VECTOR



VectorCAST

Quick Start Guide

VectorCAST 2026

New editions of this guide incorporate all material added or changed since the previous edition. Update packages may be used between editions. The manual printing date changes when a new edition is printed. The contents and format of this manual are subject to change without notice.

Generated: 10/19/2026, 8:29 PM

Rev: 6be4125

Part Number: VectorCAST Quick Start Guide for VectorCAST 2026 SP5

U.S. Government Restricted Rights

This computer software and related documentation are provided with Restricted Rights. Use, duplication or disclosure by the Government is subject to restrictions as set forth in the governing Rights in Technical Data and Computer Software clause of

DFARS 252.227-7015 (June 1995) and DFARS 227.7202-3(b).

Manufacturer is Vector North America, Inc. East Greenwich RI 02818, USA.

Imprint

Vector reserves the right to modify any information and/or data in this user documentation without notice. Vector disclaims all liabilities for the completeness or correctness of the contents and for damages which may result from the use of this documentation. This documentation nor any of its parts may be reproduced in any form or by any means without the prior written consent of Vector. To the maximum extent permitted under law, all technical data, texts, graphics, images and their design are protected by copyright law, various international treaties and other applicable laws. Any unauthorized use may violate copyright and other applicable laws or regulations.

© Copyright 2026, Vector Informatik, GmbH. All rights reserved.

Protected Trademarks

All brand names in this documentation are either registered or non-registered trademarks of their respective owners.

Third-Party copyright notices are contained in the folder: 3rdPartyLicenses, located in the VectorCAST installation directory.

TABLE OF CONTENTS

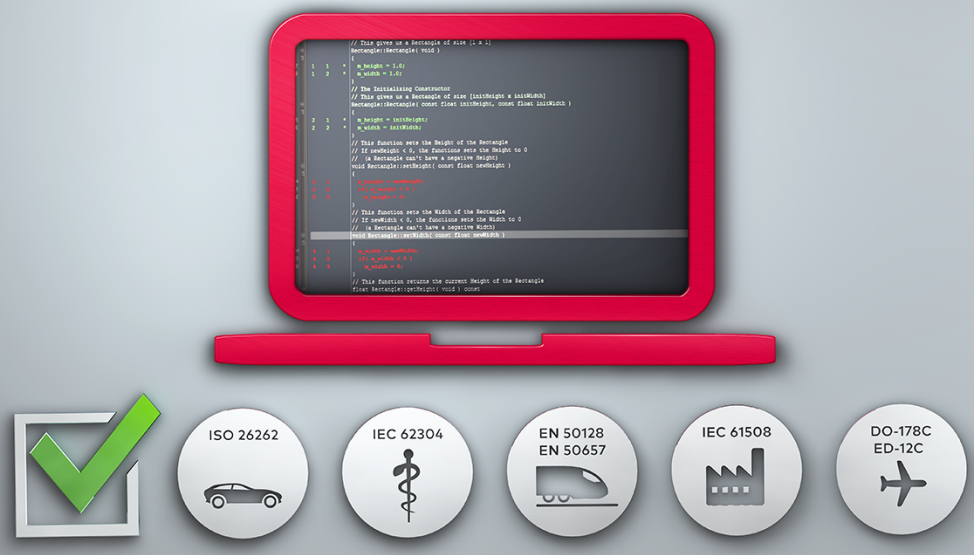
Introduction 4

VectorCAST Overview 5
Starting VectorCAST (Windows Users) 5
Starting VectorCAST (Linux Users) 6
Troubleshooting 8
VectorCAST Interface 8

Getting Started With VectorCAST 10

Create a VectorCAST Project 11
Set the Working Directory 11
Create a New Project 12
Add a Unit Test Environment 13
Add a Test Case 18
Execute All Tests 20

Index 22



Introduction

VectorCAST Overview

VectorCAST is a suite of test automation tools:

- VectorCAST/C++ and VectorCAST/Ada automate the unit and integration testing of C, C++, and Ada code, allowing you to easily test any subset of files (or packages for Ada) in isolation from the rest of the application.
- VectorCAST/QA enables teams to implement consistent and repeatable processes for managing test activities and reporting key quality metrics such as code coverage.

This *Quick Start Guide* is intended to get you started with the basic features of VectorCAST. Use it for a quick reference.

For more detailed information about VectorCAST product features, please refer to the *User's Guides* for VectorCAST/C++, VectorCAST/Ada, Enterprise Testing, and VectorCAST/QA.

Starting VectorCAST (Windows Users)

As you start: Ensure that VectorCAST is installed. Refer to the *Installation Guide* for detailed installation instructions.

To start VectorCAST, a startup script is recommended. A startup script makes it easier for you to understand configuration settings and allows more flexibility regarding the initial startup. It is a good practice to have two scripts, one to setup the environment and another to start VectorCAST. For example, you may have a need to integrate VectorCAST into a CI system such as Jenkins, and having a script that only sets up the environment will be useful.

To begin, create an empty directory you can use for the examples in this guide. There can be no spaces in the path and there are some restricted characters, so it is suggested to start off simple. In this example, we use **`D:\setup_dir`**.

In this example, we have two environment variables that VectorCAST uses to find where VectorCAST is installed and identify what license manager it should use. There are multiple ways to do this, but to get started, we will start with a basic example.

Environment Variable	Purpose	Example Setting
VECTORCAST_DIR	Full path to the folder where VectorCAST is installed.	D:\VCAST\2026-beta
VECTOR_LICENSE_FILE	For legacy licensing using FlexLM, it has the format portnumber@ [ipaddress or hostname] For NVLM using the Vector License Client, this variable **must** be unset.	7650@licenseserver.domain.com

Create a script file like the one shown below and name it **`setup_env.bat`**

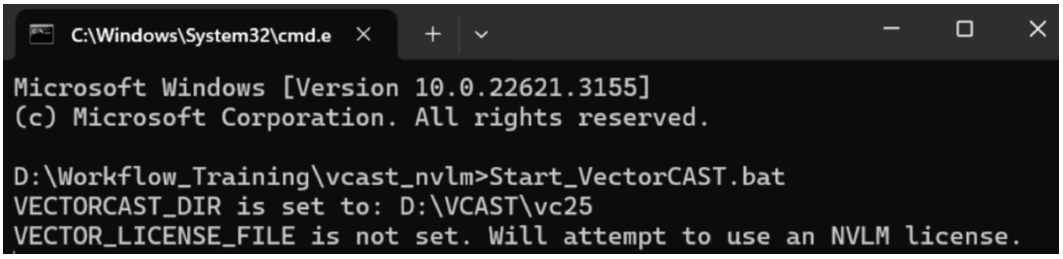
```
@ECHO OFF

REM VECTORCAST_DIR - Environment variable pointing to the base directory for
```

```
VectorCAST.
SET "VECTORCAST_DIR=D:\VCAST\2026-beta"

REM VECTOR_LICENSE_FILE - Environment variable for legacy licensing
SET "VECTOR_LICENSE_FILE=7650@licenseserver.domain.com"

REM VECTOR LICENSE FILE - Environment variable for NVLM
REM SET "VECTOR_LICENSE_FILE="
```

Create a script file like the one shown below and name it **`Start_VectorCAST.bat`**.

```
@ECHO OFF

CALL setup_env.bat
if not defined VECTOR_LICENSE_FILE echo VECTOR_LICENSE_FILE is not set. Will attempt
to use an NVLM license.
if defined VECTOR_LICENSE_FILE echo VECTOR_LICENSE_FILE is set to:
%VECTOR_LICENSE_FILE% Will attempt to use a legacy license.

REM Start VectorCAST
%VECTORCAST_DIR%\vcastqt
```

Now open a command prompt and run the **`Start_VectorCAST.bat`** script.



