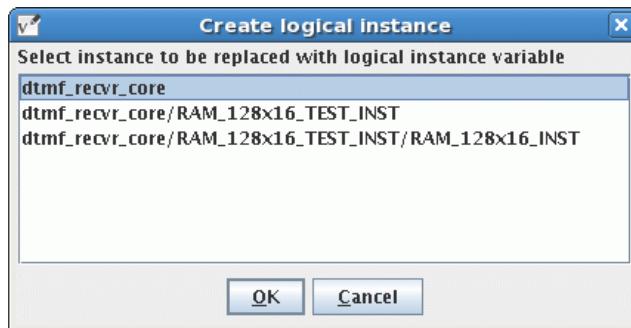
**Figure 4-17 Create Logical Instance**



- To create a logical instance, select the *Create a logical instance* link.

The *Create logical instance* dialog box appears, as shown in [Figure 4-18](#) on page 132.

**Figure 4-18 Create Logical Instance**



- Select the path that you want to be used as the base of the logical instance. For example, select *dtmf\_recv\_core/RAM\_128x16\_TEST\_INST* and click *OK*.

Another dialog box appears, as shown in [Figure 4-19](#) on page 133.

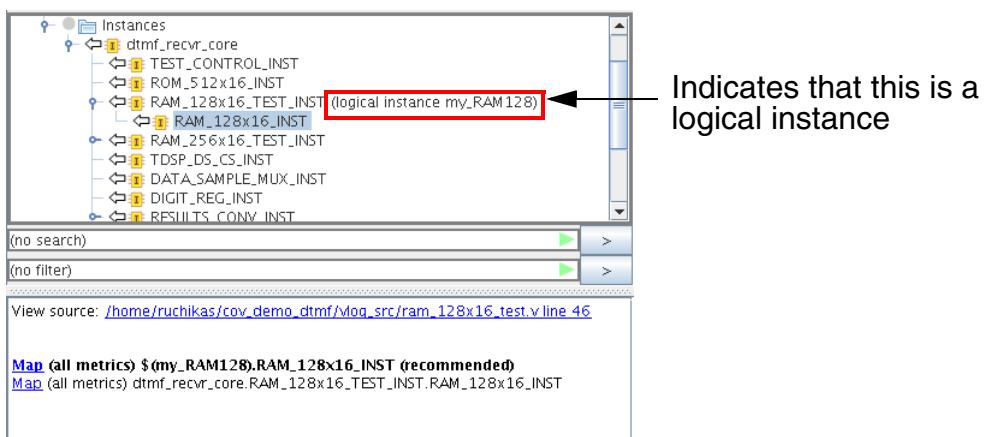
**Figure 4-19 Name Logical Instance**



6. Specify a name for the logical instance and click *OK*. For example, specify *my\_RAM128* and click *OK*.

This instance is now available as a logical instance, as shown in [Figure 4-20](#) on page 133.

**Figure 4-20 Logical Instance**



### ***Alternate Method for Creating Logical Instances***

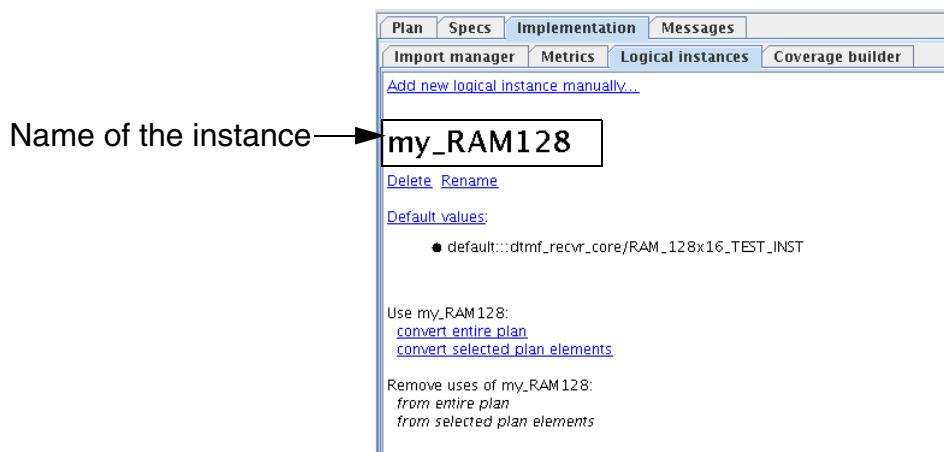
To create a logical instance:

1. Open the verification plan.
2. Load the coverage model. See [Loading Coverage Data](#) on page 143.
3. Select the instance for which you want to create a logical instance.
4. Right-click the instance and select *Create logical instance here* from the pop-up menu.
5. Specify the name for the logical instance and click *OK*.

### Logical Instances Tab Page

When you create a logical instance, it is shown on the *Logical instances* tab page, as shown in [Figure 4-21](#) on page 134.

**Figure 4-21 Logical Instances Tab Page**



For each logical instance, the following set of links appear:

- Delete—Allows you to delete the logical instance. When you click this link, a confirmation dialog box appears. Once you confirm deletion, the logical instance is removed.  
**Note:** When deleting a logical instance that is mapped to any of the items, remember that the mappings are not removed. Rather, the mappings are automatically converted to use absolute pathnames. For more details, see [“Removing Logical Instances”](#) on page 137.
- Rename—Allows you to rename the logical instance.
- Default values—Allows you to specify or change the path to be used for this logical instance.
- Convert entire plan—Allows you to replace the absolute path mappings involving this instance with the logical instance mapping in the entire plan. For more details, see [“Upgrading Absolute pathnames to use logical instances”](#) on page 140.
- Convert selected plan elements —Allows you to replace the absolute path mappings involving this instance with the logical instance mapping for the selected elements in the plan. For more details, see [“Upgrading Absolute pathnames to use logical instances”](#) on page 140.

- Remove uses from entire plan—Allows you to remove the use of this logical instance from the entire plan. This shows as a link only if any metric port is already mapped to this logical instance. Else, it appears as text.

**Note:** This will only remove the use of logical instance. It will not remove the mappings. All mappings will be automatically converted to use absolute pathnames. For more details, see [“Removing the Use of Logical Instance Mapping”](#) on page 138.

- Remove uses from selected plan elements —Allows you to remove the use of this logical instance from the selected elements in the plan. This shows as a link only if any metric port is already mapped to this logical instance. Else, it appears as text.

**Note:** This will only remove the use of logical instance. It will not remove the mappings. The selected mappings will be automatically converted to use absolute pathnames. For more details, see [“Removing the Use of Logical Instance Mapping”](#) on page 138.

### 4.4.2.2 Mapping Metrics Relative to Logical Instances

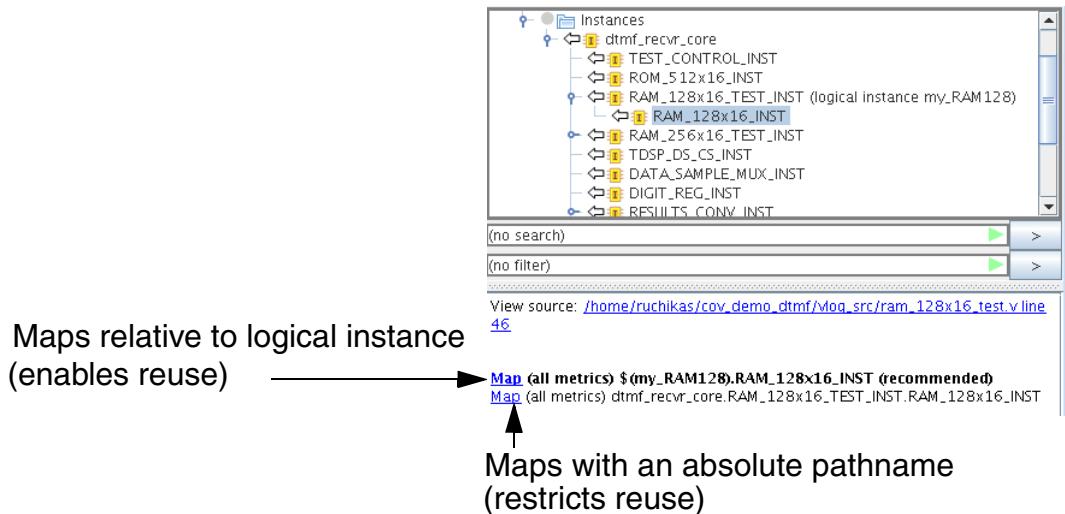
After creating a logical instance, you can use it automatically when mapping items within that instance. Mappings created relative to a logical instance will be reusable in higher plans.

To create mappings relative to a logical instance:

1. Select the metric item that you want to map to a metric port in the vPlan hierarchy. For example, select *RAM\_128x16\_INST*.

The message pane below the verification hierarchy shows two links, as shown in [Figure 4-22](#) on page 136.

**Figure 4-22 Apply Mappings**



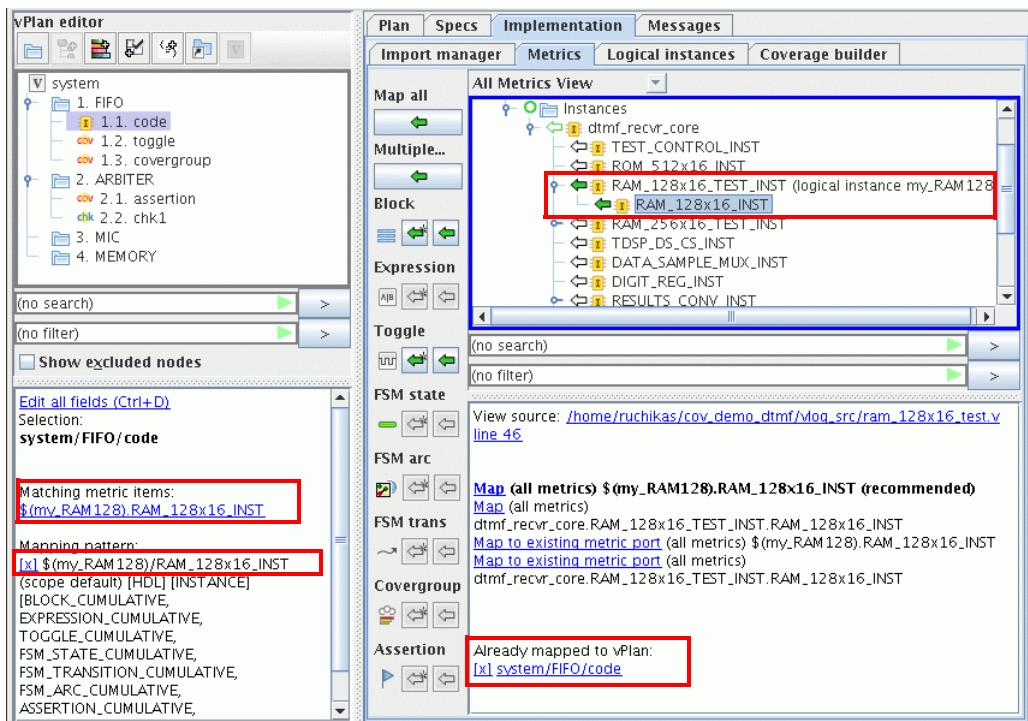
Notice that the pane below the verification hierarchy shows two links.

- ❑ The first link is labeled `recommended` because this is a logical instance. When you create mappings with this link, the mappings are not broken when this verification plan is imported or referenced in a top-level plan.
- ❑ The second link is not a logical instance. This link enables mapping using absolute pathnames and therefore, when this plan will be imported or referenced from another plan, the mappings might break.

**2. To map items such that the links do not break at the time of reuse, click the first link.**

The vPlanner window shown in [Figure 4-23](#) on page 137 shows the mapped item.

**Figure 4-23 Logical Instance Mapping**



Notice that the mapping pattern begins with `$(my_RAM128)` in the message pane below the vPlan editor. This indicates that the instance is mapped using the logical instance.

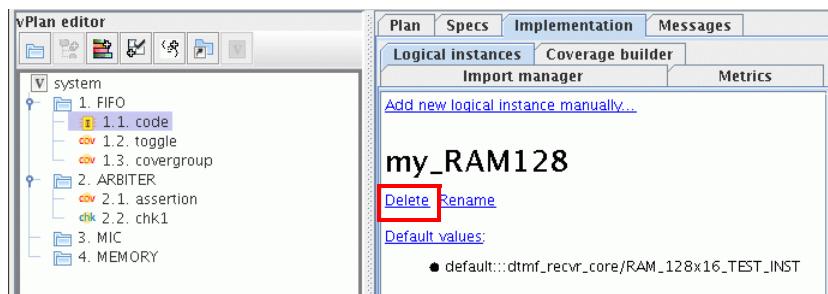
### 4.4.2.3 Removing Logical Instances

To remove a logical instance:

1. Go to the *Logical instances* tab page.

[Figure 4-24](#) on page 137 shows the *Logical instances* tab page.

**Figure 4-24 Logical Instances Tab Page**



2. Click the *Delete* link below the logical instance name.
3. A confirmation dialog appears. Click *OK* to confirm deletion.

Once you confirm deletion, the logical instance is removed.

**Note:** If the logical instance being deleted is mapped to any of the items, the mappings are not removed. Rather, the mappings are automatically converted to use absolute pathnames.

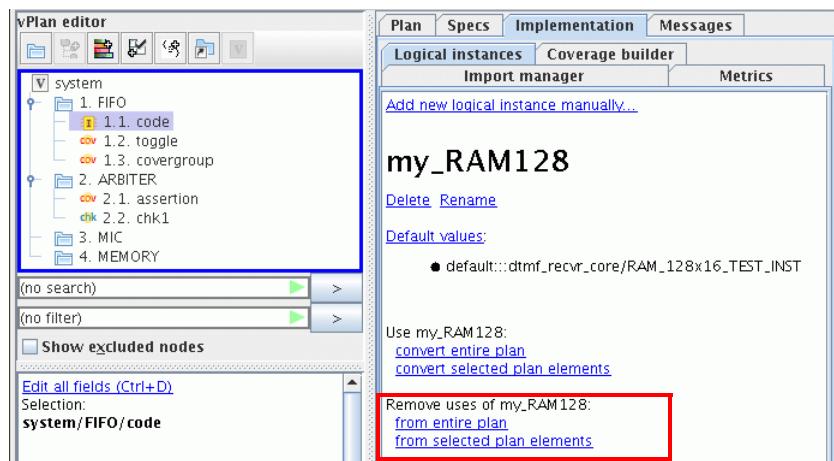
### 4.4.2.4 Removing the Use of Logical Instance Mapping

To remove the use of logical instance mapping:

1. Go to the *Logical instances* tab page.

[Figure 4-25](#) on page 138 shows the *Logical instances* tab page.

**Figure 4-25 Logical Instances Tab Page**



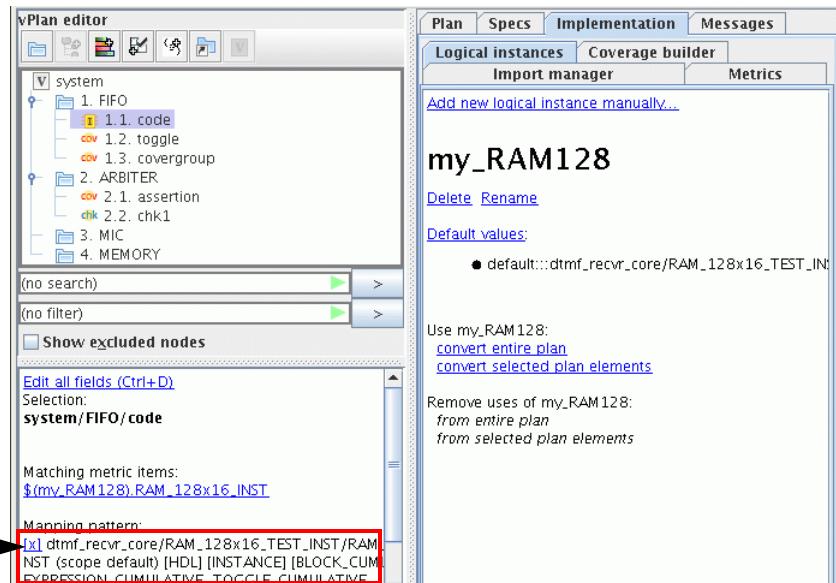
2. Click one of the following:

- From entire plan*—To remove the use of this logical instance from the entire plan.
- From selected plan elements*—To remove the use of this logical instance from the selected elements in the plan.

**Note:** This will only remove the use of logical instance. It will not remove the mappings. All mappings will be automatically converted to use absolute pathnames.

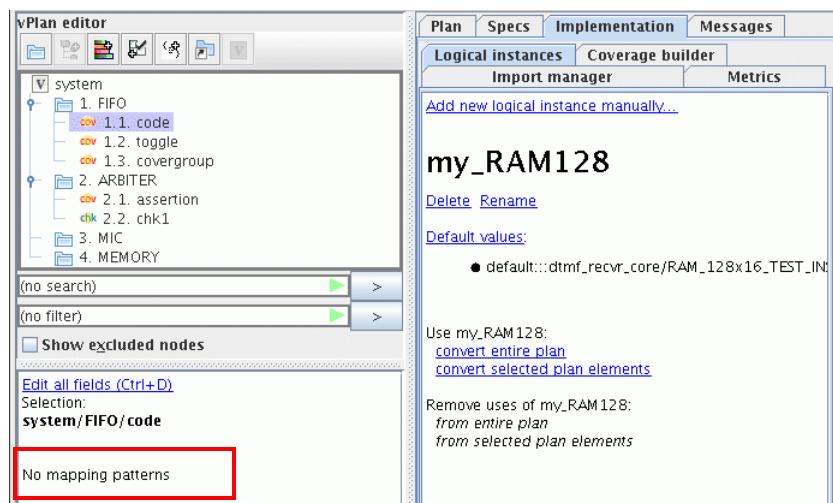
3. After the mapping is converted to use absolute pathnames, you can remove the mapping, if required. In this case, to remove the mapping, click the [x] symbol in the message pane below the vPlan editor, as shown in [Figure 4-26](#) on page 139.

**Figure 4-26 Remove Logical Instance Mapping**



This will remove the mapping, as shown in Figure 4-27 on page 139.

**Figure 4-27 Logical Instance Mapping Removed**



The message pane below the vPlan editor indicates that no mapping pattern is associated with the selected vPlan element.

### 4.4.2.5 Upgrading Absolute pathnames to use logical instances

If your plan currently contains mappings that use absolute pathnames, you can define logical instances and upgrade the mappings for all the metric items under that logical instance to use pathnames relative to the logical instance.

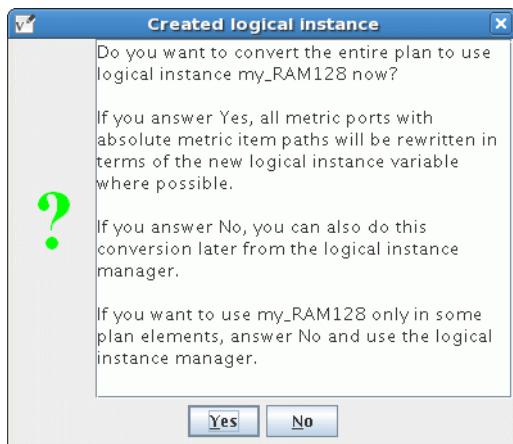
To update absolute pathnames to logical instance paths:

1. Create a logical instance. See “[Create Logical Instances](#)” on page 131 for details on creating a logical instance. If the item for which you are creating a logical instance is already mapped to some of the elements in the verification plan, then a confirmation dialog box appears, as shown in [Figure 4-29](#) on page 141. You can choose to convert absolute path names with logical instances at the time of creation, or convert them later. In case you want to use logical instance only for some of the elements in the plan, select *No*.

**Note:** Selecting a *Yes*, converts all the plan items to use logical instances.

In this case, select *No*.

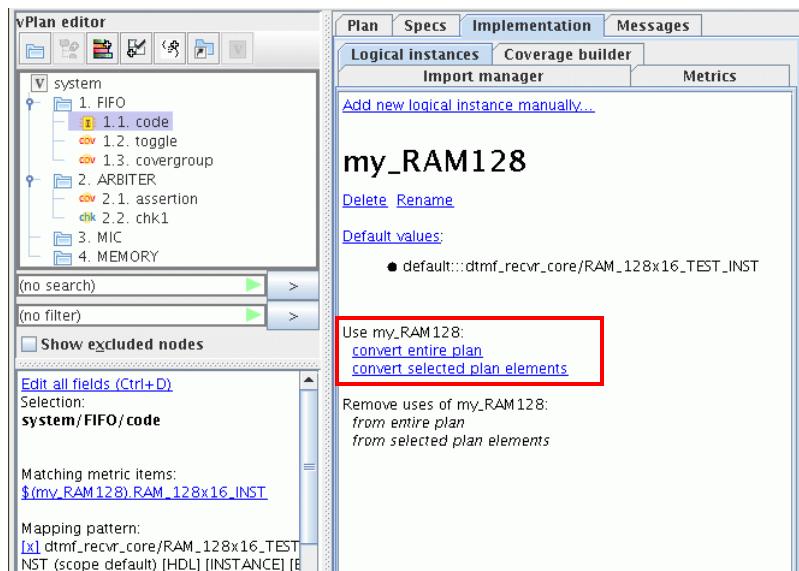
**Figure 4-28 Logical Instances Created**



2. After creating a logical instance, go to the *Logical instances* tab page.

[Figure 4-29](#) on page 141 shows the *Logical instances* tab page.