If all goes well, you should see the VectorCAST GUI.

Starting VectorCAST (Linux Users)

As you start: Ensure that VectorCAST is installed. Refer to the *Installation Guide* for detailed installation instructions.

To start VectorCAST, a startup script is recommended. A startup script makes it easier for you to understand configuration settings and allows more flexibility regarding the initial startup. It is a good practice to have two scripts, one to setup the environment and another to start VectorCAST. For example, you may have a need to integrate VectorCAST into a CI system such as Jenkins, and having a script that only sets up the environment will be useful.

To begin, create an empty directory you can use for the examples in this guide. There can be no spaces in the path and there are some restricted characters, so it is suggested to start off simple. In this example, we use **`/home/vector/setup_dir`**.

In this example, we have two environment variables that VectorCAST uses to find where VectorCAST is installed and identify what license manager it should use. There are multiple ways to do this, but to get started, we will start with a basic example.

Environment Variable	Purpose	Example Setting
VECTORCAST_DIR	Full path to the folder where VectorCAST is installed.	/home/vector/vcast/2026-beta
VECTOR_LICENSE_FILE	For legacy licensing using FlexLM, It has the format portnumber@[ipaddress or hostname] For NVLM using the Vector License Client, this variable **must** be unset.	27000@192.168.0.70

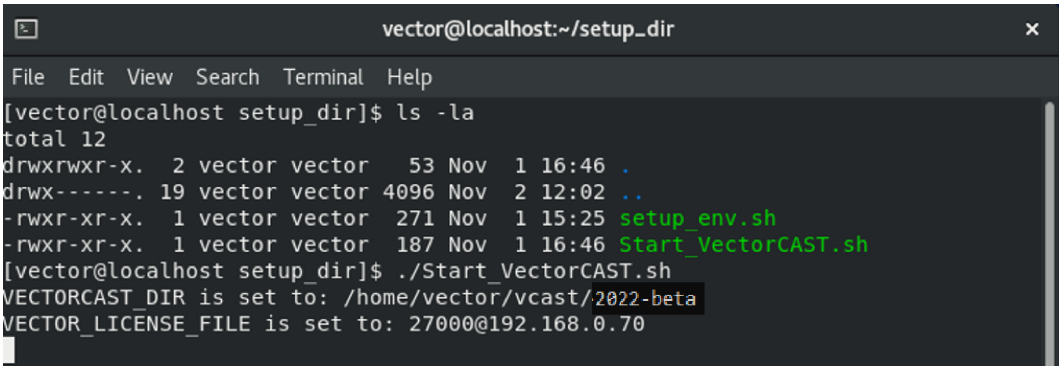
Create a script file like the one shown below and name it **`setup_env.sh`**

```
#!/bin/bash

# VECTORCAST_DIR - Environment variable pointing to the base directory for VectorCAST

export VECTORCAST_DIR=/home/vector/vcast/2026-beta

# VECTOR_LICENSE_FILE - Environment variable for FLEXlm

export VECTOR_LICENSE_FILE=27000@192.168.0.70

# VECTOR_LICENSE_FILE - Environment variable for Vector License Client

#unset VECTOR_LICENSE_FILE
```

Create a script file like the one shown below and name it **`Start_VectorCAST.sh`**.

```
#!/bin/bash

. ./setup_env.sh
echo "VECTORCAST_DIR is set to: ${VECTORCAST_DIR}"
echo "VECTOR_LICENSE_FILE is set to: ${VECTOR_LICENSE_FILE}"
if [ -n "${VECTOR_LICENSE_FILE}" ]
then
    echo "VECTOR_LICENSE_FILE is set to: ${VECTOR_LICENSE_FILE}"
elseif [ "${VECTOR_LICENSE_FILE+defined}" = defined ]
then
    echo “VECTOR_LICENSE_FILE empty but defined, possible error”
else
    echo 'VECTOR_LICENSE_FILE is unset, will attempt to use an NVLM license'
# Start VectorCAST
${VECTORCAST_DIR}/vcastqt
```

Now open a shell and run the script.



If all goes well, you should see the VectorCAST GUI.

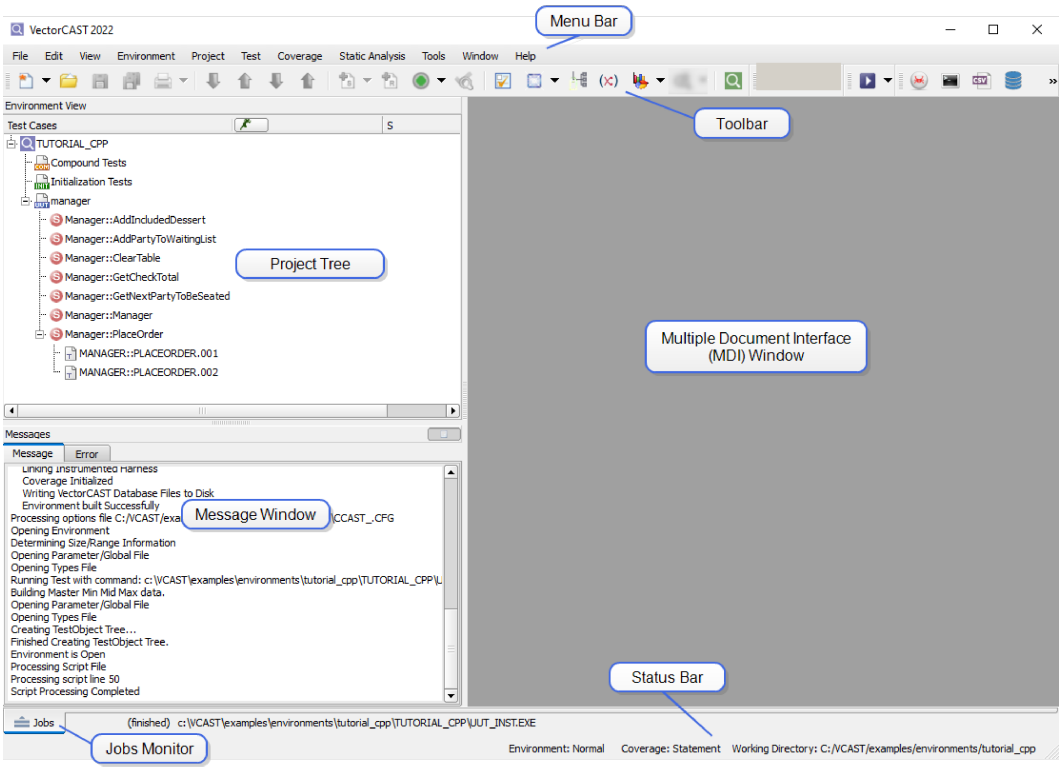
Troubleshooting

If the script did not work, there are a few things to check to be sure you have everything correct:

- Check that the path to the VectorCAST installation is correct
- It's possible you mistyped the path to where VectorCAST is installed.
- If the path is very long or has special characters, it may not work.
- If the path is on a networked filesystem, there may be issues.
- If you are sure the path is correct, maybe re-installing on a local disk with a simple pathname will fix the problem.

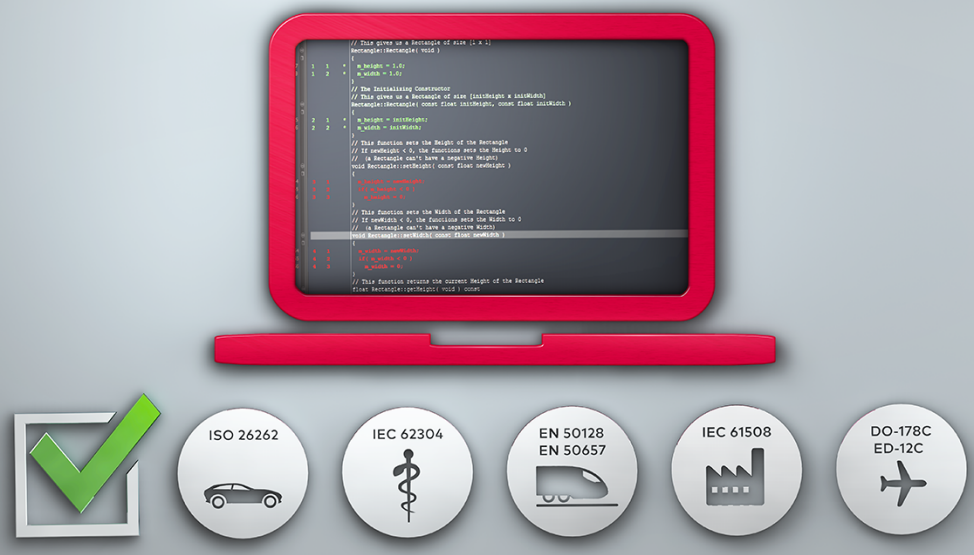
VectorCAST Interface

Below we discuss the default controls of the VectorCAST GUI. Note that you can return to this default arrangement at any time by using: **View =>Default Layout** from the menu bar.



The VectorCAST main window is divided into four panes:

- **The Project Tree** is located on the left-hand side of the main window. It provides a high level view of the project structure.
- **The Message Window** is located along the bottom left of the main window. It contains tabs for informational messages and for error messages.
- **The Multiple Document Interface (MDI) Window** is located to the right of the Project Tree. It displays a variety of windows, including Test Case editors, Coverage Viewers, Report Viewers and Text Editors. Windows are collected into groups. See the *VectorCAST User Guides* for more information on MDI Window Groups.
- **The Jobs Monitor** is located on the bottom of the main window. It displays the status of jobs as they execute and exposes the associated back-end commands. See the *VectorCAST User's Guides* for more information on the Jobs Monitor.



Getting Started With VectorCAST

Create a VectorCAST Project

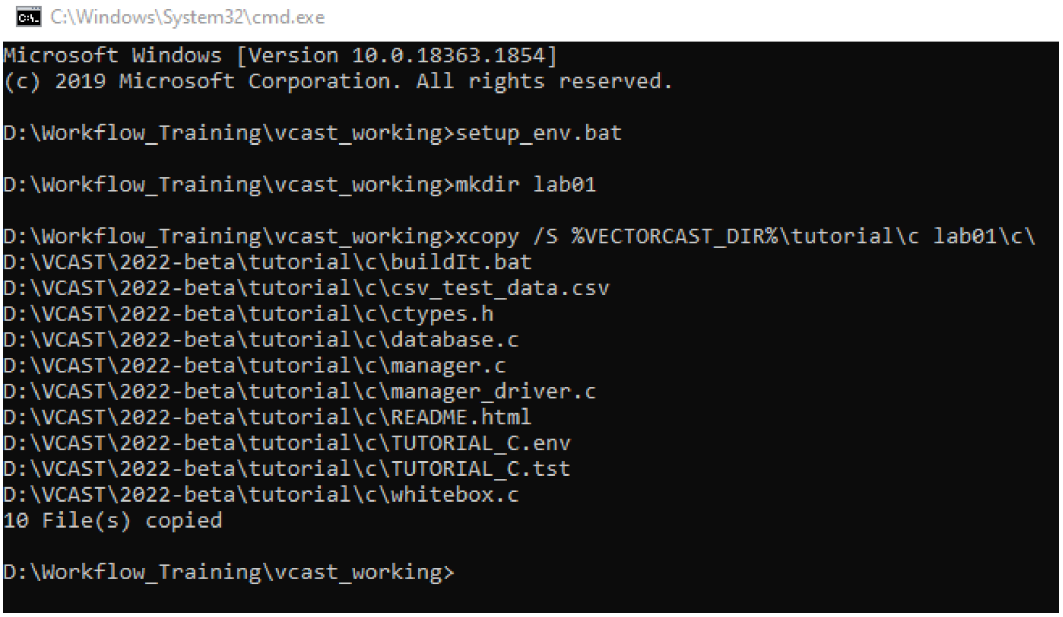
Enterprise Testing is a Test Automation Framework that sits on top of VectorCAST/C++ or VectorCAST/Ada test environments and allows test design, execution, and reporting to be distributed across the enterprise. The VectorCAST Project supports a variety of work flows allowing for team collaboration, testing of multiple configurations, change-based testing, and massively parallel testing.

Enterprise Testing can import existing VectorCAST/C++ and VectorCAST/Ada test environments, or be used to create new environments.

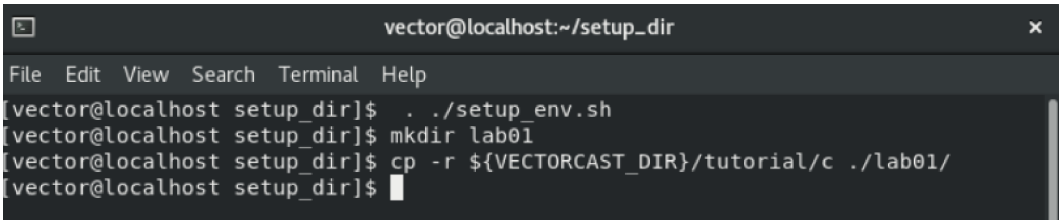
In this section, you will create a new VectorCAST Project starting with an initial unit test environment.

To prepare for this step, some initial setup is required. First, you will copy the source files used in this example. The following examples show how this is done for both Windows and Linux hosts.

Windows Example



Linux Example



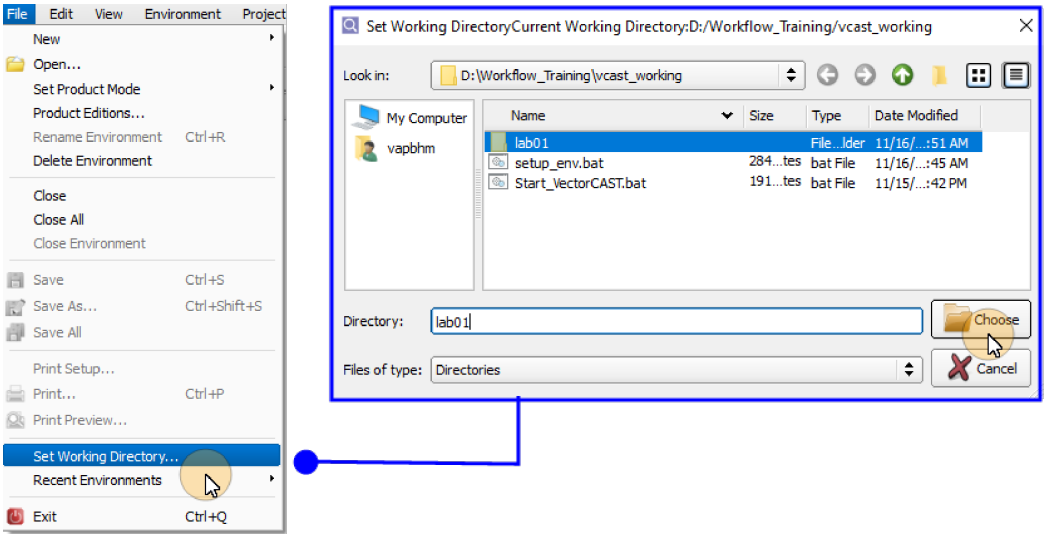
Next, you will start VectorCAST using the **`Start_VectorCAST.bat`** script located in this directory.