**Figure 4-29 Logical Instances Tab Page**



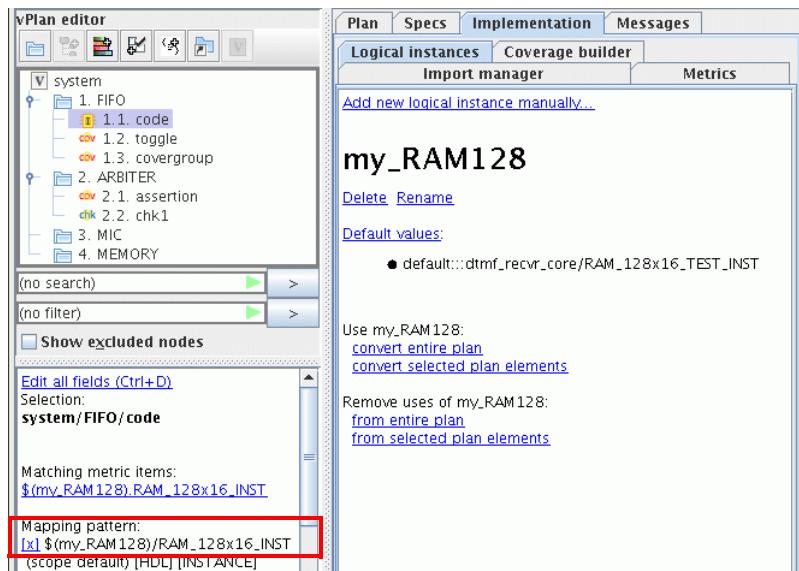
**3. Click one of the following:**

- Convert entire plan*—To replace the absolute path mappings of this instance with the logical instance mapping in the entire plan.
- Convert selected plan elements*—To replace the absolute path mappings of this instance with the logical instance mapping for the selected elements in the plan.

For example, in this case select *Convert selected plan elements*.

This converts the absolute pathname mapping of instance *RAM\_128x16\_INST* with logical instance mapping, as shown in [Figure 4-30](#) on page 142.

**Figure 4-30 Absolute Pathname converted to Logical Instance**



Notice that the mapping pattern has changed to reflect the use of logical instance instead of absolute pathname.

Similarly, you can upgrade more such instances to use logical instance mapping instead of absolute pathnames.

---

# Creating the Coverage Model

---

After creating the base verification plan, for each planned metric port you can map coverage metrics. You can also use the Functional Coverage Builder and create new **e** coverage metrics. This chapter covers the following topics:

- [Loading Coverage Data](#)
- [Unloading Coverage Data](#)
- [Mapping Existing Coverage Metrics to the vPlan](#)
- [Viewing Mappings](#)
- [Deleting Mappings](#)
- [Searching and Filtering Coverage Items in the Metrics Tree](#)
- [Creating Bin Filters](#)
- [Using the Functional Coverage Builder](#)

## 5.1 Loading Coverage Data

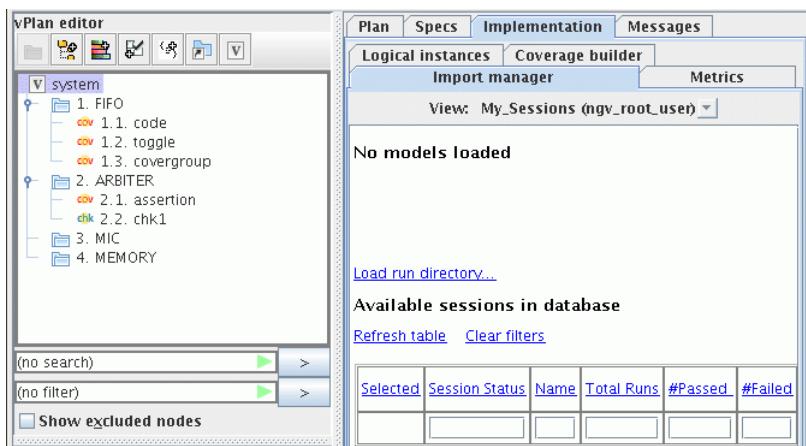
Using vPlanner, you can load coverage data from a run directory or a vsotf file.

### 5.1.1 Loading Coverage Data from Run Directory

To load coverage data from a run directory:

1. Invoke vPlanner and open the verification plan.
2. Click the *Implementation* tab. By default, the *Import manager* tab page is displayed, as shown in [Figure 5-1](#) on page 144.

**Figure 5-1 Import Manager Tab Page**



3. Click *Load run directory*.

The *Load metrics* dialog box opens.

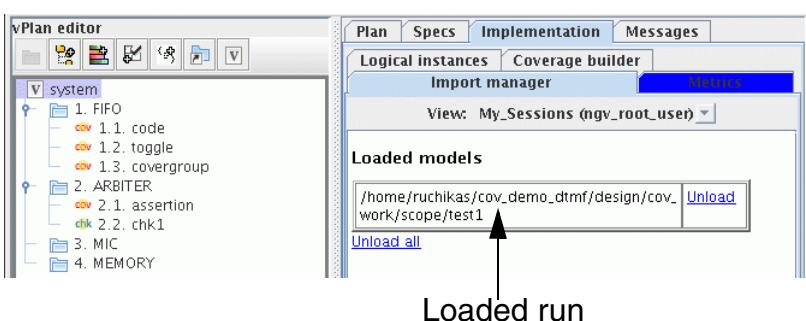
4. Navigate to the run directory that you want to load.

**Note:** In vPlanner, you can also load coverage coming from Palladium.

5. Select the file you want to load and click *OK*.

The loaded run is shown in the table, and the *Metrics* tab page is highlighted, as shown in [Figure 5-2](#) on page 144.

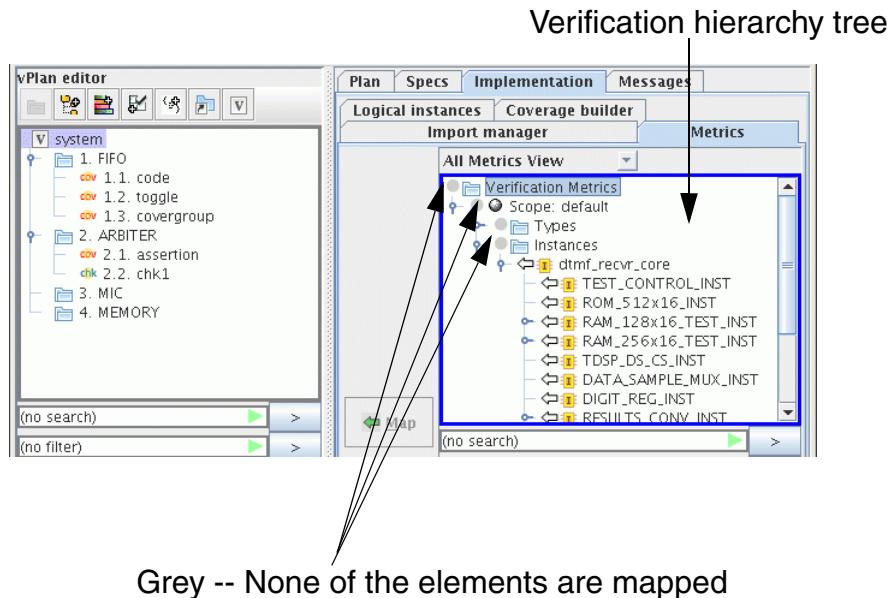
**Figure 5-2 Import Manager Tab Page**



6. Click the *Metrics* tab page to display the verification hierarchy tree.

[Figure 5-3](#) on page 145 shows the verification hierarchy tree in the metrics tab page.

**Figure 5-3 Metrics Tab Page**



In the verification hierarchy tree:

- A grey dot to the left of a metrics tree element indicates that none of the child elements of that element are mapped.
- A hollow green dot or arrow indicates that some child elements are mapped.
- A solid green dot or arrow indicates that all child elements are mapped.

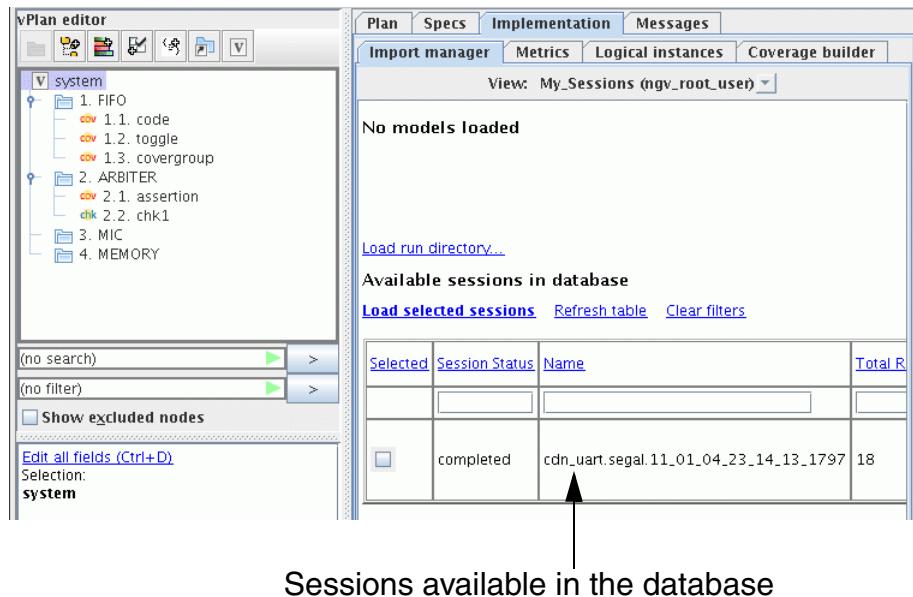
### 5.1.2 Loading Coverage Data from a vsof File

To load coverage data from a vsof file, the sessions data must already be available in the database. For this, you must import the vsof file in vManager. For more details in importing the vsof file in vManager, see *vManager User Guide*.

After importing the required vsof files in vManager, perform the following steps:

1. Invoke vPlanner and open the verification plan.
2. Click the *Implementation* tab. The *Import manager* tab page is displayed, as shown in [Figure 5-4](#) on page 146.

**Figure 5-4 Import Manager Tab Page**



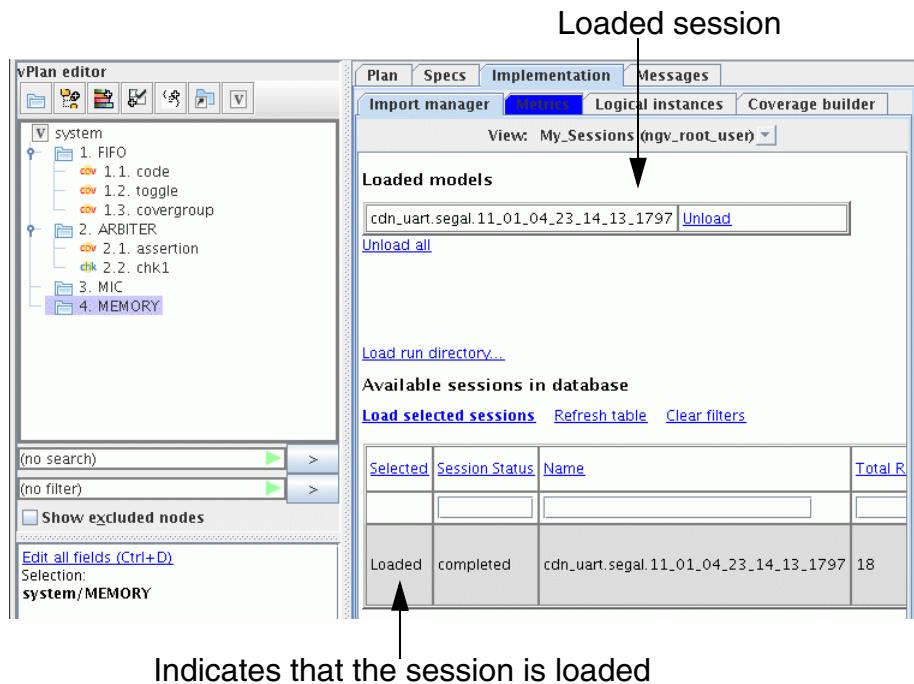
Sessions available in the database

- The table shows the list of available sessions. To load a particular session, select the check box corresponding to that session, and click *Load selected sessions*. For example, to load `cdn_uart.segal.11_01_04_23_14_13_1797`, select the check box corresponding to it, and click *Load selected sessions*.

**Note:** By default, the sessions where you are the owner are displayed. However, you can also list sessions based on other criteria. The row below the table header is the filter row. You can specify a filter criteria and list only the required runs in the table. For example, if you want to list only the sessions that passed, specify *passed* in the text box below the column header *Session Status* and press *Enter*.

The loaded session is shown in the table, and the *Metrics* tab page is highlighted, as shown in [Figure 5-5](#) on page 147.

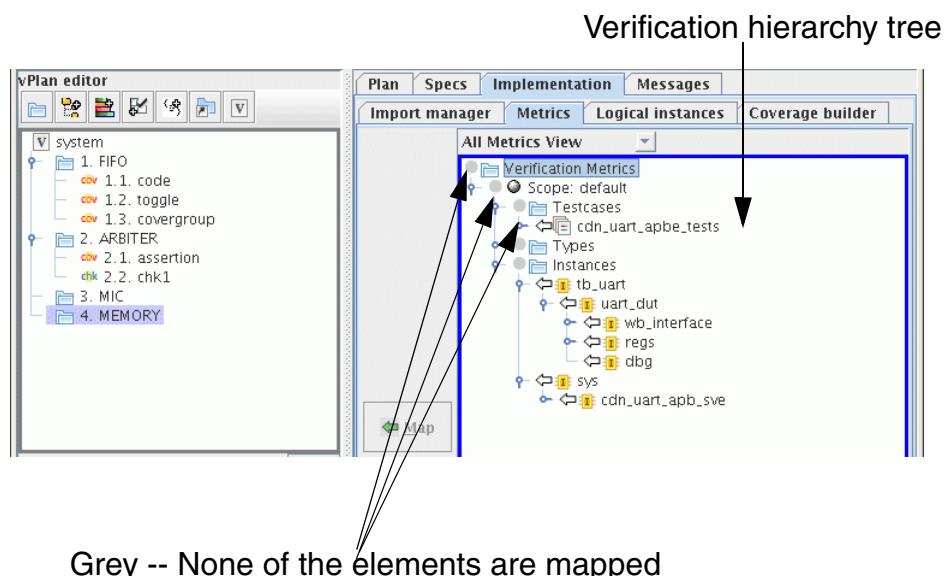
**Figure 5-5 Import Manager Tab Page**



- Click the *Metrics* tab page to display the verification hierarchy tree.

Figure 5-6 on page 147 shows the verification hierarchy tree in the metrics tab page.

**Figure 5-6 Metrics Tab Page**



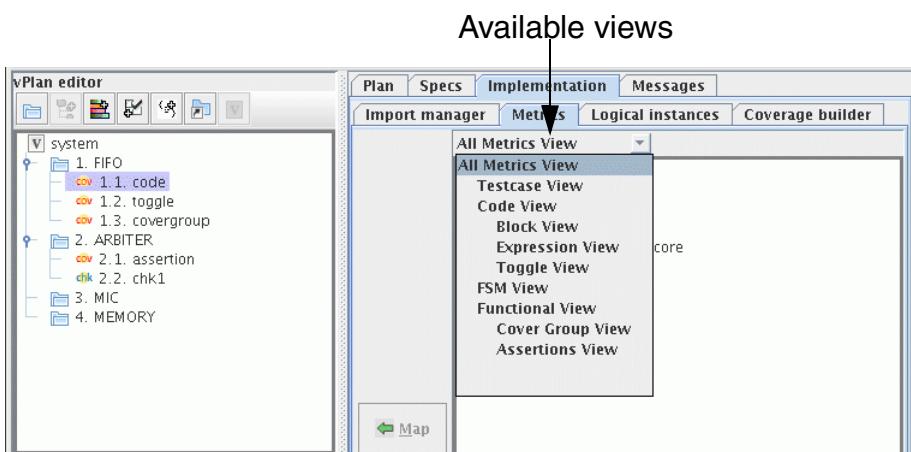
In the verification hierarchy tree:

- A grey dot to the left of a metrics tree element indicates that none of the child elements of that element are mapped.
- A hollow green dot or arrow indicates that some child elements are mapped.
- A solid green dot or arrow indicates that all child elements are mapped.

## 5.2 Listing Specific Items in Verification Hierarchy Tree

When you load coverage, by default, all the items are listed in the verification hierarchy tree. However, you can control the items to be listed in the verification hierarchy tree using the *All Metrics View* drop-down shown in [Figure 5-7](#) on page 148.

**Figure 5-7 All Metrics View Drop-Down**

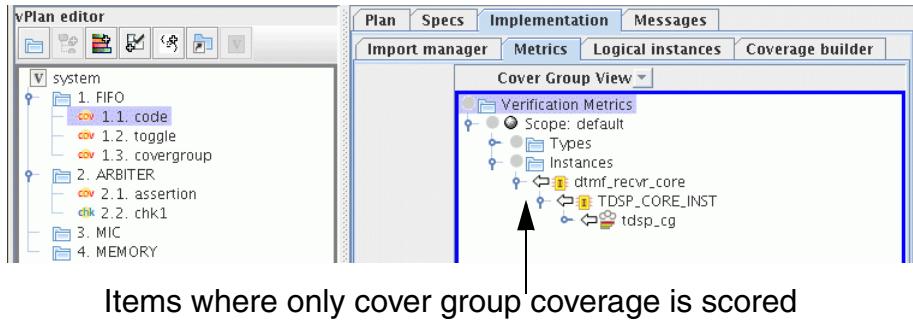


Using the *All Metrics View* drop-down, you can quickly identify the instances or types in which a specific metrics is scored.

For example, to view the types or instances in which cover group coverage is scored, select *Cover Group View* from the *All Metrics View* drop-down. This will filter the verification hierarchy tree to list only the instances or types in which cover group coverage is scored.

[Figure 5-8](#) on page 149 shows the verification tree with *Cover Group View* selected.

**Figure 5-8 Cover Group View**



Notice that the verification tree now shows only the items where cover group coverage is scored.

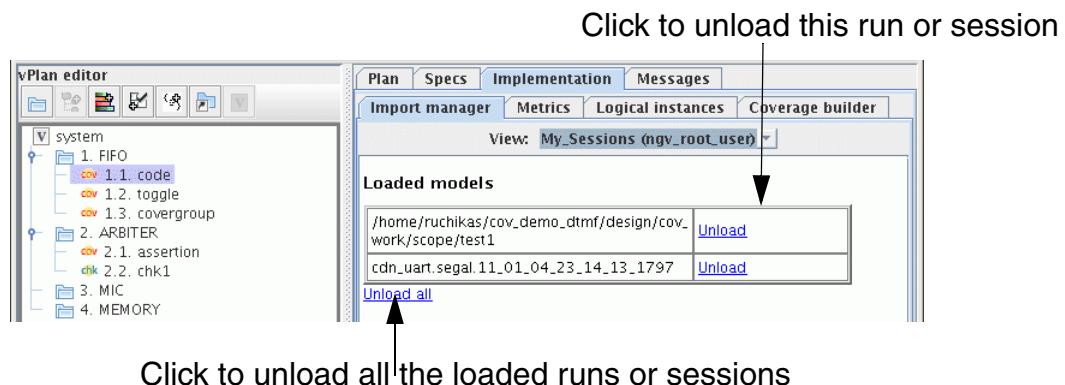
### 5.3 Unloading Coverage Data

To unload coverage data:

1. Select the *Implementation* tab and then select *Import manager*.

The loaded runs are shown in the table, as shown in [Figure 5-9](#) on page 149.

**Figure 5-9 Import Manager Tab Page**



2. Click *Unload* next to the run directory or the session that you want to remove.

A dialog box appears, asking you to confirm the deletion. Once you confirm, the coverage run or session is unloaded from the verification plan.