Set the Working Directory

Once VectorCAST starts and the GUI opens, you will set the working directory to the directory you

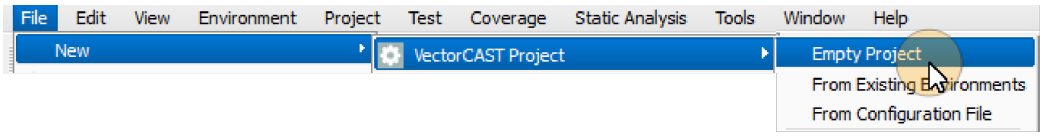
created above (`lab01`).

From the Menu Bar, select **File => Set Working Directory...**. Navigate to your working directory and select **Choose**. For our example, we select the `lab01` directory as our working directory.



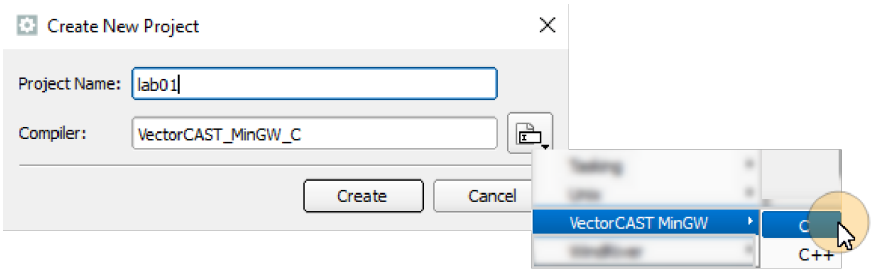
Create a New Project

Create a new empty project by selecting **File => New => VectorCAST Project => Empty Project** from the Menu Bar.

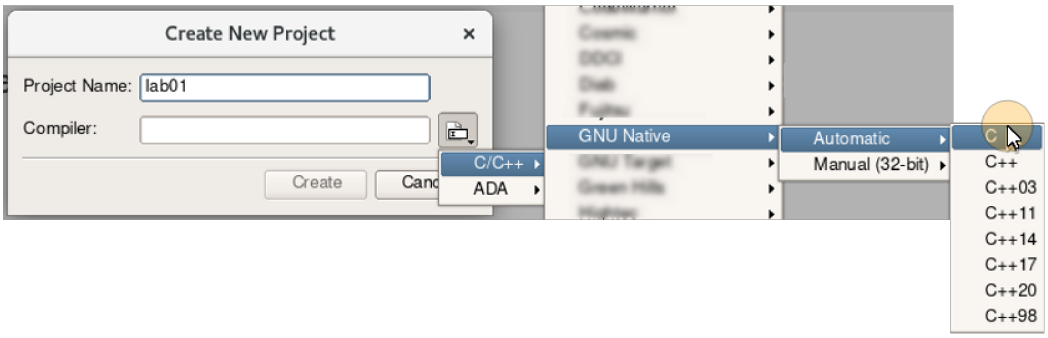


The next step is to name the Project and select a compiler suite to use. The compiler suite selection will be different depending on if you are using Windows or a Linux host. With Windows, VectorCAST includes a Windows MinGW compiler that can be used. For Linux, you need to use a native gcc compiler and insure that the compiler is on the current path.

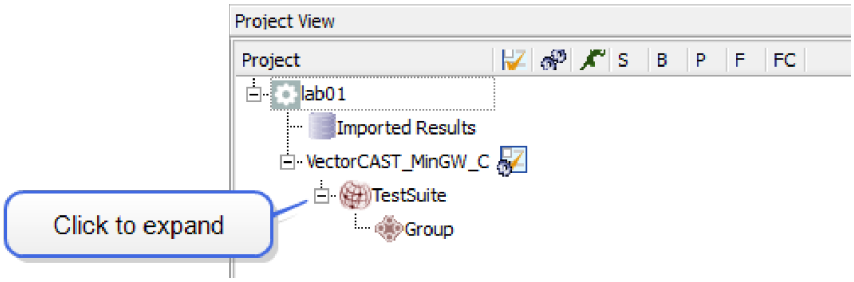
Windows Example



Linux Example

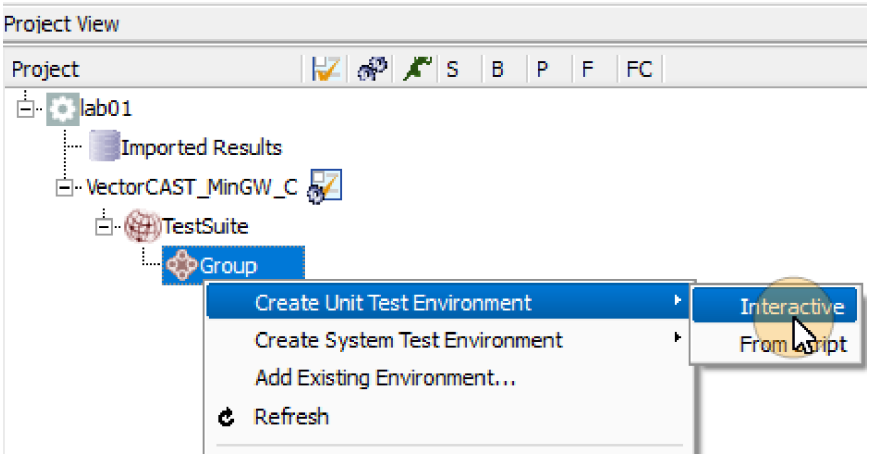


Select the **Create** button. The new VectorCAST Project, `lab01`, is displayed in the Project View.