**Note:** You can click the *Unload all* link to unload all the loaded runs or sessions.

## 5.4 Mapping Existing Coverage Metrics to the vPlan

After loading the coverage run or the session data, you can map coverage metrics to vPlan elements.

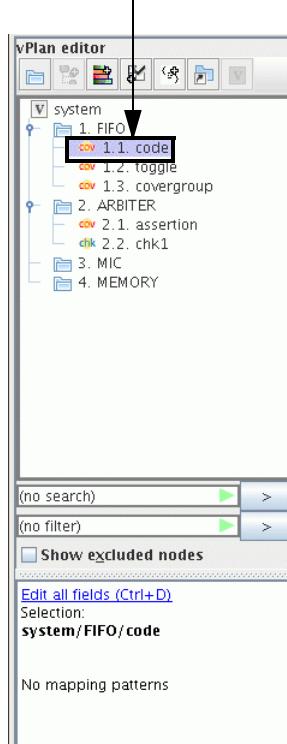
1. Select the vPlan element that you want to map. For example, to map coverage metric port *code* to the metric item *TEST\_CONTROL\_INST*, select *code* in the vPlan editor.

**Note:** You can select a single vPlan element and map multiple metrics to it. Or, you can select multiple vPlan elements and map a single metrics to it.

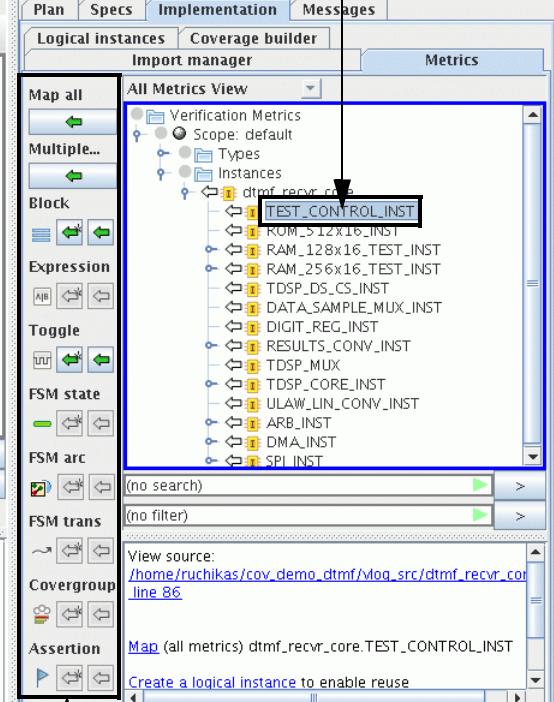
2. Select the coverage metrics in the metrics hierarchy tree. For example, select *TEST\_CONTROL\_INST* in the metrics hierarchy tree.
3. Click the relevant green arrow button, as shown in [Figure 5-10](#) on page 150.

**Figure 5-10** Mapping Coverage Metrics

1. Select the vPlan element.



2. Select the item for mapping.



3. Select the mapping by clicking the relevant arrow button.

You can select any of the following:

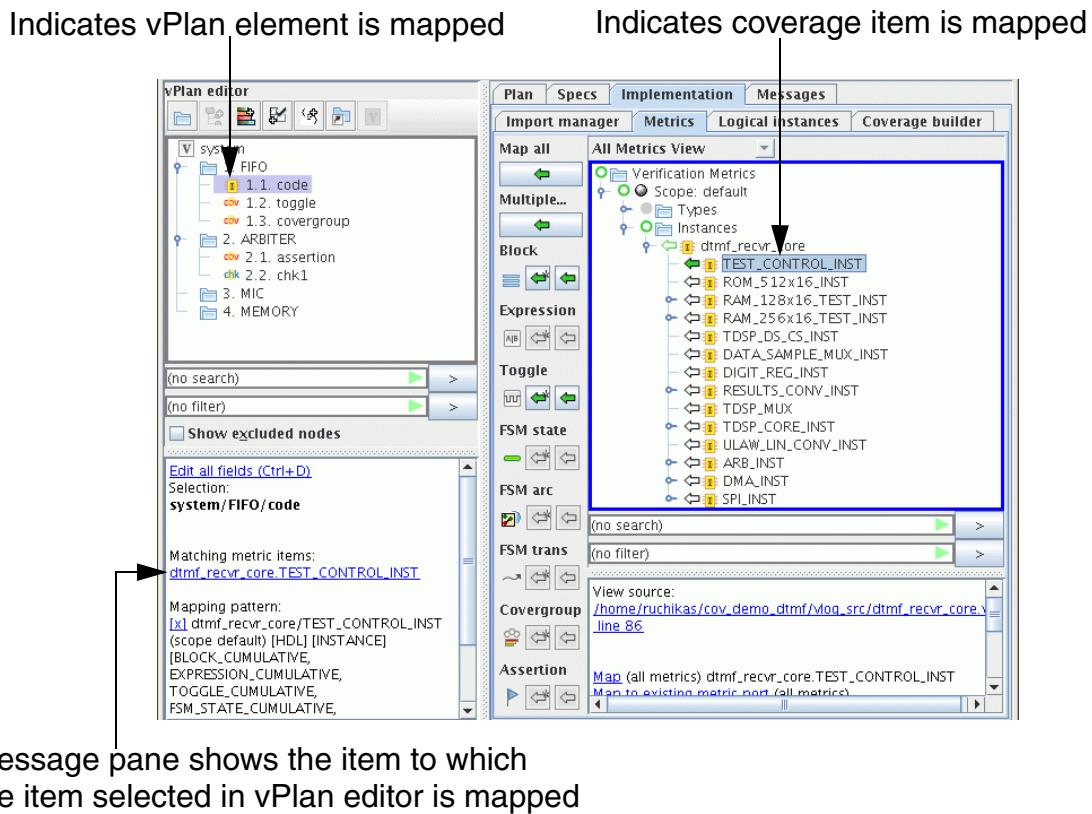
- Map all — Maps cumulative coverage of all applicable metrics types.
- Multiple... — Opens a dialog box from which you can select the metrics types and the grade for mapping.
- Block — Has two buttons: The button with an \* is for mapping block cumulative coverage and the other button (without the \*) is for mapping block local coverage.
- Expression — Has two buttons: The button with an \* is for mapping expression cumulative coverage and the other button (without the \*) is for mapping expression local coverage.
- Toggle— Has two buttons: The button with an \* is for mapping toggle cumulative coverage and the other button (without the \*) is for mapping toggle local coverage.
- FSM state— Has two buttons: The button with an \* is for mapping FSM state cumulative coverage and the other button (without the \*) is for mapping FSM state local coverage.
- FSM arc— Has two buttons: The button with an \* is for mapping FSM arc cumulative coverage and the other button (without the \*) is for mapping FSM arc local coverage.
- FSM trans — Has two buttons: The button with an \* is for mapping FSM transition cumulative coverage and the other button (without the \*) is for mapping FSM transition local coverage.
- Covergroup — Has two buttons: The button with an \* is for mapping covergroup cumulative coverage and the other button (without the \*) is for mapping covergroup transition local coverage.
- Assertion— Has two buttons: The button with an \* is for mapping assertion cumulative coverage and the other button (without the \*) is for mapping assertion local coverage.

**Note:** A disabled button indicates that the specified metrics is not available for the selected item.

**Note:** You can also click the *Map* link in the message pane shown below the verification hierarchy tree to map the selected item. This will map the cumulative coverage of all applicable metrics types. Another link *Create a logical instance* appears below the *Map* link. For more details on logical instances, see [Create Logical Instances](#) on page 131.

After performing the above steps, *TEST\_CONTROL\_INST* (in the verification hierarchy tree) is mapped to *code* (in the vPlan editor), as shown in [Figure 5-11](#) on page 152.

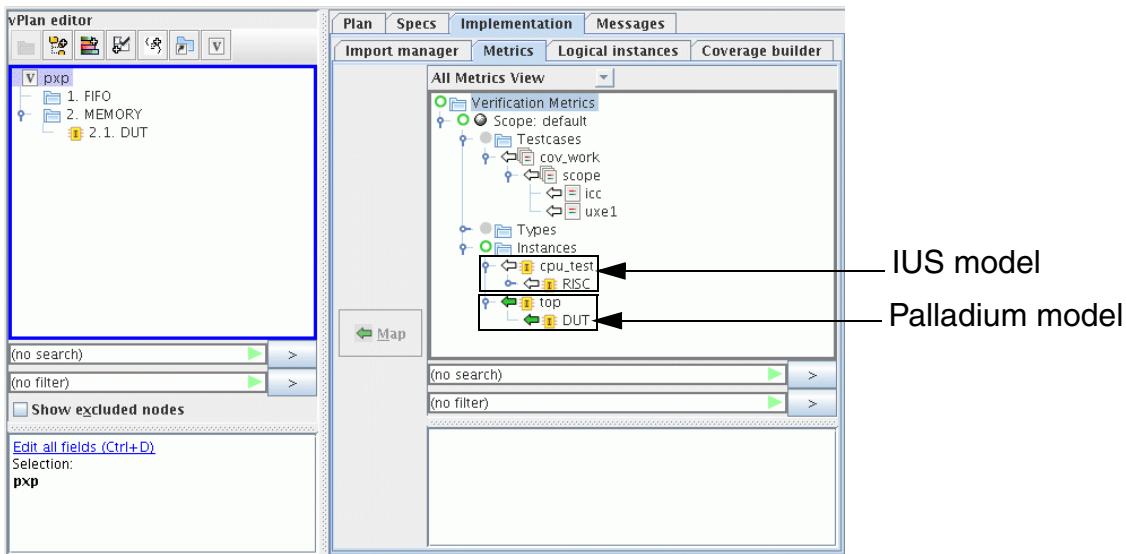
**Figure 5-11 Mapped Coverage Metrics**



In the above figure, note that the icon next to the vPlan item has changed. In the *Metrics* tab page, the coverage item that is mapped shows a solid green arrow. A hollow green dot or arrow indicates that some of the items in the hierarchy are mapped.

**Note:** You can also apply mappings from coverage coming from Palladium. However, the coverage coming from Palladium is not merged with coverage coming from IUS, as shown in [Figure 5-12](#) on page 153.

**Figure 5-12 Coverage from IUS and Palladium**



In the above figure, coverage coming from Palladium is mapped to a vPlan element.

### 5.4.1 Alternate Methods to Apply Mappings

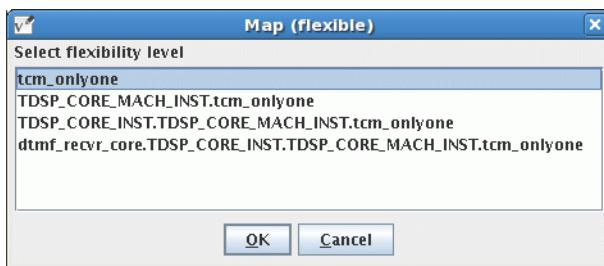
You can also apply mappings by performing the following steps:

1. Select the metric item in the coverage hierarchy tree.
2. Right-click and select any of the following:
  - Map—This option maps the selected coverage item to the item selected in the vPlan hierarchy tree. If the selected vPlan item is already mapped to another coverage item, then a new metric port is created in the vPlan hierarchy tree and the mapping is applied to the newly created item.
  - Map to selected metric port—This option maps the selected coverage item to the item selected in the vPlan hierarchy tree. Unlike the *Map* option, the *Map to selected metric port* option prevents creating of a new metric port if the selected vPlan item is already mapped to another coverage item. This option is useful when you want to map a single vPlan item to multiple coverage items. For more details, see [Mapping Multiple Coverage Metrics to a Single vPlan Item](#) on page 155.

**Note:** In the case of *Map* and *Map to selected metric port*, the mapping pattern created is similar to sys/a/b/c/d/e.

- Map (flexible) —This option is similar to the *Map* option with an additional facility of specifying a flexibility level at the time of mapping. It is useful when you want to map multiple items without explicitly listing all of the items. If you select this option, an additional dialog box appears, for instance, as shown in [Figure 5-14](#) on page 155.

**Figure 5-13 Mapping Flexible**



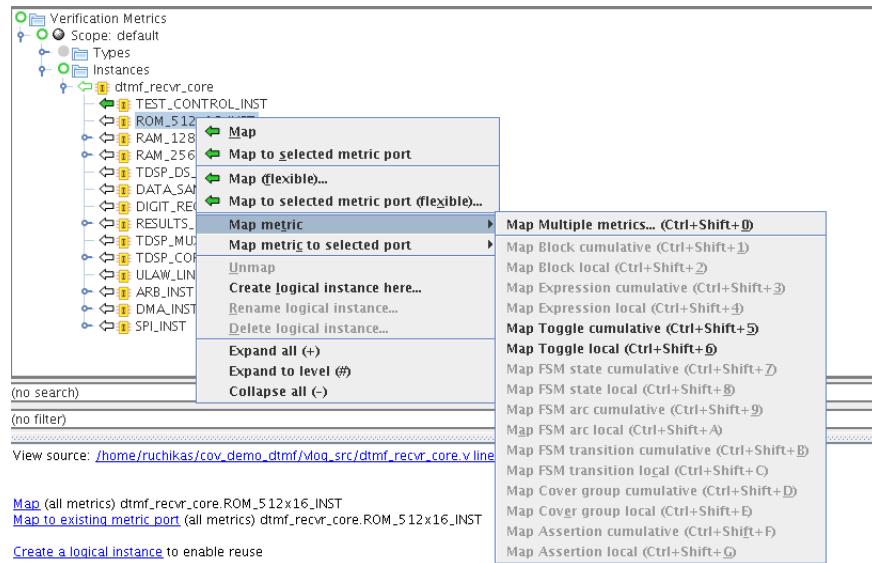
You can select the flexibility level and click *OK*. Based on your selection, the mapping pattern is created which can be of any of the following types:

- .../e
- .../d/e
- .../c/d/e

where ... is a wildcard. For more details, see [Using Wildcards While Specifying Mapping Patterns](#) on page 160.

- Map to selected metric port (flexible) —This option is similar to the *Map to selected metric port* option, with an additional facility of specifying a flexibility level at the time of mapping. See *Map (flexible)* for more details.
- Map metric—This option allows you to select the metric and its grade (cumulative or local) at the time of mapping. This option opens another pop-up menu, as shown in [Figure 5-14](#) on page 155.

**Figure 5-14 Mapping Coverage Metrics**



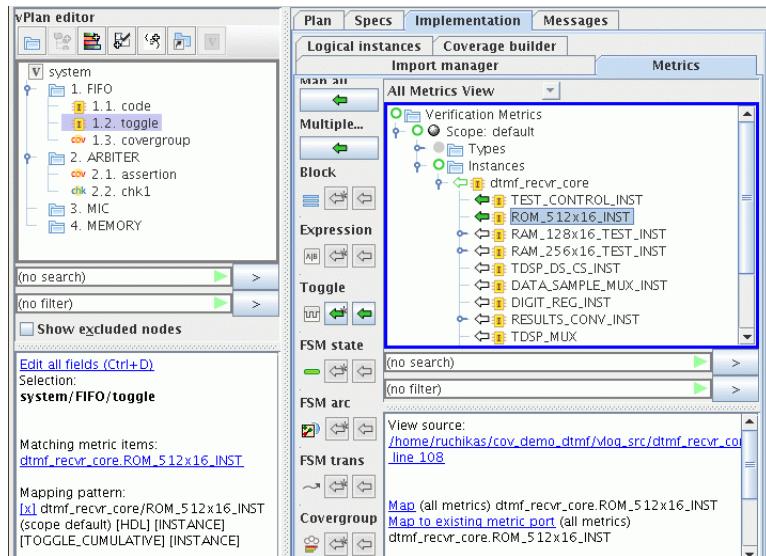
From the pop-up menu, you can select the metric and its grade that must be mapped to the item selected in the vPlan hierarchy.

- Map metric to selected port—This option is similar to the *Map metric* option with the only difference that instead of creating a new metric port, it applies mappings to the port selected in the vPlan hierarchy.

## 5.4.2 Mapping Multiple Coverage Metrics to a Single vPlan Item

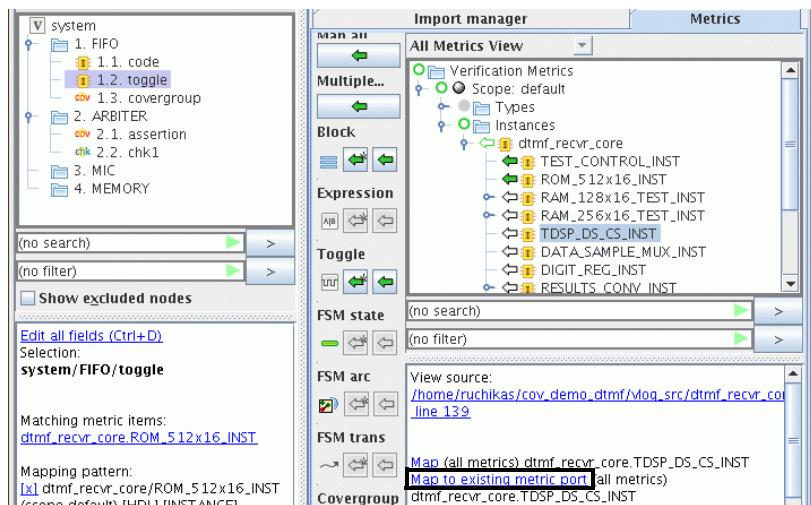
You can also map multiple coverage metrics to a single vPlan item. Consider an example where ROM\_512x16\_INST is mapped to system/FIFO/toggle, as shown in [Figure 5-15 on page 156](#).

**Figure 5-15 Mapping Coverage Metrics**



In case you want to map another coverage metrics, such as TDSP\_DS\_CS\_INST to system/FIFO/toggle, you must select TDSP\_DS\_CS\_INST in the coverage hierarchy and click the *Map to existing metric port* link shown at the bottom of the message pane below the coverage hierarchy tree, as shown in Figure 5-16 on page 156.

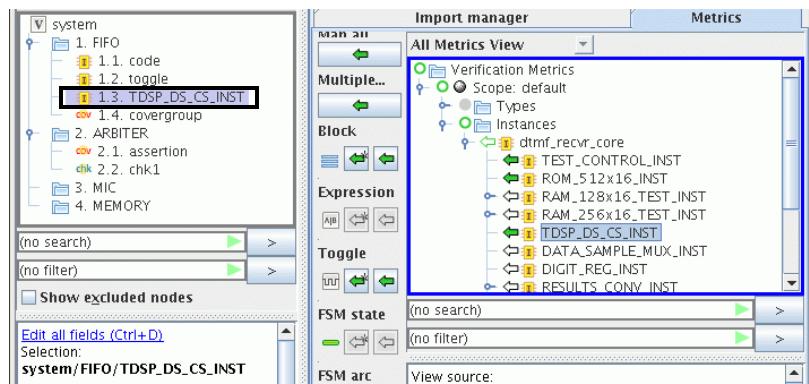
**Figure 5-16 Mapping Coverage Metrics**



This will map the selected coverage item TDSP\_DS\_CS\_INST to system/FIFO/toggle selected in the vPlan tree.

**Note:** There is another link named *Map* just above the *Map to existing metric port* link. If you click *Map*, then a new metric port will be created in the vPlan tree and the selected coverage item will be mapped to the newly created metric port, as shown in [Figure 5-17](#) on page 157.

**Figure 5-17 Mapping Coverage Metrics**



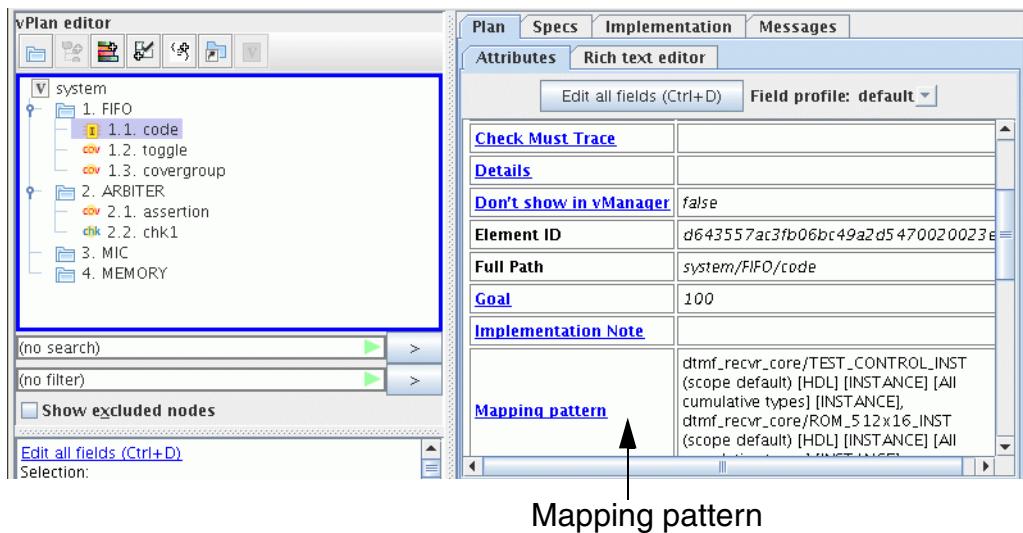
This happens because a coverage metrics was already mapped to the selected vPlan item, `system/FIFO/toggle`.