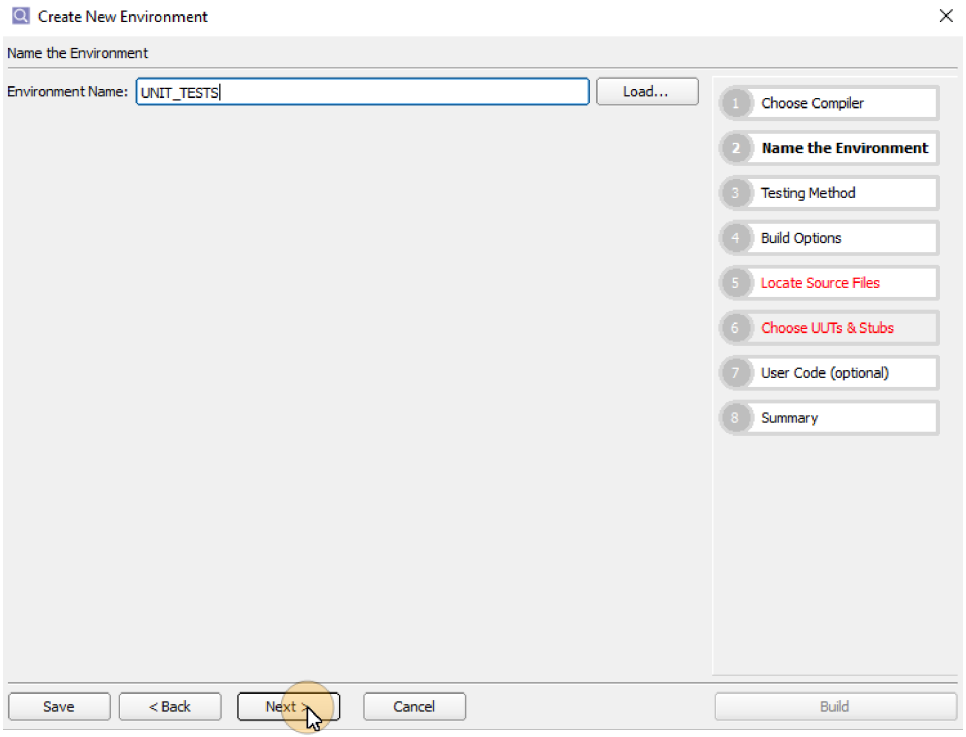
Add a Unit Test Environment

Expand the Project Tree and right-click on the **Group** node. Select **Create Unit Test Environment => Interactive** from the context menu.

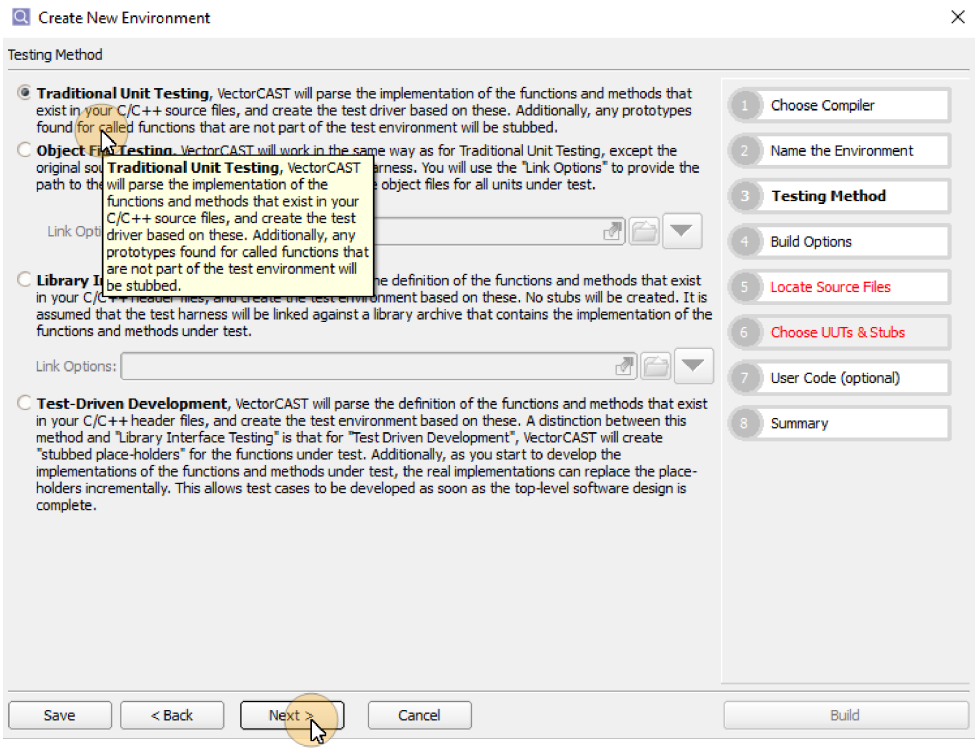


The Create New Environment wizard opens.

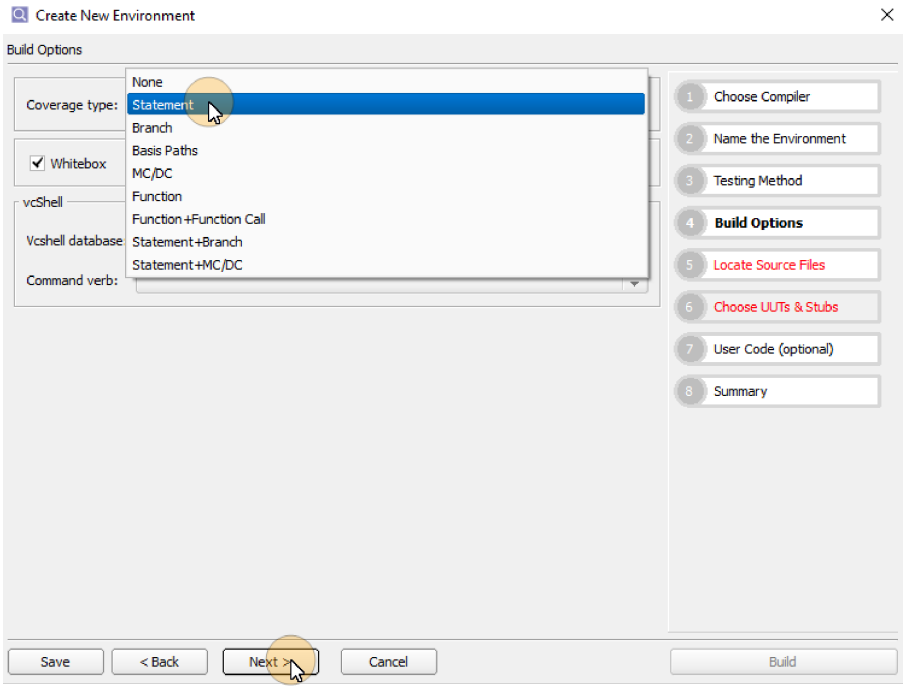
Enter the Environment name. Our example environment is named **`UNIT_TESTS`**. Select the **Next** button.



Select the Testing Method. For our example, we select **Traditional Unit Testing**. Select the **Next** button.

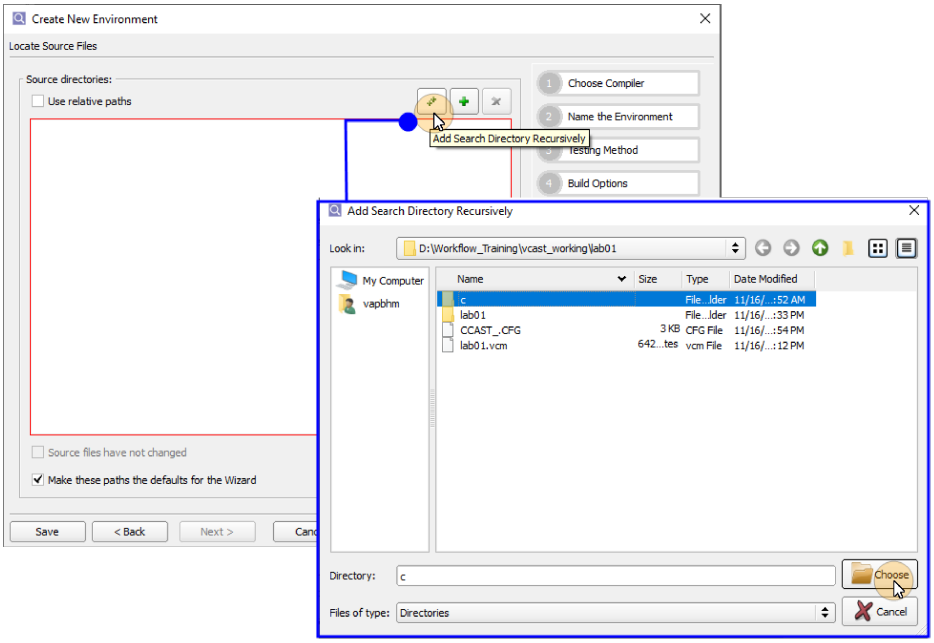


Select the Coverage Type. Using the Coverage type drop-down menu, select **Statement** coverage. Select the **Next** button.

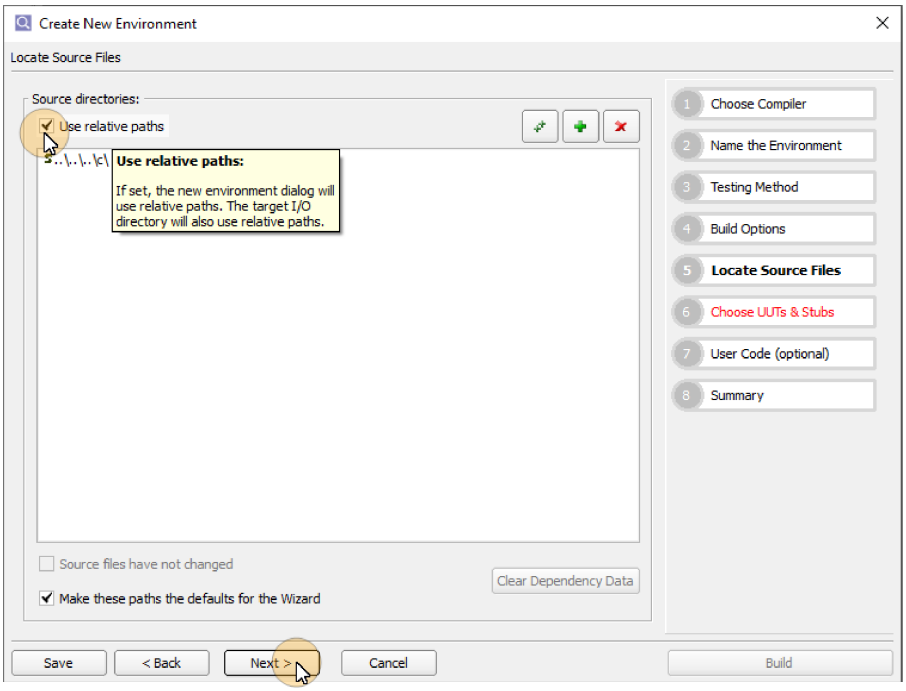


Add Search Directory. Add a Search Directory recursively by selecting the

button. Add the `c` directory that was created earlier. Select the **Choose** button.

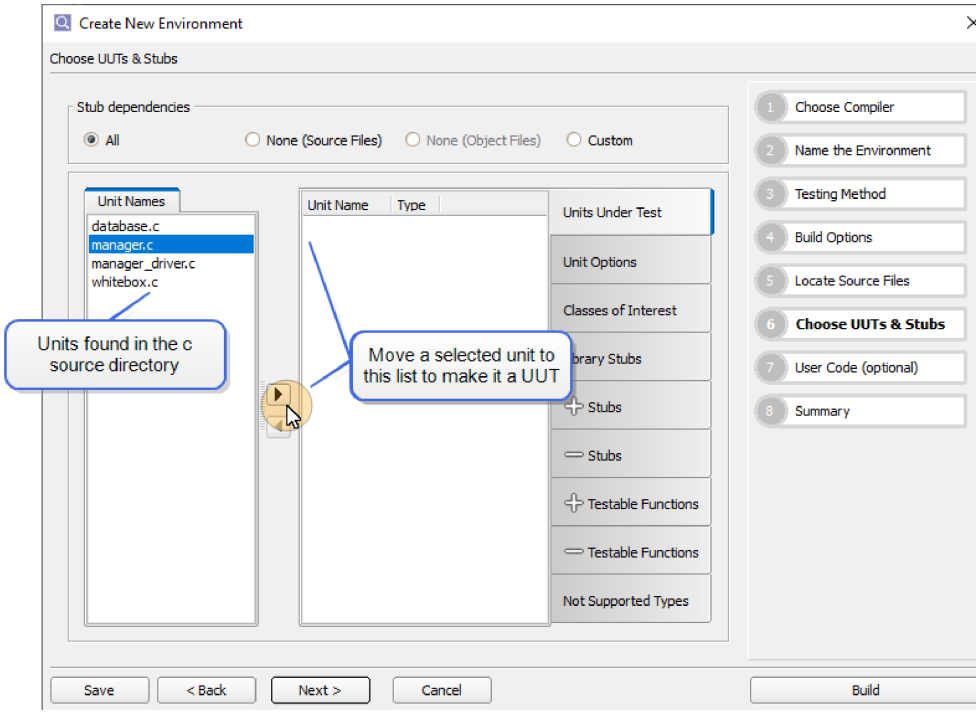


Use Relative Paths. Check the **Use relative paths** checkbox. Select the **Next** button.

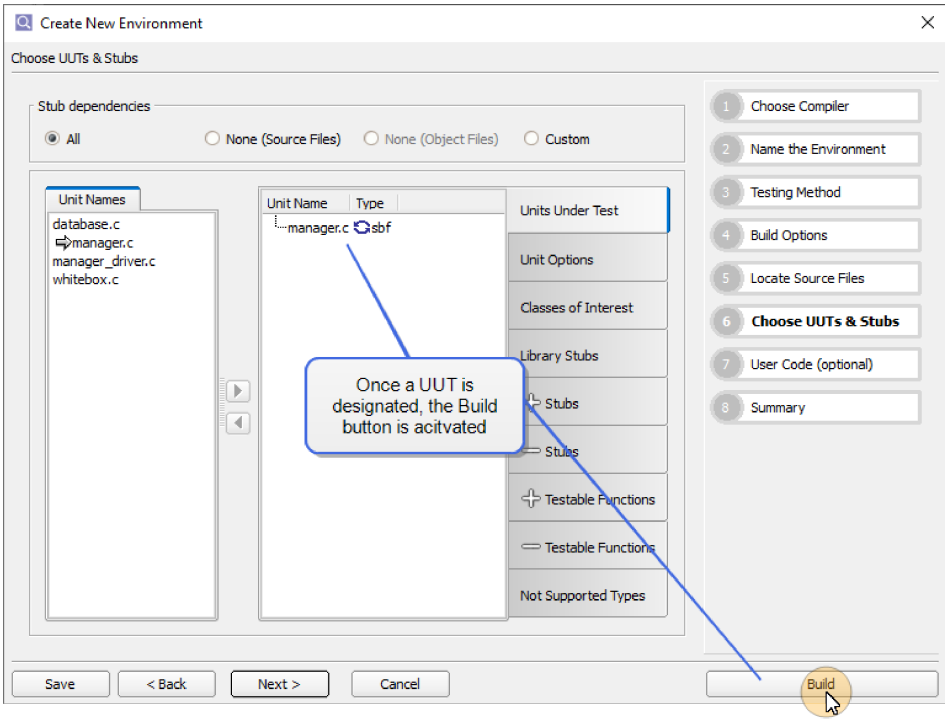


Choose the UUT. For our example, we will select **`manager.c`** as our UUT. Under the Unit Names tab, select **`manager.c`**. Click the move-to-right button to place **`manager.c`** in the Units Under