As a result, if you want to map multiple coverage metrics to a single vPlan item, then for the first mapping, click the *Map* link, and thereafter, for subsequent mappings, click the *Map to existing metric port* link.

### 5.4.3 Mapping Pattern

After you map an item, the mapping pattern is shown in the *Attributes* table on the *Plan* tab page, as shown in [Figure 5-18](#) on page 158.

**Figure 5-18 Mapping Pattern**



The mapping pattern field is populated to show the mapping details. You can modify the mapping pattern, as required and also add more patterns.

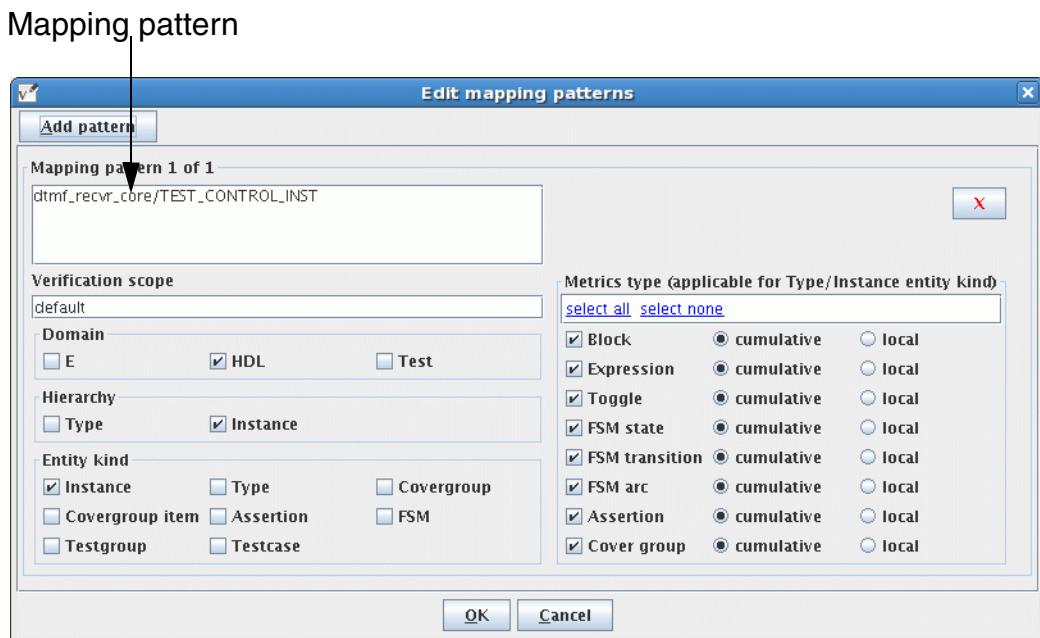
### 5.4.3.1 Modifying an Existing Pattern

To modify an existing pattern:

1. Click the *Mapping patterns* link in the *Attributes* table on the *Plan* tab page.

The *Edit mapping patterns* dialog box appears, as shown in [Figure 5-19](#) on page 159.

**Figure 5-19 Edit Mapping Pattern**



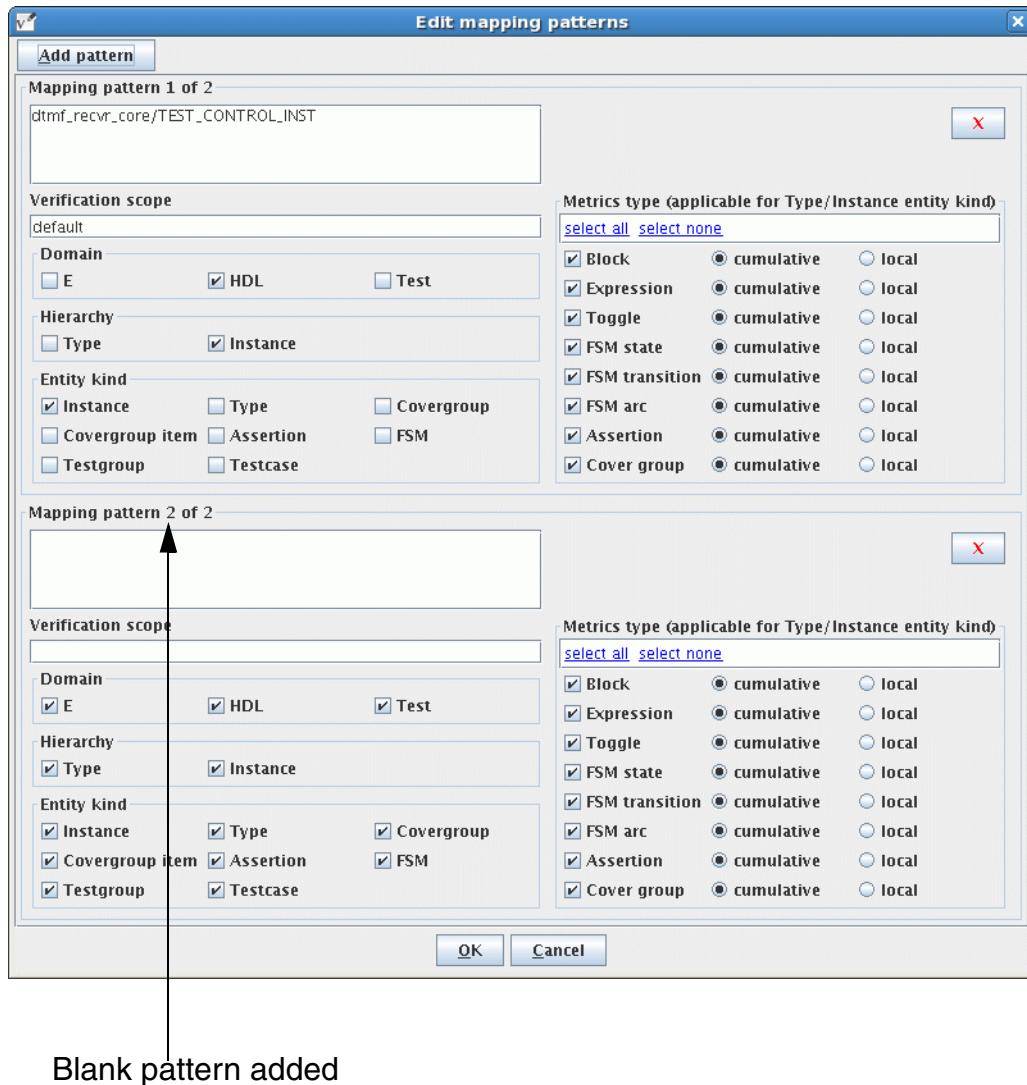
2. In the *Edit mapping patterns* dialog box, you can select or deselect the items such as, domain, hierarchy, entity kind, and metrics type. You can also modify the mapping pattern shown in the text box. Select and deselect the items, as required and click *OK*.

### 5.4.3.2 Adding a New Pattern

To add a new mapping pattern, click the *Add pattern* button in the *Edit mapping patterns* dialog box.

When you click the *Add pattern* button, another pattern is appended in the *Edit mapping patterns* dialog box, as shown in [Figure 5-20](#) on page 160.

**Figure 5-20 Edit Mapping Pattern**



Blank pattern added

Notice that a blank pattern is appended in the dialog box. You can specify the pattern in the text box and also select or deselect associated items based on your requirement.

## Using Wildcards While Specifying Mapping Patterns

When specifying the pattern, you can use wildcards \*, . . . , and ?.

- Use \* to match zero or any number of characters in the absolute or relative path or instance name, or type name.

For example, to map all instances starting with TDSP in the complete instance hierarchy to the selected vPlan element, use:

\* /TDSP\*

- Use ? to match a single character in the instance or type name.

For example, the following pattern will match all instances starting with TDSP\_CORE followed by any single character, such as TDSP\_CORE1, TDSP\_CORE2, and so on.

dtmf\_recv\_core/TDSP\_CORE?

- Use . . . to specify all descendants or ascendants of an instance. If a pattern ends with . . ., it matches all descendants.

For example, to map all descendants of instance TDSP\_CORE\_INST to the selected vPlan element, use:

dtmf\_recv\_core/TDSP\_CORE\_INST/ . . .

If a pattern starts with . . ., it matches all the items whose full path ends with rest of the pattern. For example, the following pattern will match all items whose full path ends with test.

. . . /test

**Note:** A . . . can also be used in the middle of the pattern, for example:

sys/ . . . /test

**Note:** If an instance name includes special characters like [ ], then enclose the instances name within quotes " ". For example, if the instance name is xbus0.masters[0], then specify the pattern as:

"xbus0.masters[0]"

## 5.5 Viewing Mappings

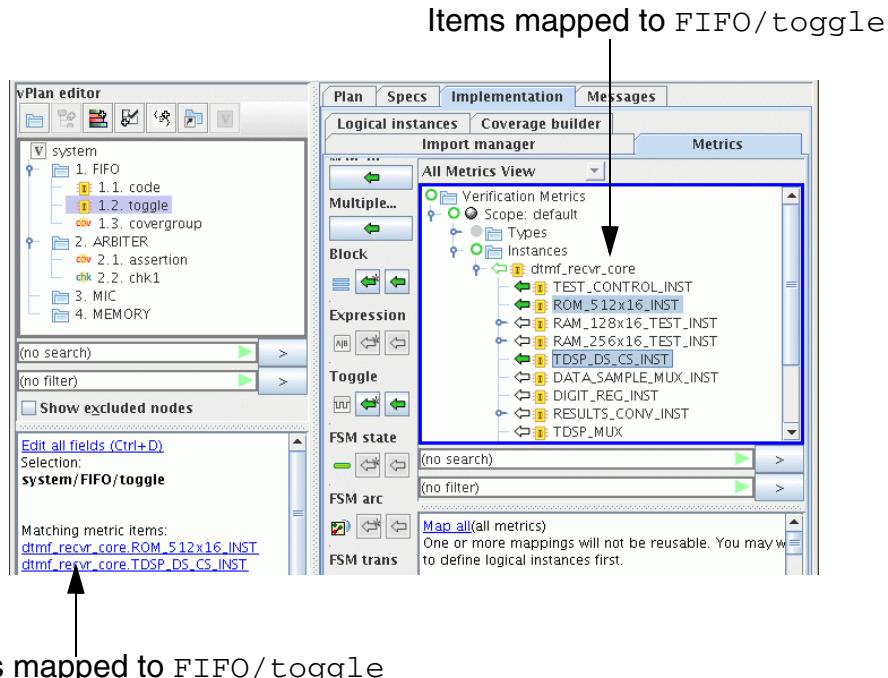
To view the items mapped to a selected vPlan element:

1. Select a mapped port in the vPlan editor.
2. Right-click and select *Show mapped implementation*.

For example, right-click *system1.vplan/FIFO/toggle* in the vPlan editor and select *Show mapped implementation*.

vPlanner highlights all the items mapped to the selected vPlan element in the verification metrics tree, as shown in [Figure 5-21](#) on page 162.

**Figure 5-21 Mapped Items**



The items mapped to the selected vPlan element are highlighted in the *Metrics* tab page.

In addition, the *Message* pane below the vPlan editor also shows the items mapped with a selected vPlan element.

When you click any of the mapped implementation in the *Message* pane, it is selected in the verification hierarchy tree.

## 5.6 Deleting Mappings

To delete the mapping associated with a vPlan element:

1. Select the mapped vPlan element in the vPlan editor.
2. Right-click and select *Unmap implementation*.

This will delete mappings from the selected item.

**Note:** You can also delete mappings by right-clicking the instance or type in the verification hierarchy tree and selecting *Unmap* from the pop-up menu.

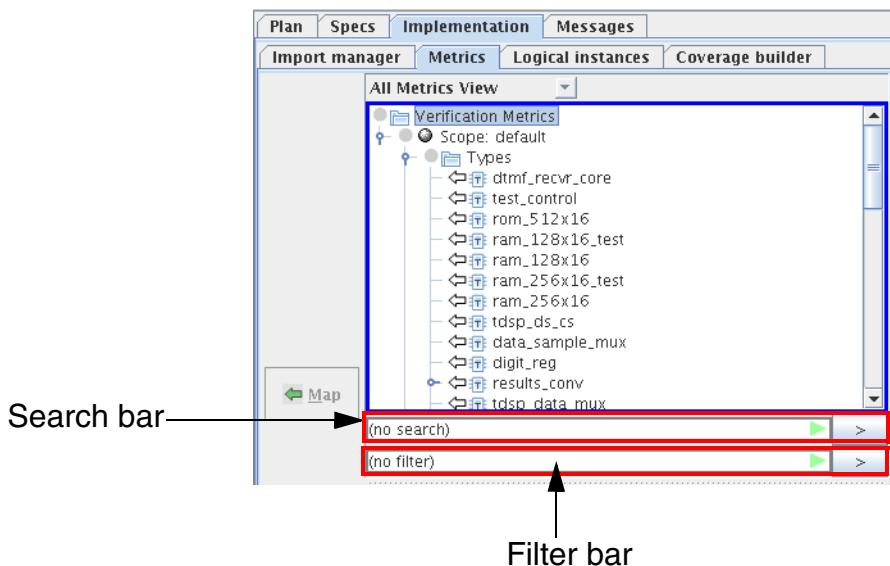


If a vPlan element is mapped using a logical instance, you cannot delete its mapping using the above procedure. For details on deleting mappings that use logical instances, see [Removing the Use of Logical Instance Mapping](#) on page 138.

## 5.7 Searching and Filtering Coverage Items in the Metrics Tree

[Figure 5-22](#) on page 163 displays the search bar and the filter bar available at the bottom of the verification hierarchy tree.

**Figure 5-22 Search and Filter Data**



The search bar allows you specify the search criteria and once you click the green arrow button next to the text box, it highlights the items that meet the search criteria in the verification hierarchy tree.

The filter bar allows you specify the filter criteria and once you click the green arrow button next to the text box, it filters the verification hierarchy tree to show only the items that meet the filter criteria.

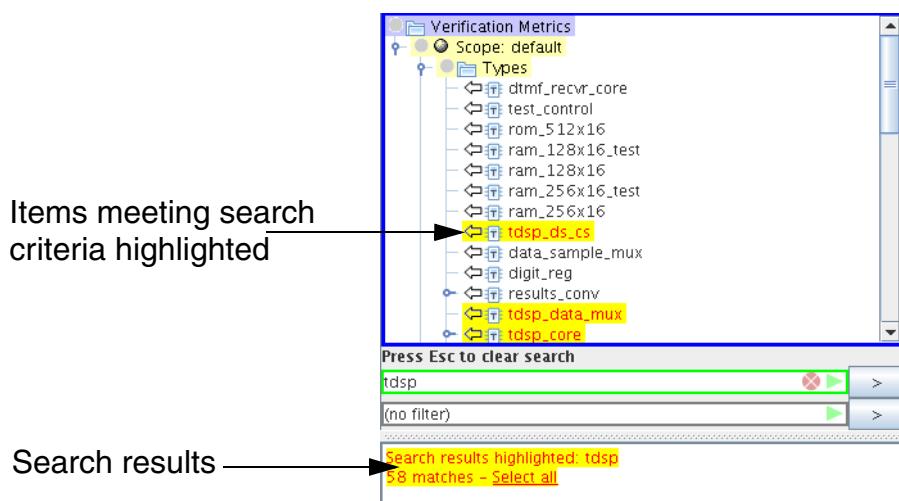
### **Example 1: Search the Verification Hierarchy Tree**

To search the coverage hierarchy for coverage items that include `tdsp` in the name, specify `tdsp` in the search text box and click the green arrow button.

**Note:** Instead of clicking the green arrow button, you can also press the `Enter` key.

Figure 5-23 on page 164 shows the coverage hierarchy with items that meet search criteria highlighted.

**Figure 5-23** Search for Data



In the coverage hierarchy, items that meet the search criteria are highlighted.

To clear the search criteria, click the cross marked red in the search text box. You can also press the `Esc` key to clear the search criteria.

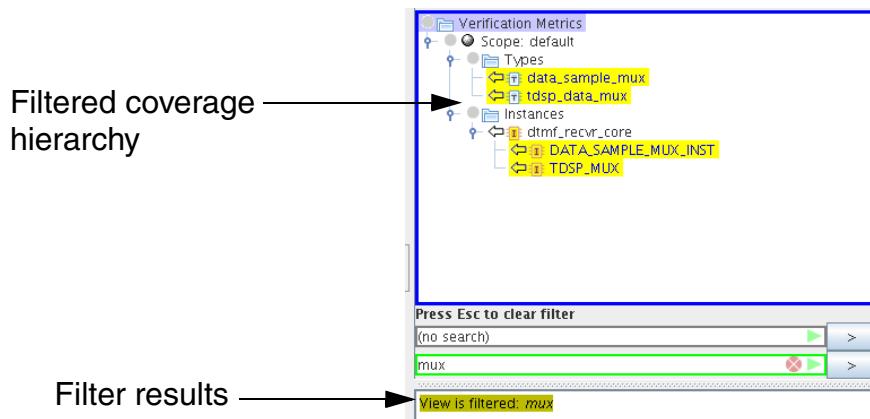
### **Example 2: Filter the Verification Hierarchy Tree**

To filter the coverage hierarchy to display only the coverage items that include `mux` in the name, specify `mux` in the filter text box and click the green arrow button.

**Note:** Instead of clicking the *Filter* button, you can also press the `Enter` key.

Figure 5-24 on page 165 shows the filtered coverage hierarchy.

**Figure 5-24 Filtered Data**



Notice that the coverage hierarchy now displays only the instances and types that include `mux` in their name.

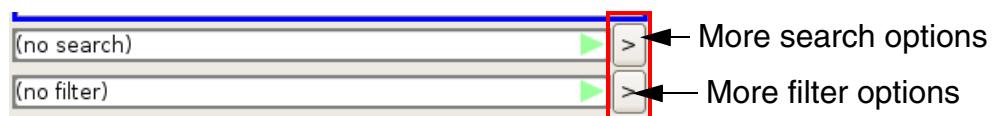
**Note:** The nodes that match the filter criteria are highlighted. In addition, nodes that are shown because a descendant matches are not highlighted.

To clear the filter criteria, click the cross marked red in the filter text box. You can also press the `Esc` key to clear the filter criteria.

### 5.7.1 Advanced Search and Filtering

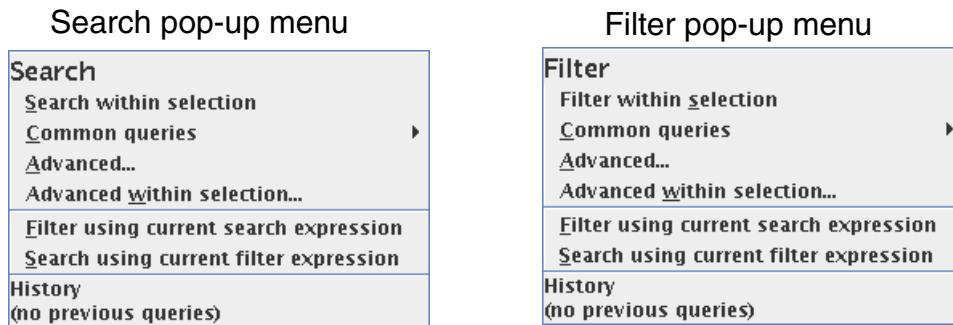
[Figure 5-25 on page 165](#) shows the buttons that allow you to invoke more search and filtering options.

**Figure 5-25 Advanced Search and Filter Buttons**



When you click the more search options or more filter options button, a pop-up menu is displayed, as shown in [Figure 5-26 on page 166](#).

**Figure 5-26 Advanced Search**



You can select any of the following:

- Search within selection—To apply search only within the nodes selected in the verification hierarchy tree.
- Filter within selection—To apply filter only within the nodes selected in the verification hierarchy tree.
- Common queries—To quickly search or filter based on the list of pre-defined queries. See [Table 5-1](#) on page 167 for list of pre-defined queries.
- Advanced—To create your own search or filter query.
- Advanced within selection—To create your own search or filter query and apply that query only to the nodes selected in the verification hierarchy tree.
- Filter using current search expression—To apply filtering based on the expression specified in the search criteria text box.
- Search using current filter expression—To apply search based on the expression specified in the filter criteria text box.
- The History section maintains a history of search or filter queries ran within the current session of Planner. You can quickly run the query by selecting it. It saves time as you do not need to type the query again.

### ***Common queries***

This option provides a list of pre-defined queries that you can use to quickly search the verification hierarchy tree. When you click on *Advanced* —> *Common queries*, a submenu is displayed to show the list of available queries that you can select.

[Table 5-1](#) on page 167 lists the submenu options within *Common queries* and their description.

**Table 5-1**

Common Queries	Description
Testcases	To search or filter the verification hierarchy tree to highlight or show the nodes of type testcase.
Test groups	To search or filter the verification hierarchy tree to highlight or show the nodes of type test group.
All code coverage	To search or filter the verification hierarchy tree to highlight or show the nodes of type block, expression, and toggle.
Block	To search or filter the verification hierarchy tree to highlight or show only the nodes of type block.
Expression	To search or filter the verification hierarchy tree to highlight or show only the nodes of type expression.
Toggle	To search or filter the verification hierarchy tree to highlight or show only the nodes of type toggle.
All FSM coverage	To search or filter the verification hierarchy tree to highlight or show only the nodes of type FSM (state, transition, arc).
State	To search or filter the verification hierarchy tree to highlight or show only the nodes of type state.
Transition	To search or filter the verification hierarchy tree to highlight or show only the nodes of type transition.
Arc	To search or filter the verification hierarchy tree to highlight or show only the nodes of type arc.
Functional coverage (all)	To search or filter the verification hierarchy tree to highlight or show only the nodes that include all functional metrics (cover groups, assertions).
Functional coverage (no assertion)	To search or filter the verification hierarchy tree to highlight or show only the nodes that include all functional metrics except assertions.
Cover groups	To search or filter the verification hierarchy tree to highlight or show only the nodes of type cover group.

Common Queries	Description
Scalar items	To search or filter the verification hierarchy tree to highlight or show only the nodes that include scalar items.
Transition items	To search or filter the verification hierarchy tree to highlight or show only the nodes that include transition items.
Cross items	To search or filter the verification hierarchy tree to show or highlight only the nodes that include cross items.
Assertions	To search or filter the verification hierarchy tree to show or highlight only the nodes that include assertions.
Specman e	To search or filter the verification hierarchy tree to show or highlight only the nodes where underlying source code is written in Specman <b>e</b> .
SystemVerilog	To search or filter the verification hierarchy tree to show or highlight only the nodes where underlying source code is written in SystemVerilog.
VHDL	To search or filter the verification hierarchy tree to show or highlight only the nodes where underlying source code is written in VHDL.
SystemC	To search or filter the verification hierarchy tree to show or highlight only the nodes where underlying source code is written in SystemC.
Mapped	To search or filter the verification hierarchy tree to show or highlight only the mapped items.
Unmapped	To search or filter the verification hierarchy tree to show or highlight only the unmapped items.
Gradable items	To search or filter the verification hierarchy tree to show or highlight only the gradable items.
Items that are not disabled	To search or filter the verification hierarchy tree to show or highlight only the items that are not disabled in the testbench source code using <code>no_collect</code> .
Logical instance	To search or filter the verification hierarchy tree to show or highlight only the instances for which a logical instance is defined.

### Advanced

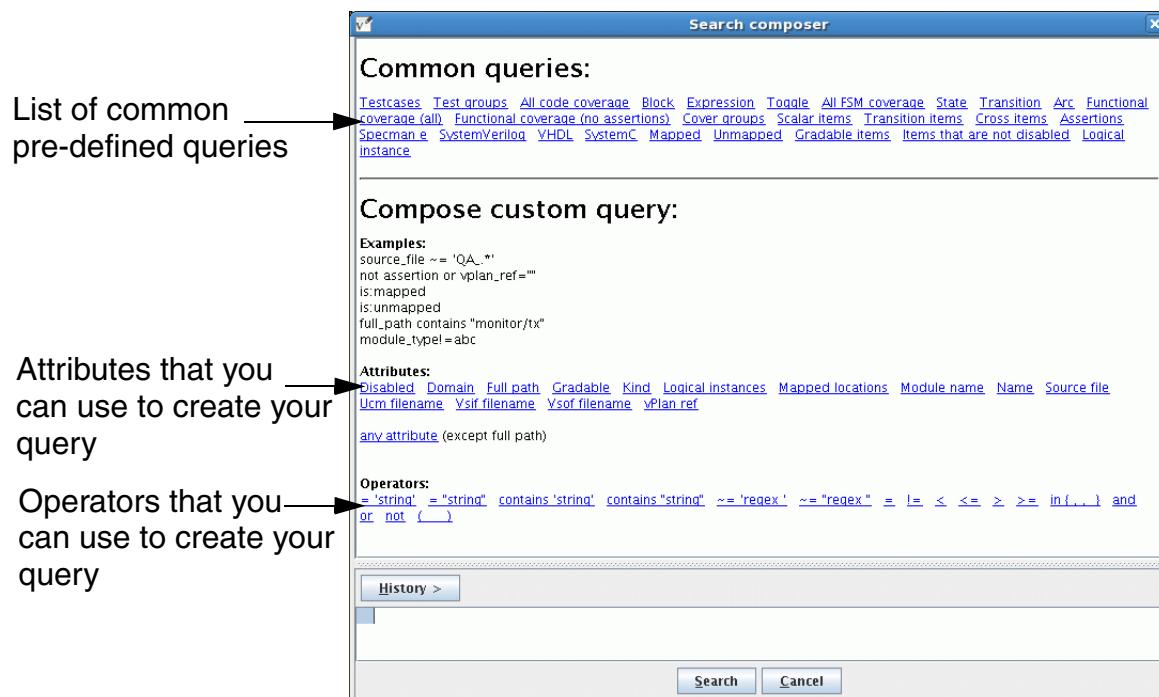
This option allows you to create your own search or filter query. This also helps when you want to specify multiple queries together.

For example, you want to search for types or instances that include cover groups or assertions. For this, you have to create a custom query.

To create a custom query:

1. Click the *More search options* button and select *Advanced* from the pop-up menu.  
A *Search composer* dialog box appears, as shown in [Figure 5-27](#) on page 169.

**Figure 5-27 Search composer**



2. To use a particular pre-defined query, attribute, or operator, click its link and it will be added to the text box shown at the bottom of the *Search composer* dialog box.

For example, to search for types or instances that include cover groups or assertions, click *Cover groups*. As you click the *Cover groups* link, the query starts adding to the text box, as shown in [Figure 5-28](#) on page 170.

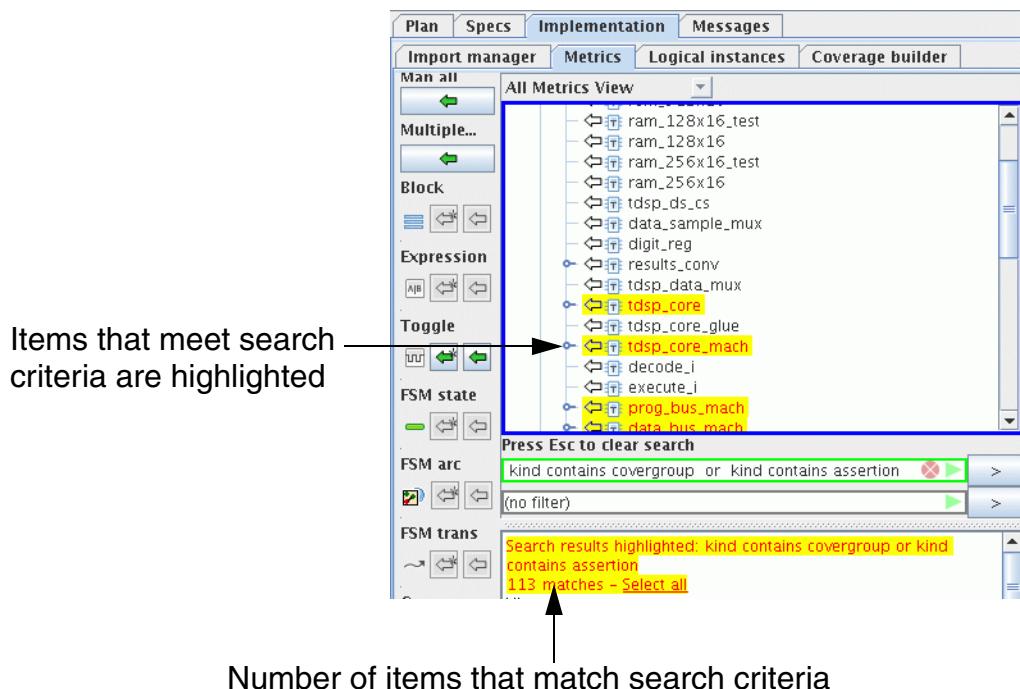
**Figure 5-28 Search composer**



3. Click the required operator. For example, click *or* from the list of operators shown.
4. Complete the query by adding the required attributes or common queries. For example, click *Assertion* from the list of common queries shown in the dialog box.
5. Click *Search* after completing the query.

The Search composer dialog box is closed and the items that match the search criteria are highlighted in the verification hierarchy tree, as shown in [Figure 5-29](#) on page 170.

**Figure 5-29 Search Results**



Similarly, you can create more customized queries.

### **Advanced within selection**

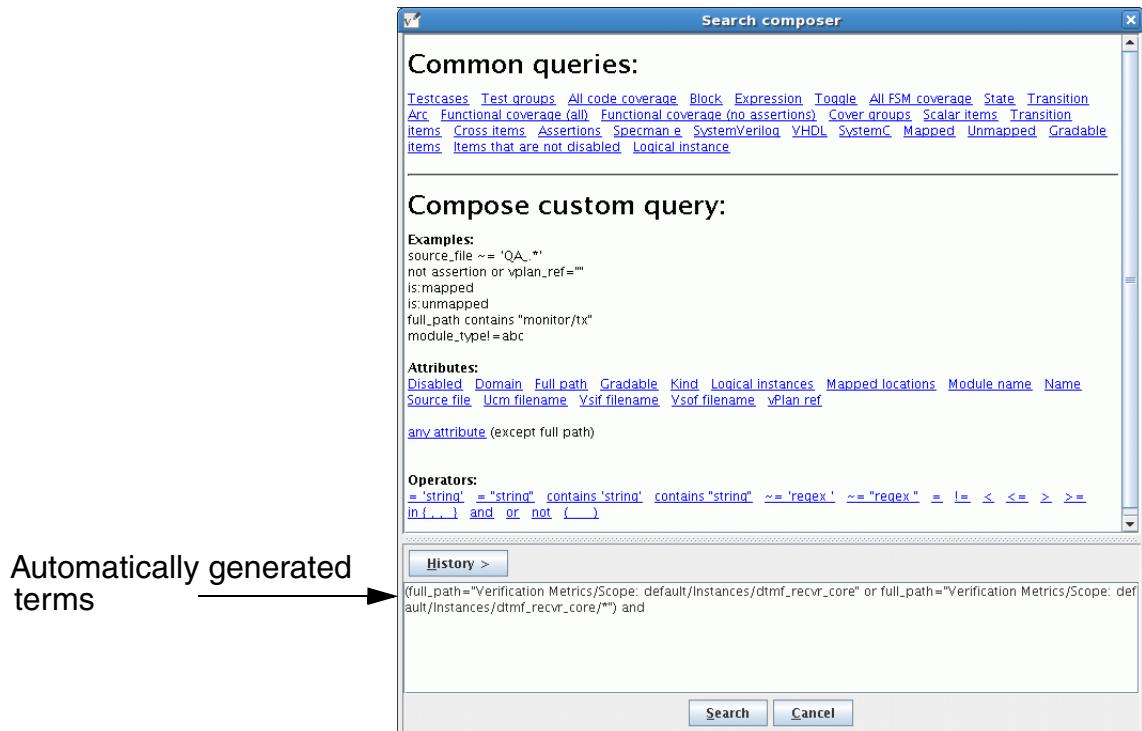
The functionality of *Advanced within selection* is same as *Advanced* with a difference that in this case, the custom query applies only to the nodes that are selected in the verification hierarchy tree.

For example, you want to apply search only within instance `dtsmf_recv_core` and want to search the items that include cover groups. For this:

1. Select `dtsmf_recv_core` in the verification hierarchy tree.
2. Click the *More search options* button and select *Advanced within selection* from the pop-up menu.

A *Search composer* dialog box appears, as shown in [Figure 5-30](#) on page 171.

**Figure 5-30 Search composer (Advanced within selection)**



When you select *Advanced within selection*, the query box in the *Search composer* dialog box shows automatically generated terms. These terms are automatically generated by Planner based on the nodes selected in the verification hierarchy tree.

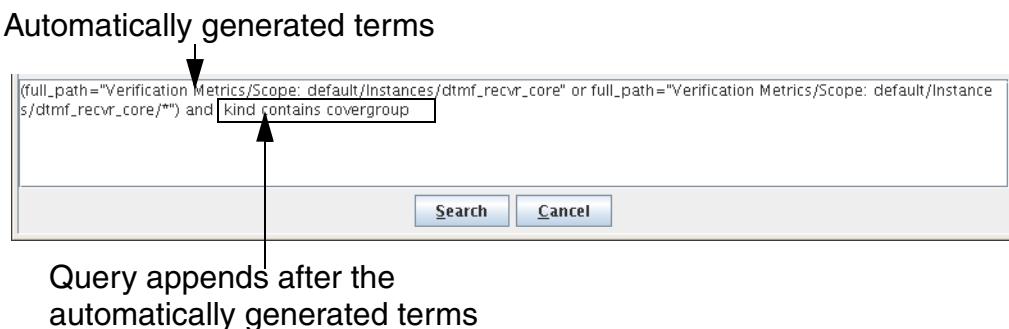
**Note:** You should not modify or tamper the automatically generated terms. Any modification to automatically generated terms will mess up the query and will not

generate the required search or filter results. The query that you create using the Search composer dialog box will be appended after the automatically generated terms.

3. To use a particular pre-defined query, attribute, or operator, click its link and it will be added to the text box shown at the bottom of the *Search composer* dialog box.

For example, to search for instances that include cover groups, click *Cover groups*. As you click the *Cover groups* link, the query starts adding to the text box after the automatically generated terms, as shown in [Figure 5-31](#) on page 172.

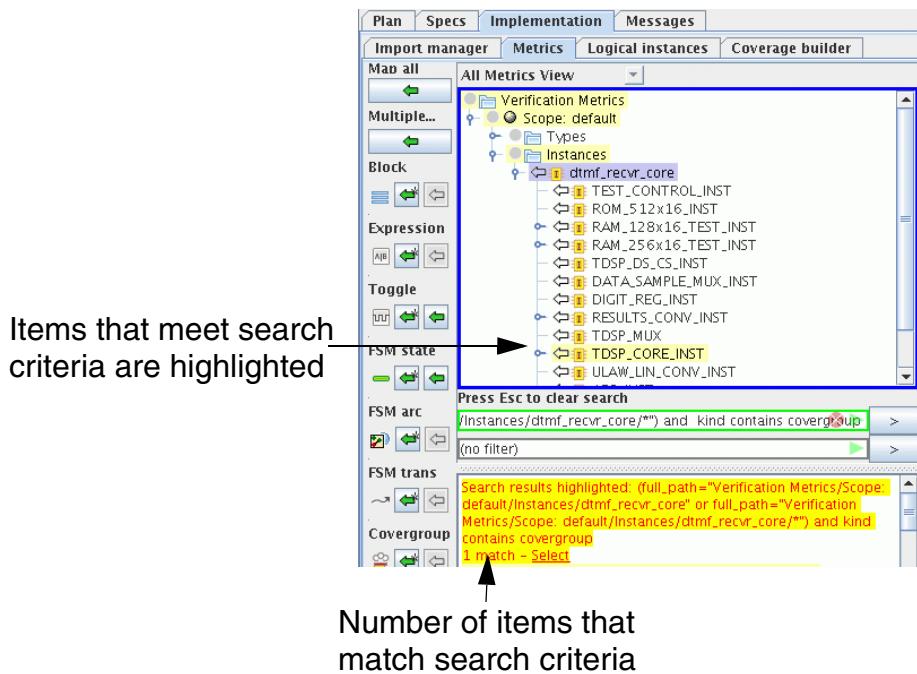
**Figure 5-31** Search composer



4. Click *Search* after completing the query.

The Search composer dialog box is closed and the items that match the search criteria are highlighted in the verification hierarchy tree, as shown in [Figure 5-32](#) on page 173.

**Figure 5-32 Search Results**



Similarly, you can create more customized queries.

## 5.8 Creating Bin Filters

**Note:** The term bucket has been replaced with bin, and particularly bucket filter has been renamed to bin filter.

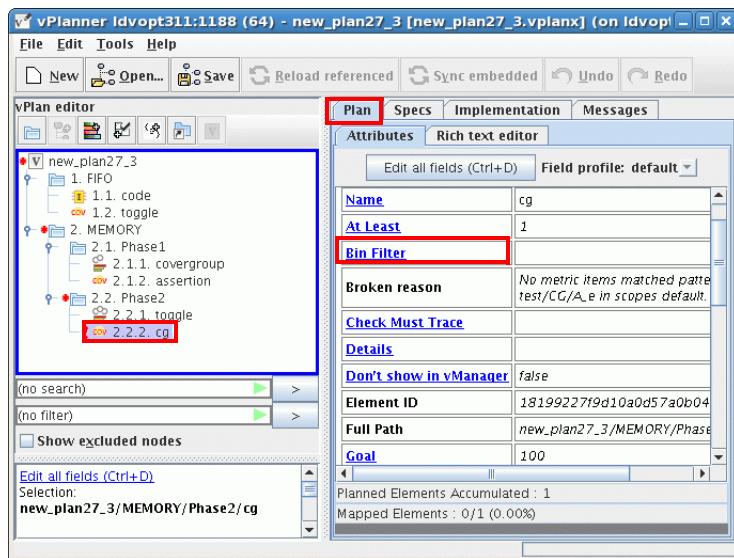
After mapping the coverage items to the vPlan elements, you can specify bin filters so that the specified bins are not included in grade calculations.

**Note:** Bin filters are specified using the *Bin Filter* attribute. This attribute does not affect the way data is displayed in vPlanner. This attribute is used by vManager during vPlan grading.

To specify the bin filters, perform the following steps:

1. Select the vPlan element in the vPlan editor. For example, select `cg` in the vPlan editor.
2. Click the *Plan* tab page in the right pane, as shown in [Figure 5-33](#) on page 174.

**Figure 5-33 Specifying Bin Filter**



- Click the *Bin Filter* attribute.

The *Edit bin filter* dialog box is displayed, as shown in [Figure 5-34](#) on page 174.

**Figure 5-34 Edit Bin Filter**

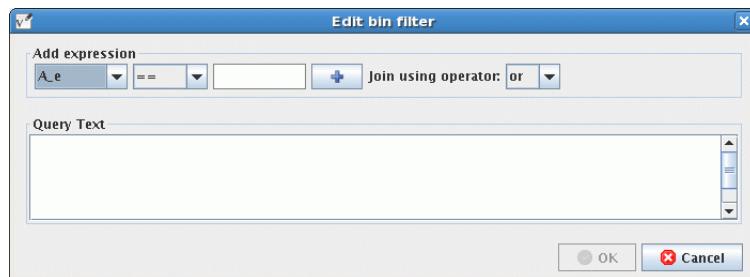


The bin filter is specified as an expression. You can either type the expression manually in the text box or you can build the expression using mapped elements by clicking the *Edit* button. You can also define parameterized filter by clicking the *Add parameterized filter* button. For more details on parameterized filters, see [Creating Parameterized Filters](#) on page 176.

- To build the expression using mapped elements click *Edit*.

The *Edit bin filter* dialog box is displayed, as shown in [Figure 5-35](#) on page 175.

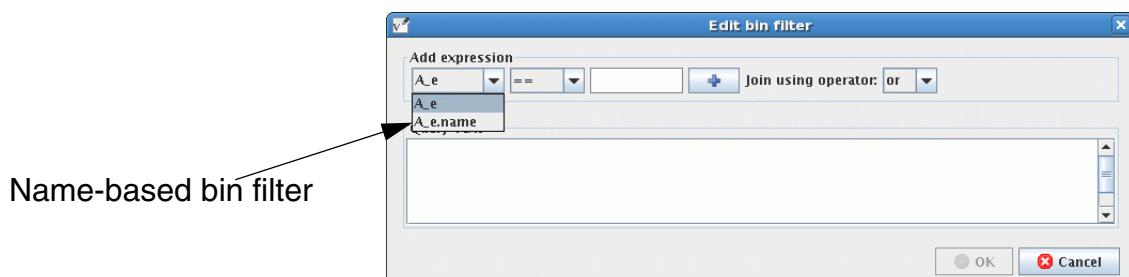
**Figure 5-35 Edit Bin Filter**



5. In the *Add expression* section, you can select the mapped element from the drop-down list to build the query. For example, consider that in vManager, you want vPlan grades to consider only the items where `A_e` is 1. For this, you can build a bin filter expression as `A_e == 1`. In this case, `A_e` is already selected.