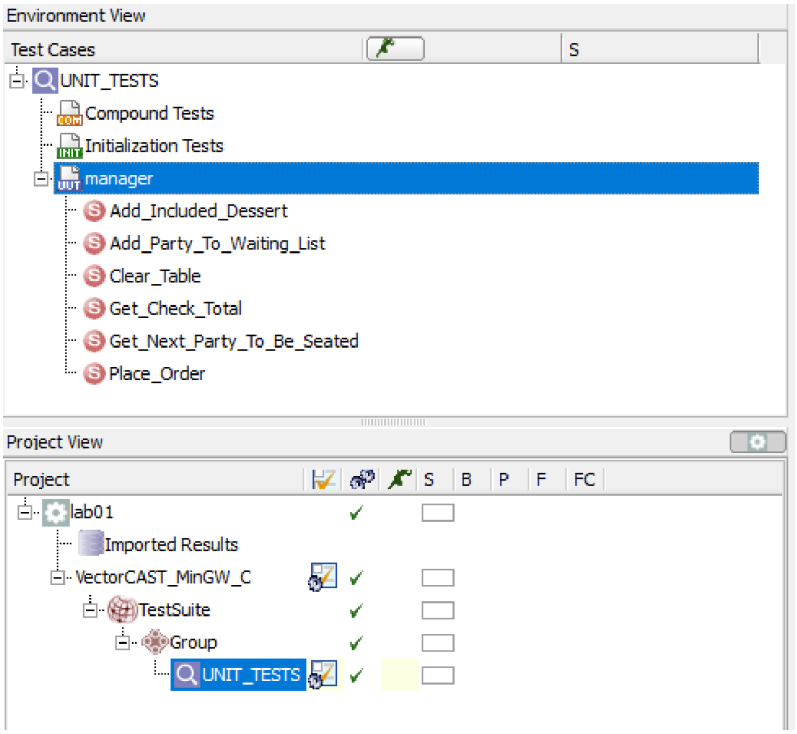
Test list.



Build the Environment. Once a UUT is designated, the **Build** button is enabled. Select the **Build** button.



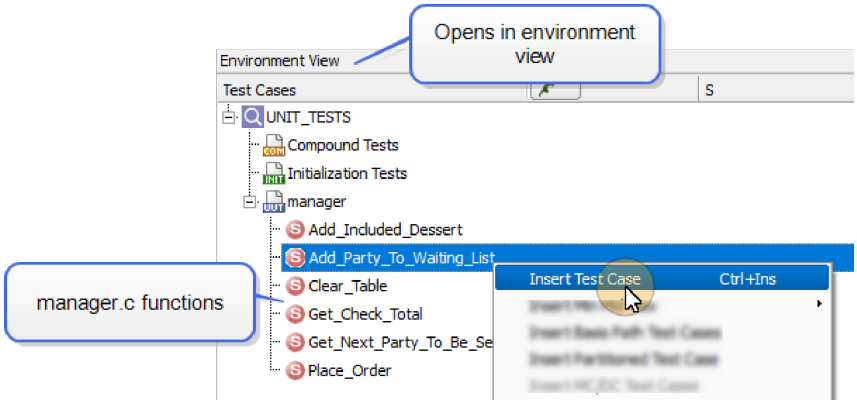
You now have an initial Unit Test environment in your VectorCAST Project.



Add a Test Case

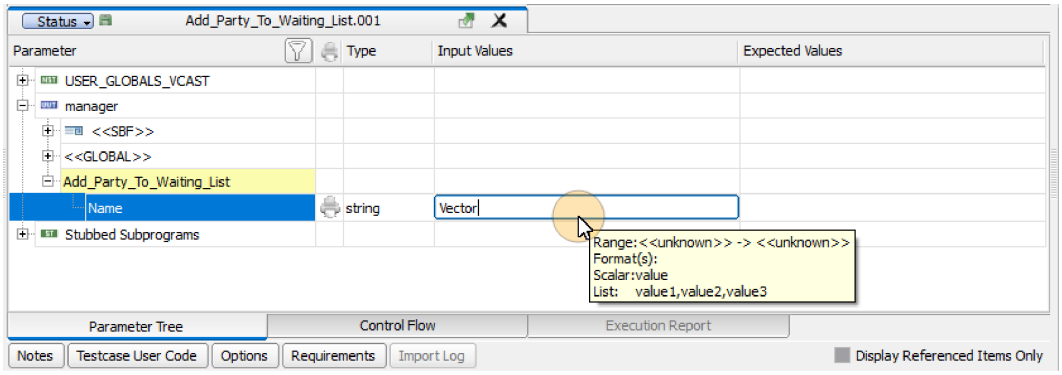
To add a test case, select the UNIT_TESTS environment in the Project View and double-click on the selection. The environment opens in the Environment View pane. All the **`manager.c`** functions are displayed in the Environment View.

Select **`Add_Party_To_Waiting_list`** and right click. Select **Insert Test Case** from the menu to open the Unit Test Case Editor.

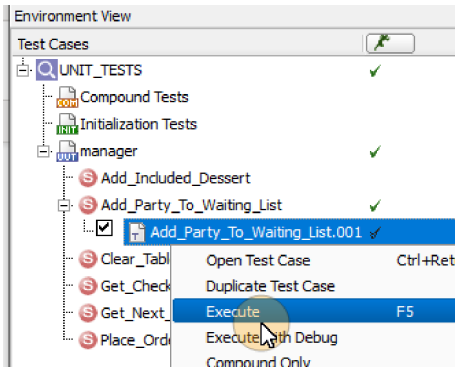


You will see that VectorCAST has already parsed the UUT and all the parameters and their type are listed. For this unit, we have one parameter called "**`Name`**" which is of type **`string`**. For this test case,

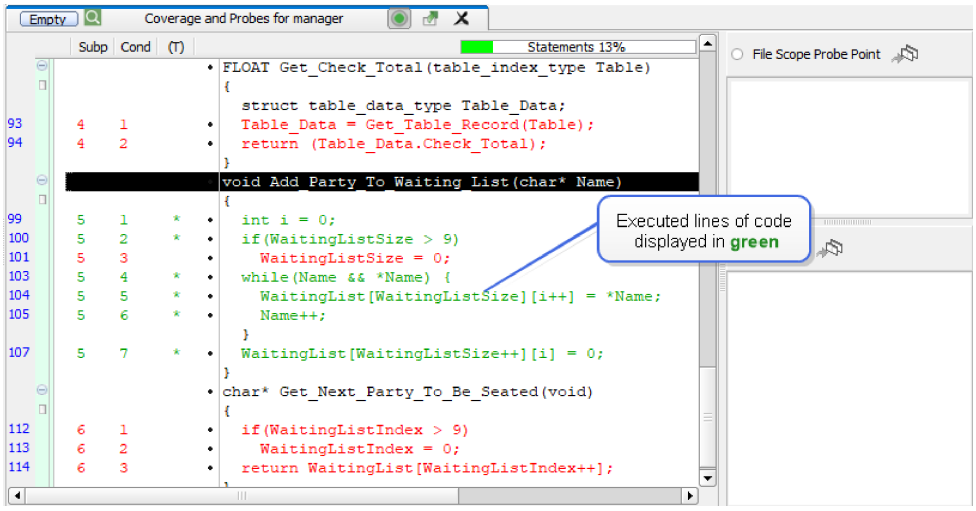
we will add a string to pass to the unit. Select the cell under Input Values on the same row as "**`Name`**". Type in **`Vector`**.