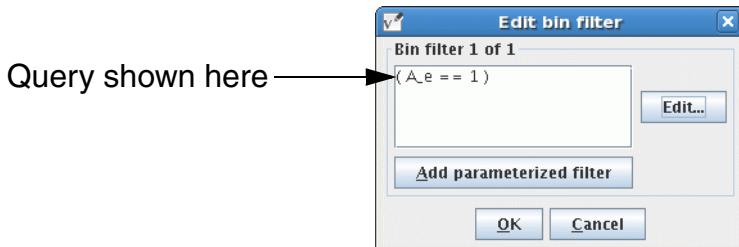
**Note:** In case, you want to apply a bin filter based on name, click on the drop-down list, as shown in [Figure 5-36](#) on page 175 and select `A_e.name`. For each listed mapped item, an additional item `<item>.name` is available in the drop-down list for applying name-based filter.

**Figure 5-36 Edit Bin Filter**



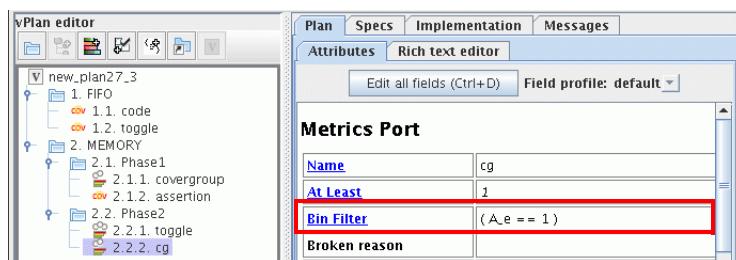
6. Select the relevant operator. In this case, `==` is already selected.
  7. Specify the value of mapped element. For example, specify 1.
  8. Click the `+` button to add the expression to the *Query Text* text box.
- Note:** You can also build an expression using more than one mapped elements. For this, you can select the relevant operator (`or` or `and`) from the *Join using operator* drop-down list.
9. After building the query, click *OK*.
  10. The query is shown in the *Edit bin filter* dialog box, as shown in [Figure 5-37](#) on page 176. Click *OK* to close the dialog box.

**Figure 5-37 Edit Bin Filter**



After you click *OK*, the query shows as the value of the *Bin Filter* attribute, as shown in [Figure 5-38](#) on page 176.

**Figure 5-38 Bin Filter Created**



After the bin filter is added, when you load this verification plan in vManager, the vPlan grades will be calculated based on the specified filter.

### ***Creating Parameterized Filters***

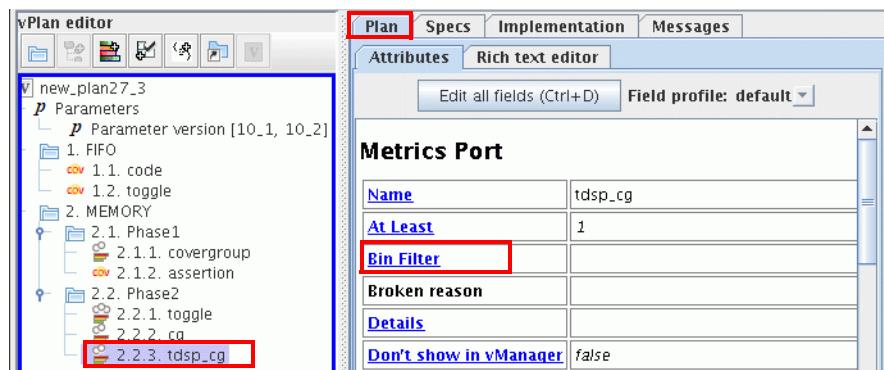
If your verification plan has parameters defined in it, you can define bin filters depending on the parameters value.

For example, consider that your verification plan has a parameter named `version` defined in it and the valid values for this parameter are `10_1` and `10_2`. While defining bin filters, you can use the parameters.

To specify parameterized bin filters, perform the following steps:

1. Ensure that the vPlan has parameters defined in it. For more details on defining parameters, see [Using Parameters](#) on page 127.
2. Select the vPlan element in the vPlan editor. For example, select `tdsp_cg` in the vPlan editor.
3. Click the *Plan* tab page in the right pane, as shown in [Figure 5-39](#) on page 177.

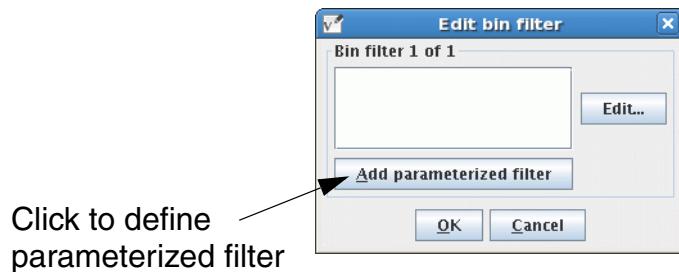
**Figure 5-39 Specifying Parameterized Bin Filter**



- Click the *Bin Filter* attribute.

The *Edit bin filter* dialog box is displayed, as shown in [Figure 5-40](#) on page 177.

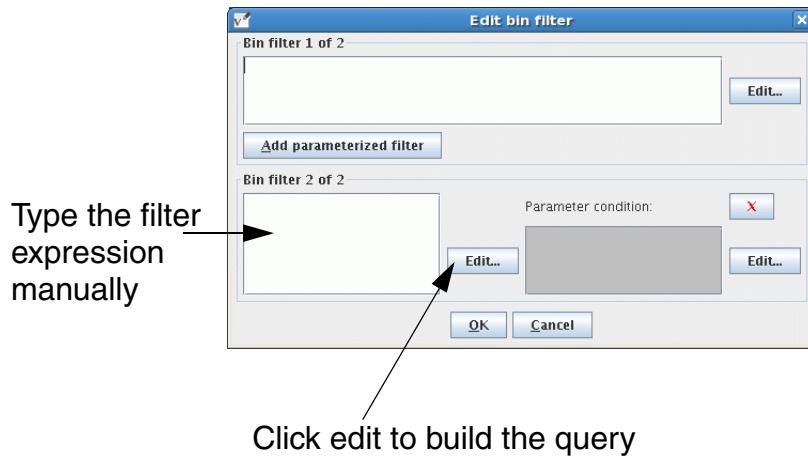
**Figure 5-40 Edit Bin Filter**



- To define parameterized filters, click the *Add parameterized filter* button.

The *Edit bin filter* dialog box is displayed, as shown in [Figure 5-41](#) on page 178.

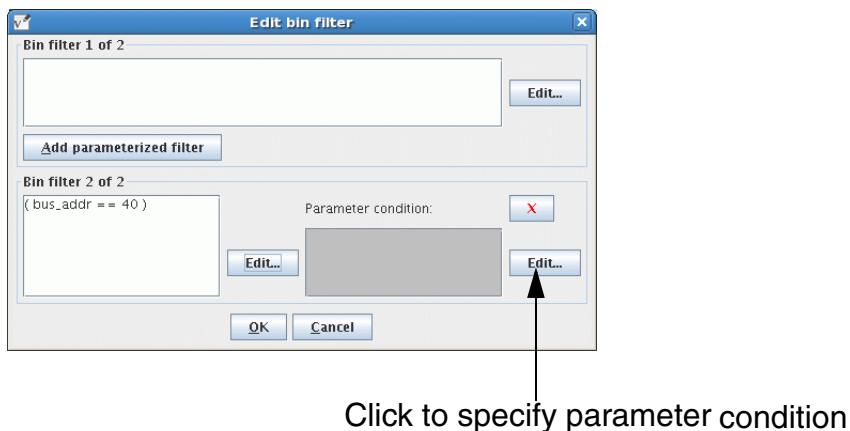
**Figure 5-41 Edit Bin Filter**



6. You can specify the filter expression manually, or click the *Edit* button to build the filter query. For example, specify the query as (`bus_addr == 40`) in the text box.

[Figure 5-42](#) on page 178 shows the *Edit bin filter* dialog box with the query.

**Figure 5-42 Edit Bin Filter**



7. Click the *Edit* button next to the *Parameter condition* section to add the parameter condition.

The *Bin filter parameter condition* dialog box is displayed, as shown in [Figure 5-43](#) on page 179.

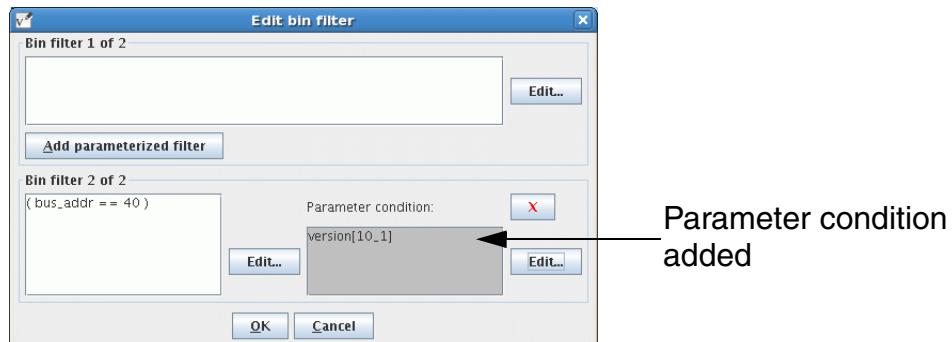
**Figure 5-43 Bin Filter Parameter Condition**



8. By default, all values are shown as selected. Deselect the values you do not want to associate with the filter and click *OK*. For example, deselect *10\_2* and click *OK*.

The parameter condition is added, as shown in [Figure 5-44](#) on page 179.

**Figure 5-44 Bin Filter Parameter Condition**



9. Click *OK* to close the dialog box.

After you click *OK*, the query shows as the value of the *Bin Filter* attribute, as shown in [Figure 5-45](#) on page 179.

**Figure 5-45 Parameterized Bin Filter Created**



Similarly, you can create more parameterized bin filters.

For more details in defining parameters and using them, see [Using Parameters](#) on page 127.

## 5.9 Using the Functional Coverage Builder

The Functional Coverage Builder helps you design scalar, cross, and transition coverage items in **e**. It lets you customize the coverage item bins as well as other coverage item options such as weight, at least, and so on.

The Functional Coverage Builder automatically creates **e** code based on the coverage metric you designed. You can export the functional coverage **e** code and load it on top of the testbench (verification environment). After you export the functional coverage **e** code, you can map it to vPlan elements, as required.

This section covers the following topics:

- [Importing a Testbench](#)
- [Creating a Scalar Functional Coverage Item](#)
- [Customizing a Functional Coverage Item](#)
- [Designing a Cross or Transition Item](#)
- [Designing a Per Instance Item](#)
- [Writing e code and Regenerating Coverage Model](#)

### 5.9.1 Importing a Testbench

If you are adding coverage to an existing **e** testbench, then you can import it to facilitate the implementation of the functional coverage model. Importing the testbench allows you to view and leverage the types, fields, events, and coverage model that you have already defined in the testbench.

If you do not have an existing **e** testbench, you can skip this step and proceed to [Creating a Scalar Functional Coverage Item](#) on page 182.

#### ***Importing e testbench code with no compiled code***

To import **e** testbench code with no compiled code:

1. Make sure the environment generates successfully in Specman before trying to import it into vPlanner.
2. If your environment includes **eVCs** or other components that define empty enumerated types, you must extend the empty types to include at least one name or name-value pair.

One way to do this is to create a file called `my_config.e` that extends any empty types and imports the top-level files of the eVC and the testbench. For example:

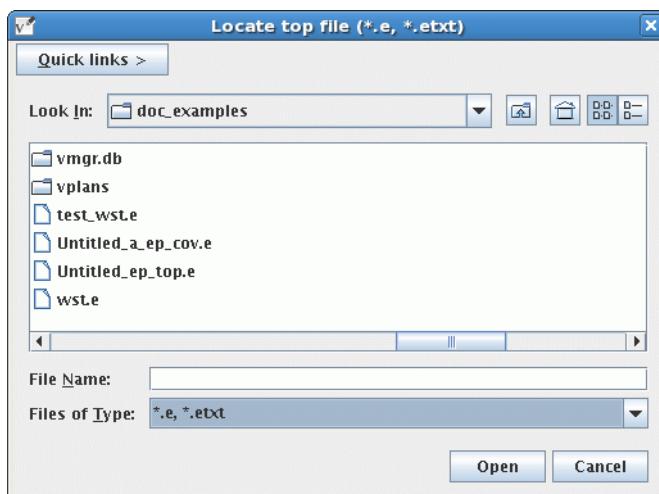
```
extend my_empty_type: [ any_name ];
import my_evc_top.e;
import my_testbench_top.e;
```

Then load `my_config.e` into Specman and generate the environment. If it generates successfully, use `my_config.e` to import the environment into vPlanner.

3. Open the *Implementation* tab.
4. Select *Coverage builder* tab page and click the *Load testbench* button.

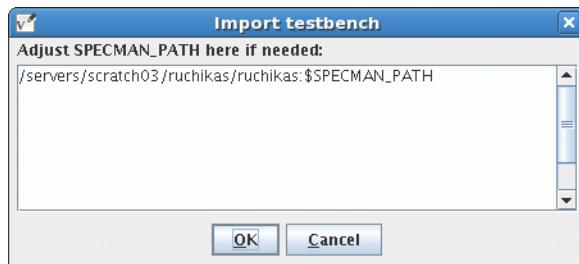
The *Locate top file (\*.e)* dialog box opens, as shown in [Figure 5-46](#) on page 181. It displays a list of **e** files.

**Figure 5-46 Locate Top File**



5. Make sure that `$VMANAGER_PATH` is set appropriately. Select the **e** file that you want to import and click *Open*.
6. The *Import testbench* dialog box opens, as shown in [Figure 5-47](#) on page 182. You can adjust the `SPECMAN_PATH`, if required. Click *OK* to confirm your selection.

**Figure 5-47 Import Testbench**



The Messages tab page opens. If no errors occur, the Messages tab page closes after the types, fields, events, and coverage model defined in the testbench have been loaded.

### ***Importing e testbench code that contains compiled C or e code***

To import **e** testbench code that contains compiled C or **e** code:

1. Exit out of vPlanner.
2. From the command prompt, set the `SPECMAN_EXEC` environment variable such that it points to the compiled Specman executable. For example:

```
setenv SPECMAN_EXEC=<path to compiled specman>
```

3. Launch vPlanner:

```
% vmanager -planner &
```

or

```
% vplanner
```

4. Follow the steps for importing **e** testbench code. For more details, see [Importing e testbench code with no compiled code](#).

### **5.9.2 Creating a Scalar Functional Coverage Item**

To design a scalar coverage item:

1. From the *Implementation* tab, select *Coverage builder*.
2. Click the *Build new coverage item* button.

The *New coverage* dialog opens, as shown in [Figure 5-48](#) on page 183.

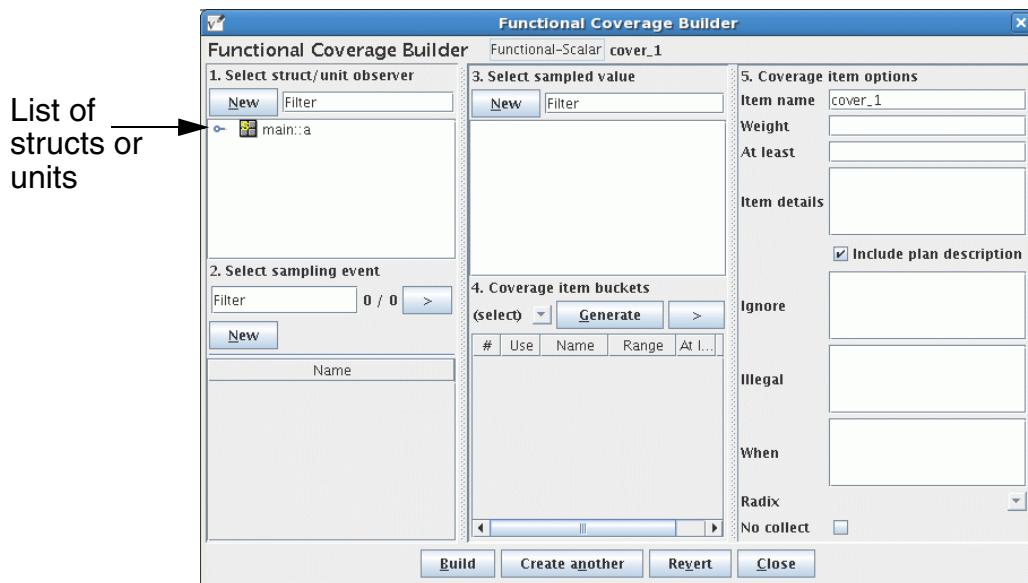
**Figure 5-48 New Coverage**



3. Enter a name for the coverage item. For example, specify *cover\_1* as the name.
4. Choose *Functional Scalar* from the drop-down menu. By default, it is already selected.
5. Click *OK*.

The *Functional Coverage Builder* displays, as shown in [Figure 5-49](#) on page 183.

**Figure 5-49 Functional Coverage Builder (Select struct/unit)**

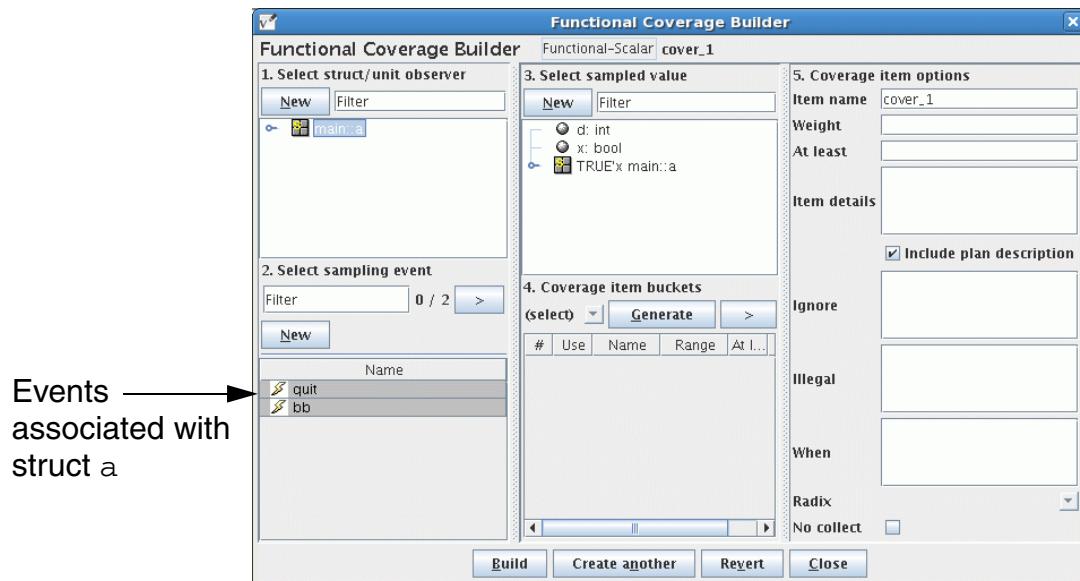


6. Specify a struct/unit as the observer. If you imported an e testbench, a list of structs and unit types is displayed. You can select from an existing struct or unit, or click *New* to create a new struct or unit. To quickly search for a struct or unit in the list of available structs or units, specify a match pattern in the *Filter* field. In this case, select struct *a*.

Typically, coverage is observed in some sort of monitor.

Once you select a struct or a unit, associated events start showing in the events list box, as shown in [Figure 5-50](#) on page 184.

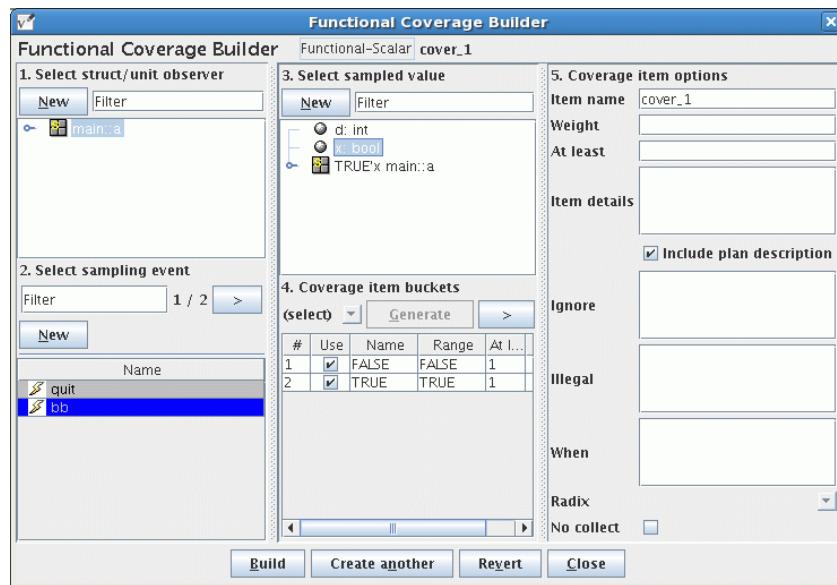
**Figure 5-50 Functional Coverage Builder (Select sampling event)**



7. You can either select a sampling event for the coverage metric from the listed events, or click *New* to add a new sampling event. To quickly search for a specific event from the listed events, type a match pattern in the *Filter* field. For example, in this case, select *bb*.
8. Select the sampled value (field) you want to cover. You can either select any of the listed fields, or click *New* to define a new field or expression. For example, in this case, select *x: bool*.

Once you select the field or expression, the *Coverage Item buckets* pane shows you the default coverage bins based on the type for the field/expression, as shown in [Figure 5-51](#) on page 185.

**Figure 5-51 Functional Coverage Builder (Select sampling event)**



9. (Optional) Customize the buckets/bins and other attributes of the item.

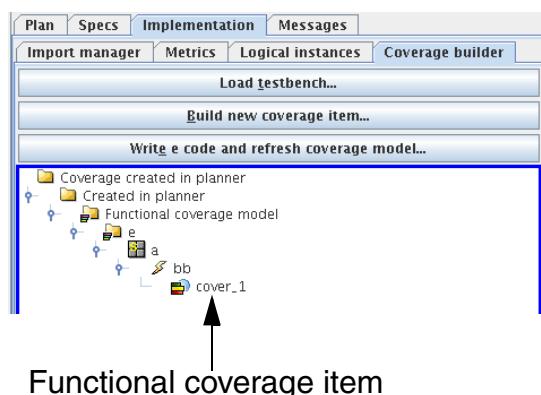
See [Customizing a Functional Coverage Item](#) on page 186 for information on how to do this.

10. Click *Build*.

11. To create another coverage item, click *Create another*. Otherwise, click *Close*.

After you click *Close*, the functional coverage item appears in the *Coverage builder* pane, as shown in [Figure 5-52](#) on page 185.

**Figure 5-52 Functional Coverage Item Created**



The functional coverage item `cover_1` now appears in the *Coverage builder* pane.

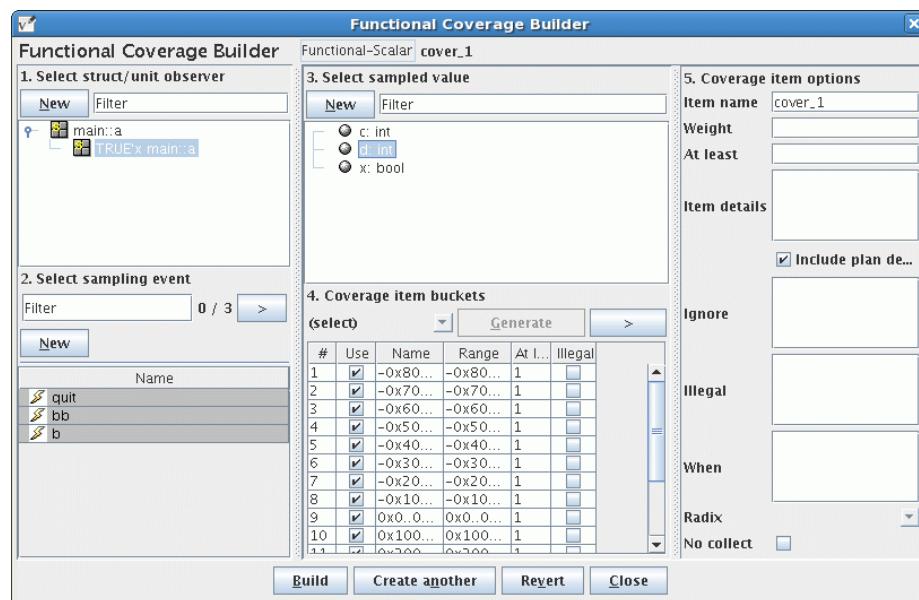
### 5.9.3 Customizing a Functional Coverage Item

To customize a functional coverage item:

1. From the *Implementation* tab, select the *Coverage builder* tab page.
2. Right-click the functional coverage item that you want to customize and select *Edit* from the pop-up menu.

The *Functional Coverage Builder* displays, as shown in [Figure 5-53](#) on page 186.

**Figure 5-53 Functional Coverage Builder**



3. Click the drop-down box under *Coverage item buckets*. It shows following items:

- Default—This is the default selection.
- If the selected item is of `enum` or `boolean` type, then by default one bucket/bin per value is created.
- If the selected item is of `int` or `uint` type, then by default one bucket/bin per value is created if bits are less than equal to 4. Else, 16 equal sized ranges are created.
- Edges—if the field is a signed integer, then when you click the *Generate* button (shown next to the drop-down box), 5 buckets/bins are created, as follows:

- Lowest negative value
- Everything between the lowest negative value and 0
- 0
- Everything between 0 and the highest positive value
- Highest positive value

If the field is an unsigned integer, then when you click the *Generate* button (shown next to the drop-down box), 3 buckets/bins are created, as follows:

- 0
- Everything between 0 and the highest positive value
- Highest positive value
- Custom—This allows you to define the number of buckets/bins to be created. When you select this option and click the *Generate* button, a dialog box is shown where you can specify the number of buckets/bins to be created. Depending on the value, the equal sized buckets/bins are created.
- EdgesCustom—This is same as Edges except that instead of creating one bucket/bin for the everything range, it prompts for a number of equal sized buckets/bins to be created (as is the case with Custom option).

**Note:** For more details on these options, see the tool tip over the *Generate* button.

4. For example, select *Custom* from the drop-down list and click *Generate*.

The *Custom uniform buckets* dialog box is displayed, as shown in [Figure 5-54](#) on page 187.

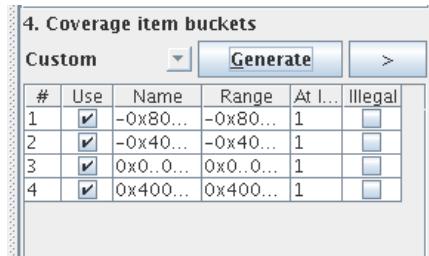
**Figure 5-54 Custom Uniform Buckets/Bins**



5. Specify the number of buckets/bins that you want to create and click *OK*. For example, specify 4 and click *OK*.

Notice that four equal-sized buckets/bins are created to store all values, as shown in [Figure 5-55](#) on page 188.

**Figure 5-55 Uniform Buckets/Bins Created**



The screenshot shows a software interface titled "4. Coverage item buckets". At the top, there is a dropdown menu set to "Custom", a "Generate" button, and a right-pointing arrow button. Below this is a table with the following data:

#	Use	Name	Range	At L...	Illegal
1	<input checked="" type="checkbox"/>	-0x80...	-0x80...	1	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	-0x40...	-0x40...	1	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	0x0..0...	0x0..0...	1	<input type="checkbox"/>
4	<input checked="" type="checkbox"/>	0x400...	0x400...	1	<input type="checkbox"/>

6. (Optional) You can change the name, range, or atleast fields by highlighting it and typing in a new value.
  7. (Optional) You can add, delete, or move buckets/bins by right-clicking on the bucket/bin row and selecting the appropriate option.
  8. (Optional) You can ignore a bucket/bin by deselecting the *Use* check box.
  9. (Optional) You can mark some of the buckets/bins as illegal by selecting the *Illegal* check box.
- Note:** The expressions in the Ignore and Illegal fields are separate from the values entered using the check boxes in the buckets/bins table. The ignore/illegal expressions from both sources are OR'ed together when creating the **e** code.
10. (Optional) You can set other coverage item attributes, such as Description, Goal, Weight, At least, and so on in the far right pane.
  11. Click *Build* to build the item with customizations. Else, click *Revert* to revert the customizations.

The selected functional coverage item is customized based on your selections.

### 5.9.4 Designing a Cross or Transition Item

Once you have designed some scalar items, you can build cross or transition items using the scalar items as long as you select the same observer and sampling event of the designed coverage item.

vPlanner currently supports building cross items only when both scalar items were built within vPlanner. There is no way to build cross items implemented in **e** code with items designed in vPlanner.

To design a cross or transition coverage item:

1. From the *Implementation* tab, select the *Coverage builder* tab page.

2. Click the *Build new coverage item* button.

The *New coverage* dialog box opens.

3. Specify a name for the coverage item.
4. Choose *Functional - Cross* or *Functional - Transition* from the drop-down menu, as shown in [Figure 5-56](#) on page 189. For example, select *Functional - Cross*.

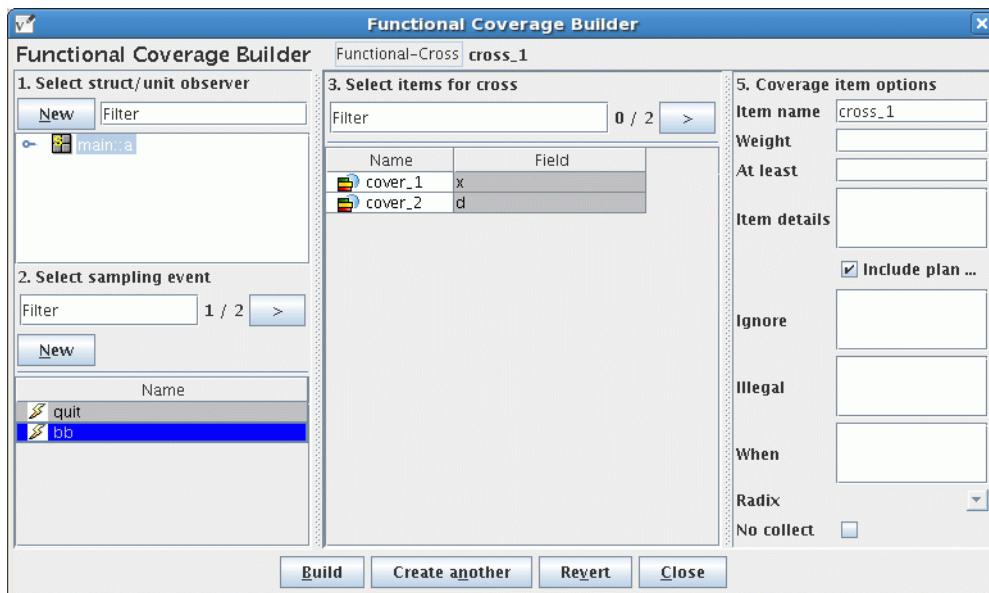
**Figure 5-56 Functional Coverage Item Created**



5. Click *OK*.

The *Functional Coverage Builder* displays, as shown in [Figure 5-57](#) on page 189.

**Figure 5-57 Functional Coverage Builder**

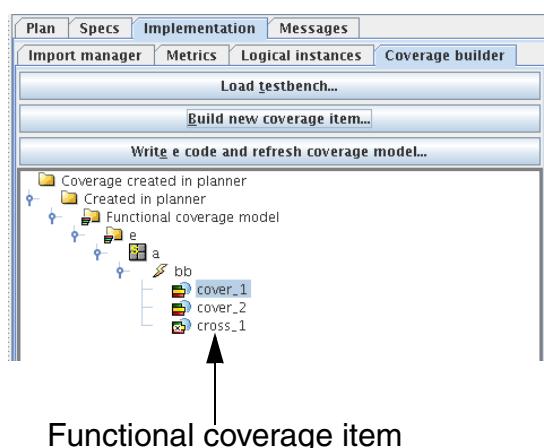


6. Specify a struct/unit as the observer. You can select from an existing struct or unit, or click *New* to create a new struct or unit. To quickly search for a struct or unit in the list of available structs or units, specify a match pattern in the *Filter* field.

7. Select the sampling event for the coverage metric, or create a new event by clicking the **New** button.
8. A list of implemented scalar items appears in the middle pane of the *Functional Coverage Builder*. Select the items you want to include in the cross, using **Ctrl-click** or **Shift-click** to select multiple items for a cross.
9. Click **Build**.
10. To create another coverage item, click *Create another*. Otherwise, click *Close*.

After you click *Close*, the functional coverage item appears in the *Coverage builder* pane, as shown in [Figure 5-58](#) on page 190.

**Figure 5-58 Functional Coverage Item Created**



The functional coverage item `cross_1` now appears in the *Coverage builder* pane.

### 5.9.5 Designing a Per Instance Item

There are two types of per-instance coverage:

- *Group per-instance coverage* (defined with the `per_unit_instance` coverage option) lets you collect separate coverage information for each instance of a unit. You can identify the unit instances in one of two ways:
  - Using logical names returned a string field or method.
  - Using the **e** pathname.

- *Item per-instance coverage* (defined with the `per_instance` coverage option) lets you collect coverage information grouped by a particular value or range of values for a specified cover item.

### 5.9.5.1 Defining Group Per-instance Coverage

To define group per-instance coverage:

1. Implement a scalar, cross, or transition item, as described above.
2. Right-click on the implemented item and select *Manage instances*.

The *Manage instances* dialog box appears, as shown in [Figure 5-59](#) on page 191.

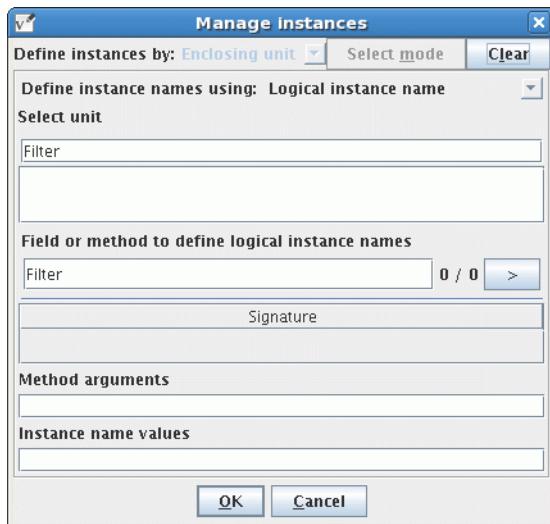
**Figure 5-59 Manage Instances**



3. Select *Enclosing unit* from the pull-down (it is selected by default) and click *Select mode*.

The *Manage instances* dialog box refreshes, as shown in [Figure 5-60](#) on page 192.

**Figure 5-60 Manage Instances**



4. Select Logical instance names or `e` pathname from the pull-down.
5. Select a unit from the list.
6. If you selected Logical instance names, enter the field name that holds the instance name or enter the signature of the method that returns the instance name.
7. (Optional) Enter the legal values for the instance name and click *OK*.

If you do not specify any instance identifiers, you can decide the names later and hand-edit the `e` code. Whether or not you specify any instance identifiers, there is an extra item in the tree like (myfield==\*) so you can map the sum of all of the per-instance results into the plan, rather than just measuring the per-type coverage. This is useful if you want hits for all buckets/bins under all possible conditions.

### 5.9.5.2 Defining Item Per-instance Coverage

To define item per-instance coverage:

1. Implement a scalar item, as described above.
2. Right-click on the implemented item and select *Manage instances*.

The *Manage instances* dialog box appears, as shown in [Figure 5-61](#) on page 193.

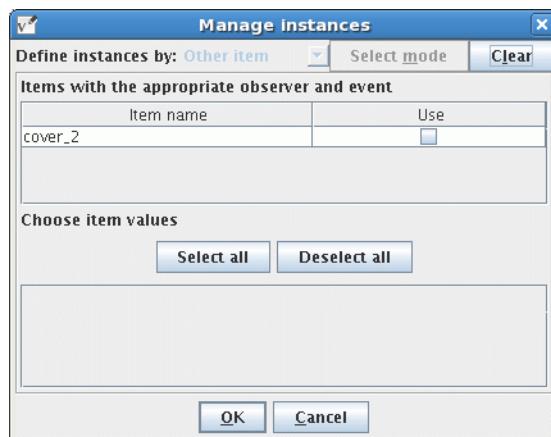
**Figure 5-61 Manage Instances**



3. Select *Other item* from the pull-down and click *Select mode*.

The *Manage instances* dialog box refreshes, as shown in [Figure 5-62](#) on page 193.

**Figure 5-62 Manage Instances**



4. Select one or more items from the list.
5. (Optional) For each item you selected, select the values that you want to create instances for and click *OK*.

If you do not specify values for the items, you can hand-edit the `e` code later. Whether or not you specify any values, there is an extra item in the tree like `(myfield==*)` so you can map the sum of all of the per-instance results into the plan.

This functionality lets you use multiple items for `per_instance` simultaneously.

### 5.9.6 Writing e code and Regenerating Coverage Model

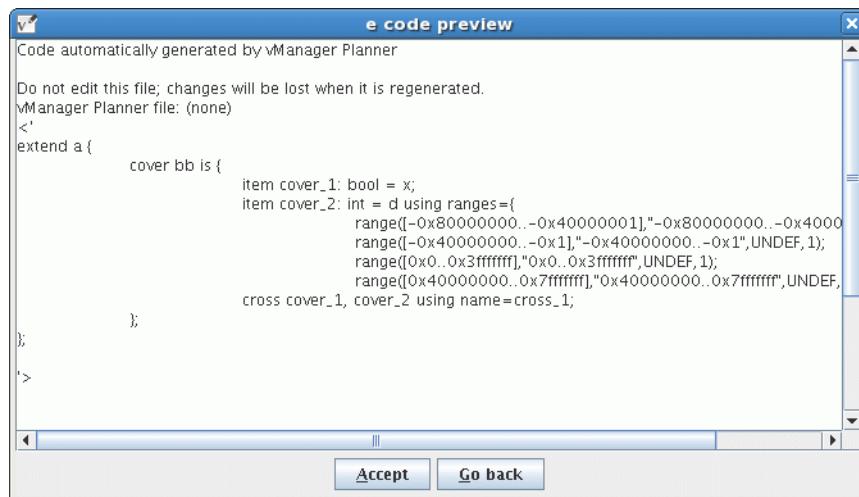
After loading the testbench data and designing scalar, cross, and transition items, you can generate **e** code for the designed items so that later you can load it on the top of the verification environment.

To generate **e** code for the designed items:

1. From the *Implementation* tab, select *Coverage builder*.
2. Click the *Write e code and refresh coverage model* button.

The *e code preview* dialog box opens, as shown in [Figure 5-63](#) on page 194.

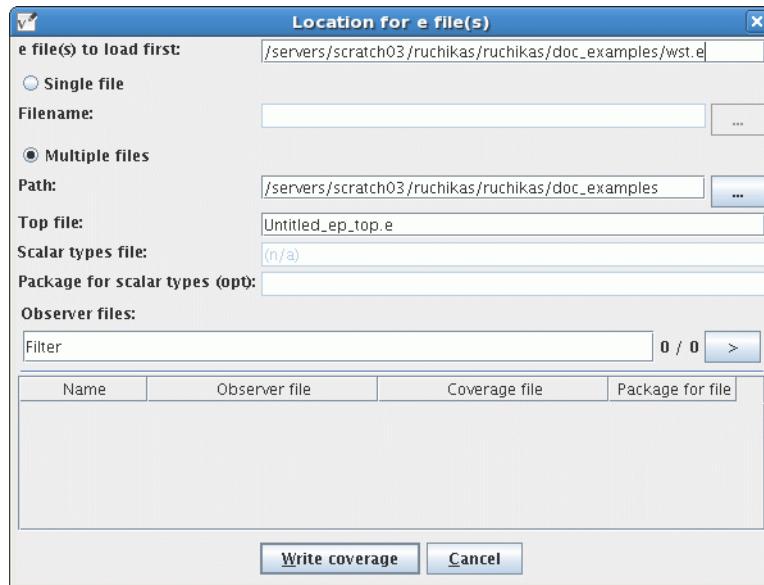
**Figure 5-63 e Code Preview**



3. Click either *Accept* or *Go Back*.

If you click *Accept*, a dialog box opens, as shown in [Figure 5-64](#) on page 195.

**Figure 5-64 Location for e File**



4. You can either write the code to a single file or to multiple files. By default, *Multiple files* is selected.

If you choose to write to multiple files, by default, any new scalar types you have defined are written to a separate file, and a top file is created to import all of your created files. Also, a separate file is created for each observer struct/unit where you designed functional coverage.

5. Specify the path where you want to generate the *e* code.

6. (Optional) Specify the filenames.