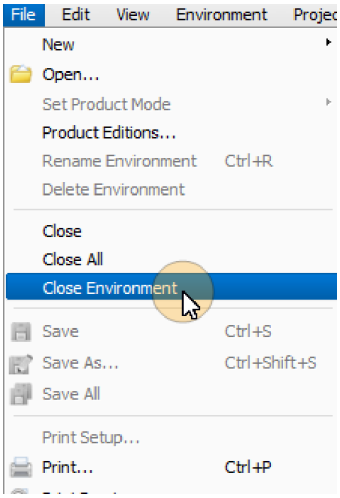
Right-click on the test **`Add_Party_To_Waiting_List`** in the Environment View and select **Execute** from the menu.



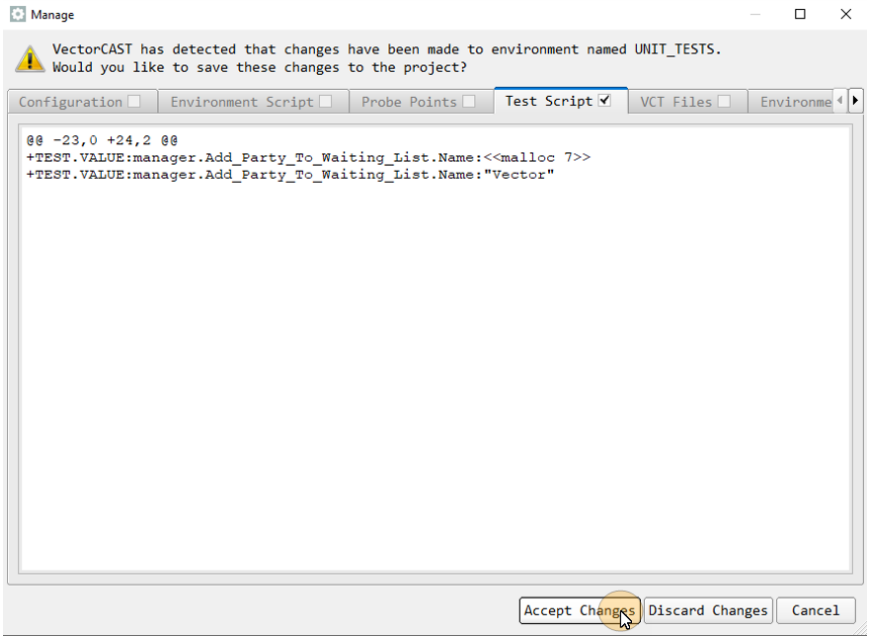
Once the test executes, the Coverage Viewer opens and you will see the lines that were executed as a result of this test displayed in green.



Close the environment and return to the Project View by selecting **File => Close Environment** from the Menu Bar.

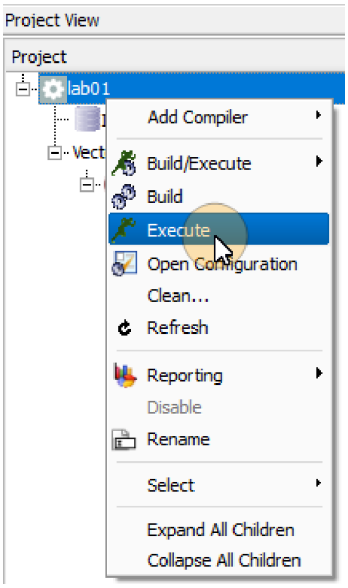


A dialog box appears asking you to confirm the changes you made to the environment. Select the **Accept Changes** button. The Project View is displayed.



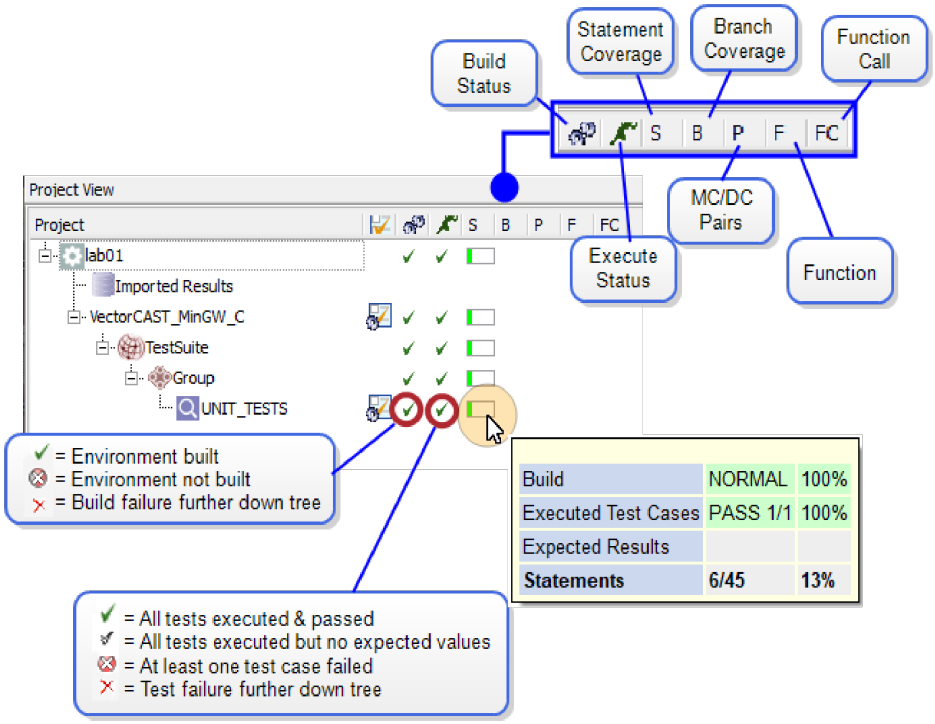
Execute All Tests

To execute all of the tests in the **`lab01`** project, right-click on the Project Name (**`lab01`**) and select **Execute** from the context menu.



You can follow the execute process in the Manage Status viewer which opens in the MDI Window. As test cases are executed data is stored in a SQL database and used to generate reports showing testing status and trends, making it easy to analyze regression trends.

The Status Panel updates to display testing status. On the status panel you will see status for the Environment Build, Test Execution and Statement Coverage. Hover over the Statement Coverage bar to see a pop-up of the Build and Coverage details.



Index

add a test case 18
add unit test environment 13
build the environment 17
choose uut 16
close environment 20
coverage type 15
create a vectorcast project 11-12
create new environment wizard 14
create unit test environment 13
enterprise testing 11
    create a project 11
environment name 14
execute a test case 19
execute all tests 20
insert a test case 18
jobs monitor 9
linux
    create a vectorcast project 11
    environment variables 6
    example setup script 7
    example startup script 7
    startup script 6
manage status viewer 21
MDI window 9
message window 9
project tree
    location of 9
relative paths 16
search directory 15
starting vectorcast
    linux 6
    windows 5
status panel 21
testing method 14
troubleshooting scripts 8
VectorCAST
    starting linux 6
    starting windows 5
VectorCAST interface
    jobs monitor 8
    MDI window 8
    message window 8
    project tree 8
windows
    create a vectorcast project 11
    environment variables 5
    example setup script 5
    example startup script 6
    startup script 5
working directory 11