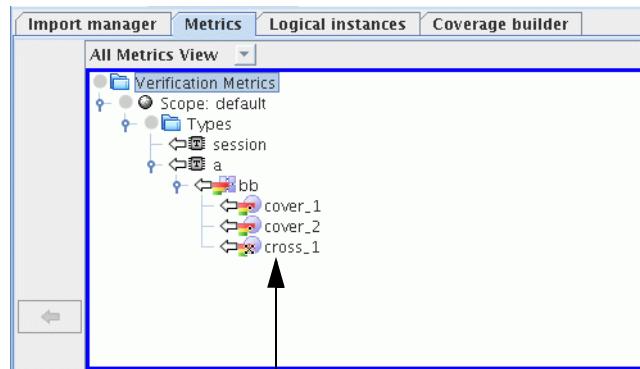
If you want, you can make some of the filenames the same. The *e* code is written to the files as specified.

7. Click the *Write coverage* button.

The *e* code is generated and saved at the specified location. In addition, the coverage model is refreshed to show the new functional coverage items you created.

[Figure 5-65](#) on page 196 shows the *Metrics* tab page, which shows the new functional coverage items you created.

**Figure 5-65 Functional Coverage Items**



Functional coverage items created using Functional Coverage Builder

The items created using the *Functional Coverage Builder* show on the *Metrics* tab page. You can map these items to any of the vPlan items, as required.

---

## Exporting and Publishing the Plan

---

After creating and updating the verification plan, you can export it to an HTML, CSV, Microsoft Word format, or as a Perl script and then share it with the team or publish it on the internet.

Exporting a plan is useful for users who want to make the verification plan available for viewing to people who might not have vPlanner installed with them.

The exported version of the plan cannot be loaded again in the vPlanner. As a result, if you make modifications to the exported version of the plan, they will not be available in vPlanner.

This chapter discusses:

- [Exporting a Verification Plan](#)
- [Viewing the Exported Plan](#)

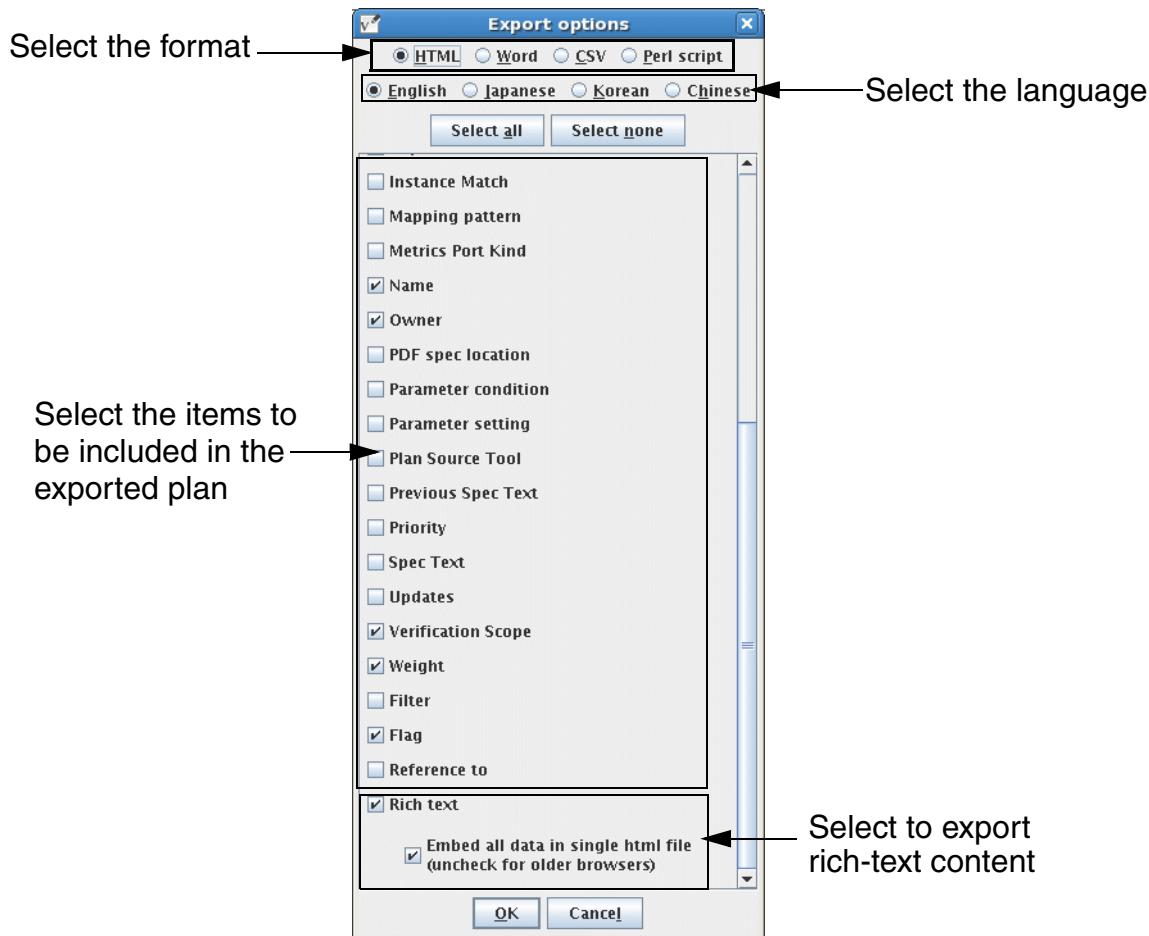
### 6.1 Exporting a Verification Plan

To export a plan:

1. Select *File* —> *Export*.

The *Export options* dialog box is displayed, as shown in [Figure 6-1](#) on page 198.

**Figure 6-1 Export Options**



In the *Export options* dialog box, you can specify:

- ❑ Format in which you want to export the plan. It can be HTML, Word, CSV, or Perl script.
- ❑ Language in which you want to export the plan. It can be English, Japanese, Korean, or Chinese.
- ❑ Items that must be included in the exported plan.

**Note:** You must select the *Embed all data in single HTML file* check box if you want to embed all rich-text data (both text and images) in a single HTML file. For more details, see [Using the Rich-text Editor](#) on page 113.

2. Specify the format in which the plan must be exported. By default, *HTML* is selected.
3. Specify the language. By default, *English* is selected.

4. Specify the items that must be included in the exported plan. For this, select the corresponding check boxes.
5. Click *OK*.
6. A dialog box appears. Browse to the directory where you want to save the exported plan.
7. Enter a name for the plan in the *File Name* field and click *Save*.

For example, to save the exported plan as `system_html`, enter `system_html` in the *File Name* field and click *Save*.

The exported plan is stored at the specified location. You can now share it with the team and also publish it, as required.

**Note:** At the time of exporting the plan, if you select the format as Word, a message box is displayed, as shown in [Figure 6-2](#) on page 199.

**Figure 6-2 Exporting Plan in Word Format**



The text in the message box indicates that vPlanner creates a Visual Basic script (`.vbs` file) that you can run on your Windows machine. This script launches Microsoft Word and programmatically generates a Word document with the vPlan content.

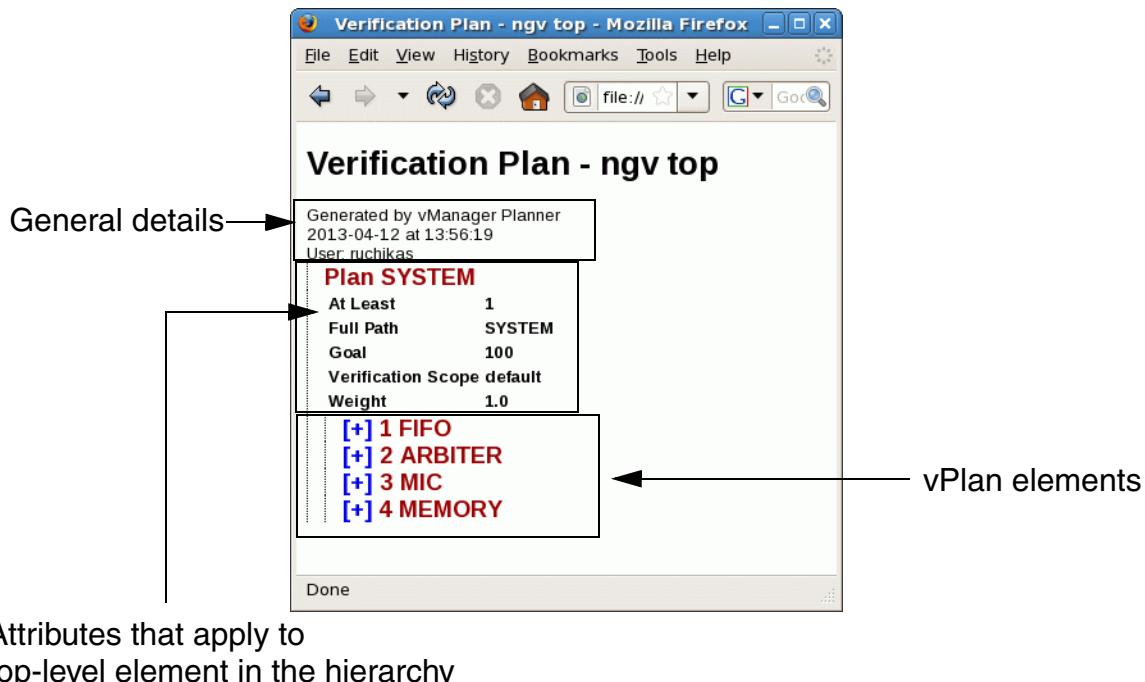
## 6.2 Viewing the Exported Plan

After exporting the plan, you can open it by double-clicking on it.

### 6.2.1 Sample of Exported Plan (HTML Format)

[Figure 6-3](#) on page 200 shows a sample of verification plan in an HTML format.

**Figure 6-3 Plan in HTML Format**



You can expand the verification plan hierarchy by clicking the + sign next to the vPlan element.

[Figure 6-4](#) on page 201 shows a the vPlan element FIFO expanded.

**Figure 6-4 Plan in HTML Format**

Attributes associated with FIFO

Generated by vManager Planner  
2013-04-12 at 13:56:19  
User: ruchikas

**Plan SYSTEM**

At Least	1
Full Path	SYSTEM
Goal	100
Verification Scope	default
Weight	1.0

**[–] 1 FIFO**

At Least	1
Full Path	SYSTEM/FIFO
Goal	100
Verification Scope	default
Weight	1.0

#	Name	At Least	Full Path	Goal	Owner	Verification Scope	Weight
1.1	Functional	1	SYSTEM/FIFO /Functional	100	Mike	default	1.0
1.2	chk_1	1	SYSTEM/FIFO/chk_1	100	Bob	default	1.0
1.3	test_cs1	1	SYSTEM/FIFO/test_cs1	100	-	default	1.0
1.4	code	1	SYSTEM/FIFO/code	100	-	default	1.0

**[+] 2 ARBITER**  
**[+] 3 MIC**  
**[+] 4 MEMORY**

Done

Similarly, you can navigate through other vPlan elements and view their details.

### 6.2.2 Sample of Exported Plan (CSV Format)

Figure 6-5 on page 202 shows a sample of verification plan in a CSV format.

**Figure 6-5 Plan in CSV Format**

	A	B	C	D	E	F	G	H	I
1	Section number	At Least	Full Path	Goal	Name	Verification	Weight	Flag	Rich text
2			1 system	100	system	default	1		
3	1		1 system/FIFO	100	FIFO	default	1		
4	1.1		1 system/FIFO/code	100	code	default	1		
5	1.2		1 system/FIFO/assert	100	assertion	default	1		
6	1.3		1 system/FIFO/fd	100	fd	default	1		
7	2		1 system/ARBITER	100	ARBITER	default	1		
8	2.1		1 system/ARBITER/covergroup	100	covergroup	default	1		
9	3		1 system/MIC	100	MIC	default	1		
10	4		1 system/MEMORY	100	MEMORY	default	1		
11									

The plan in a CSV format is shown in a tabular format as a set of rows and columns.

The items selected at the time of exporting the plan are shown in different columns.

### 6.2.3 Sample of Exported Plan (Word Format)

Figure 6-6 on page 203 shows a sample of verification plan in a Word format.

**Note:** In the case of a Word format, plan is exported as a Visual Basic script (.vbs file), which automatically generates a Word file when you run the .vbs file on a Windows machine where Microsoft Word is installed.

**Figure 6-6 Plan in Word Format**

The screenshot shows a Microsoft Word document titled "Verification Plan - ngy top". The document is generated by vManager\_Planner on 20130412-135659, user: ruchikas. It contains the following sections:

- SYSTEM**
  - At Least: 1
  - Full Path: SYSTEM
  - Goal: 100
  - Verification Scope: default
  - Weight: 1.0
- 1 FIFO**
  - At Least: 1
  - Full Path: SYSTEM/FIFO
  - Goal: 100
  - Verification Scope: default
  - Weight: 1.0

Below these sections is a table:

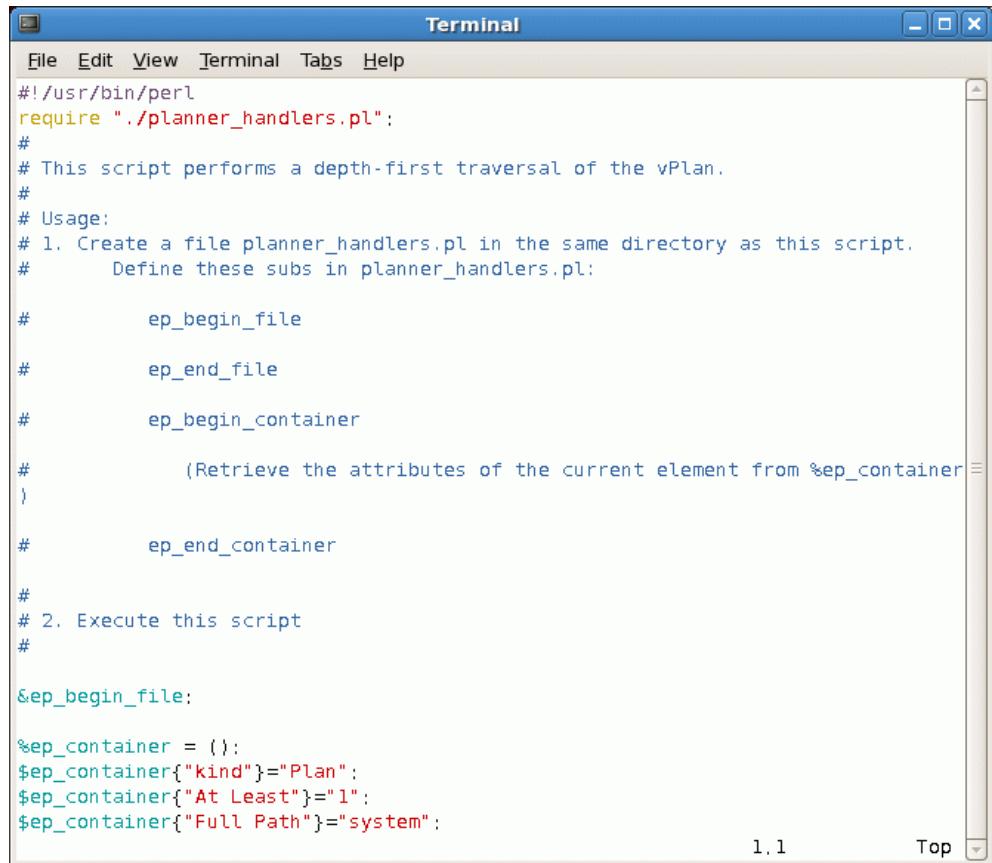
#	Name	At Least	Full Path	Goal	Owner	Verification Scope	Weight
1.1	Functional	1	SYSTEM/FIFO/Functional	100	Mike	default	1.0
1.2	chk_1	1	SYSTEM/FIFO/chk_1	100	Bob	default	1.0
1.3	test_cs1	1	SYSTEM/FIFO/test_cs1	100		default	1.0
1.4	code	1	SYSTEM/FIFO/code	100		default	1.0

You can scroll down the document and view the details of all the elements in the verification plan.

### 6.2.4 Sample of Exported Plan (Perl script Format)

Figure 6-7 on page 204 shows a sample of verification plan in a Perl script format.

**Figure 6-7 Plan in Perl Script Format**



The screenshot shows a terminal window titled "Terminal". The window contains a Perl script. The script starts with a shebang line and requires a file named "planner\_handlers.pl". It includes comments explaining its purpose (depth-first traversal of the vPlan) and usage (creating a file "planner\_handlers.pl" in the same directory). The script defines several subroutines: "ep\_begin\_file", "ep\_end\_file", "ep\_begin\_container", and "ep\_end\_container". It also initializes a hash variable "\$ep\_container" with key-value pairs: "kind" set to "Plan", "At Least" set to "1", and "Full Path" set to "system". The bottom right corner of the terminal window displays the text "1.1 Top".

```
#!/usr/bin/perl
require "./planner_handlers.pl";
#
# This script performs a depth-first traversal of the vPlan.
#
# Usage:
# 1. Create a file planner_handlers.pl in the same directory as this script.
#     Define these subs in planner_handlers.pl:
#
#         ep_begin_file
#
#         ep_end_file
#
#         ep_begin_container
#
#             (Retrieve the attributes of the current element from %ep_container)
#
#         ep_end_container
#
# 2. Execute this script
#
&ep_begin_file;
%ep_container = ();
$ep_container{"kind"}="Plan";
$ep_container{"At Least"}="1";
$ep_container{"Full Path"}="system";
```

You can scroll-down the perl script to view the complete plan.

# Index

---

## A

annotate an existing vPlan element [91](#)  
annotated text [90](#)  
Applying Navigation Filters [101](#)  
Atleast [34](#)

## B

Bottom-up Approach [9](#)

## C

Check port [23](#)  
Coverage port [23](#)  
Creating a Scalar Functional Coverage Item [182](#)  
Creating Check Ports [30](#)  
Creating Child Sections [26](#)  
Creating Coverage Ports [29](#)  
Creating Perspectives [55](#)  
Creating Sibling Sections [28](#)  
csv\_columns\_map [67](#)  
CSV-Based Plan  
    List of Attributes [63](#)  
CSV-Based Verification Plans [61](#)  
Customizing a Functional Coverage Item [186](#)

## D

Deleting Mappings [162](#)  
Deleting Referenced Plans [126](#)  
Designing a Cross or Transition Item [188](#)  
Details [35](#)  
display the annotations [98](#)  
Displaying Annotation Statistics [104](#)

## E

Editing Attributes [39](#)  
element\_id [70](#)

Enabling OpenOffice in vPlanner [112](#)  
Export options [198](#)  
Exporting a Verification Plan [197](#)

## F

Flag [38](#)  
Full path [35](#)  
Functional Coverage Builder [180](#)

## G

Goal [36](#)

## I

Implementation Note [36](#)  
Implementation tab [143, 145](#)  
Importing a plan [118](#)  
Importing a Testbench [180](#)  
Importing and Referencing [117](#)

## L

Loading Coverage Data [143](#)  
locate text in a PDF document [94](#)  
Logical Instance  
    Removing Mapping [138](#)  
Logical Instances [130](#)  
    Applying Mappings [135](#)  
    Creating [131](#)  
    Removing [137](#)  
Logical Instances Tab Page [134](#)

## M

Mapping Coverage Metrics [150](#)  
Mapping Pattern [38](#)  
    Adding a New Pattern [159](#)  
    Modifying an Existing Pattern [158](#)  
Menu Bar [13](#)

Message Pane [19](#)  
Metrics Port Kind [36](#)  
Metrics tab [144, 147](#)  
Migrating Annotations window [107](#)

## N

Navigating Between Annotations [101](#)  
New Verification Plan [24](#)

## O

Opening a Verification Plan [19](#)  
Owner [36](#)

## P

Parameter condition [38](#)  
Parameters [127](#)  
    Associating with vPlan elements [128](#)  
    Declaring [127](#)  
PDF annotated spec text [91](#)  
Perspectives [23](#)  
Planner User Interface [12](#)  
Priority [36](#)

## R

Referencing (Instantiating) a plan [122](#)  
Reloading Referenced Plans [126](#)  
Removing Coverage Data [149](#)  
Rich-text Editing [111](#)

## S

Sections [23](#)  
Spec Annotator area [84](#)  
Specs—Spec manager tab [88](#)  
subsections [23](#)  
Summary Pane [17](#)  
Synchronizing the Plan [120](#)

## T

Tabs pane [16](#)

Testcase (scenario) ports [23](#)  
Toolbar [15](#)  
Top-down Approach [9](#)

## U

Updating the Specification Document [105](#)  
Upgrading Absolute pathnames to use  
    logical instances [140](#)  
Using the Rich-text Editor [113](#)

## V

Verification Scope [37](#)  
Viewing Mappings [161](#)  
Viewing the Exported Plan [199](#)  
vPlan Editor [16](#)  
vPlan Hierarchy  
    Advanced search and filter [49](#)  
    Common Queries [48](#)  
    Searching and Filtering Items [43](#)

## W

Weight [38](#)  
Writing e code and Regenerating Coverage  
    Model [194](#)