VECTOR

VectorCAST HighTec Integration

Version 1.0

2022-08-16

Application Note AN-ACT-1-105

Author Jaime Segura

Restrictions Public Document

Abstract Information on the HighTec and VectorCAST Integration

Table of Contents

1 Overview

VectorCAST Runtime Support Package (RSP) is an extension of the VectorCAST product. It provides an interface layer that allows you to use the VectorCAST testing techniques and methods on an embedded target processor. The VectorCAST RSP will always run on your host platform (the same platform as the compiler). Only the VectorCAST generated test harness will be downloaded to, and executed on, the embedded target.

A VectorCAST RSP is always customized to a particular Target CPU, Cross Compiler, and Run-Time environment (or kernel). As such, the information presented here contains sections for this specific RSP.

This application note provides details on configuring VectorCAST to run tests on the Infineon (TSIM) Simulator or the Lauterbach Trace32 (t32mtc) Simulator using the HighTec compiler.

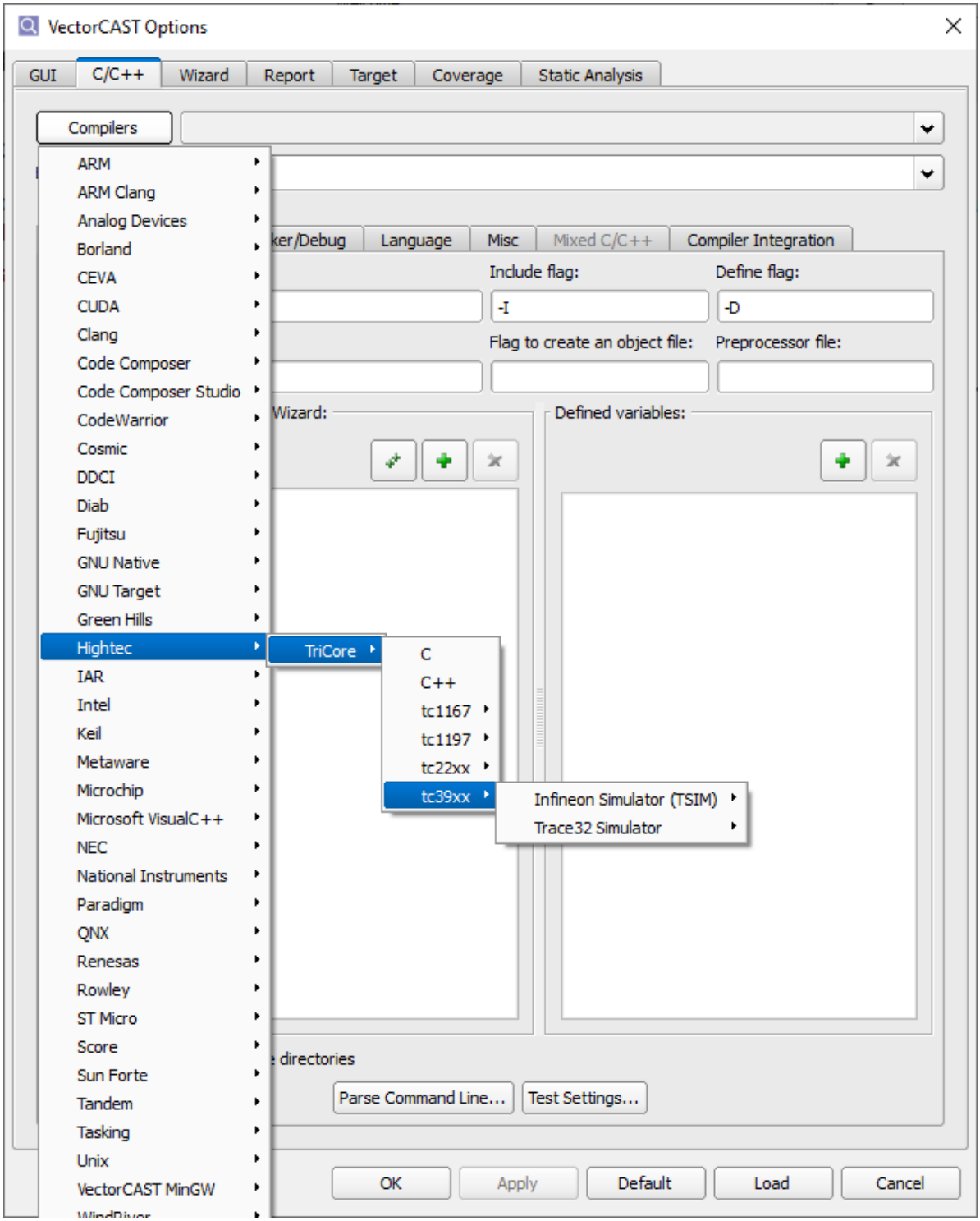
The examples shown here use the following tools and versions:

- HighTec Toolchain compiler (tricore-gcc/tricore-g++) version 4.9.3.0
- HighTec Toolchain simulator (TSIM) version 1.18.1
- t32mtc version 2021 SP2
- VectorCAST version 2022 SP3

2 Use Cases

2.1 Selecting the Proper Configuration

Choose the configuration that best matches the tools that are available for software development. All four chip options will contain the same available simulator RSPs. The subsequent sections highlight typical use cases.



2.2 Use Case: Lauterbach Trace32 Simulator

Lauterbach provides support for many chip families, including Tricore. The Trace32 Simulator for Tricore (t32mtc) can be installed in any directory.

2.2.1 Setup

The environment variable VCAST_TRACE32 must be set to the directory where the binary of the simulator exists. It is used to execute the simulator in the appropriate directory from Trace32. The HighTec compiler “tricore-gcc/tricore-g++” must be on the default search path.

2.2.2 Execution Method

VectorCAST uses a Python script to control the Trace32 (t32mtc) Simulator. As part of the RSP, VectorCAST is configured to capture the coverage and test case results by placing a breakpoint in the program execution and purging the data buffer.

The Python script performs the following tasks:

1. Write out the Trace32 (`simulator.cmm`) script for controlling t32mtc
2. Write out the Trace32 (`configsim.t32`) file for configuring t32mtc
3. Change the working directory to the location of t32mtc
4. Execute t32mtc and provide the `simulator.cmm` and `configsim.t32` files as parameters

The contents of simulator.cmm contains the logic to capture the coverage and test case results. `simulator.cmm` performs the following tasks:

1. Initialize the simulator for the given CPU
2. Load the test case data and harness image
3. Place a breakpoint at `vCAST_END()`
4. Place a break at `vcastHaltDebugger()`
5. Open a file handle to `VCAST_STDOUT.DAT`
6. Execute the binary image until the output buffer "`vcast_output_buffer`" is full or the breakpoint `vCAST_END()` is hit
7. If the buffer is full, write out the contents of the output buffer to the file VCAST_STDOUT.DAT. If more data needs to be fetched, repeat steps 6 and 7
8. If the breakpoint `vCAST_END()` is hit, close the file handle and exit the simulator

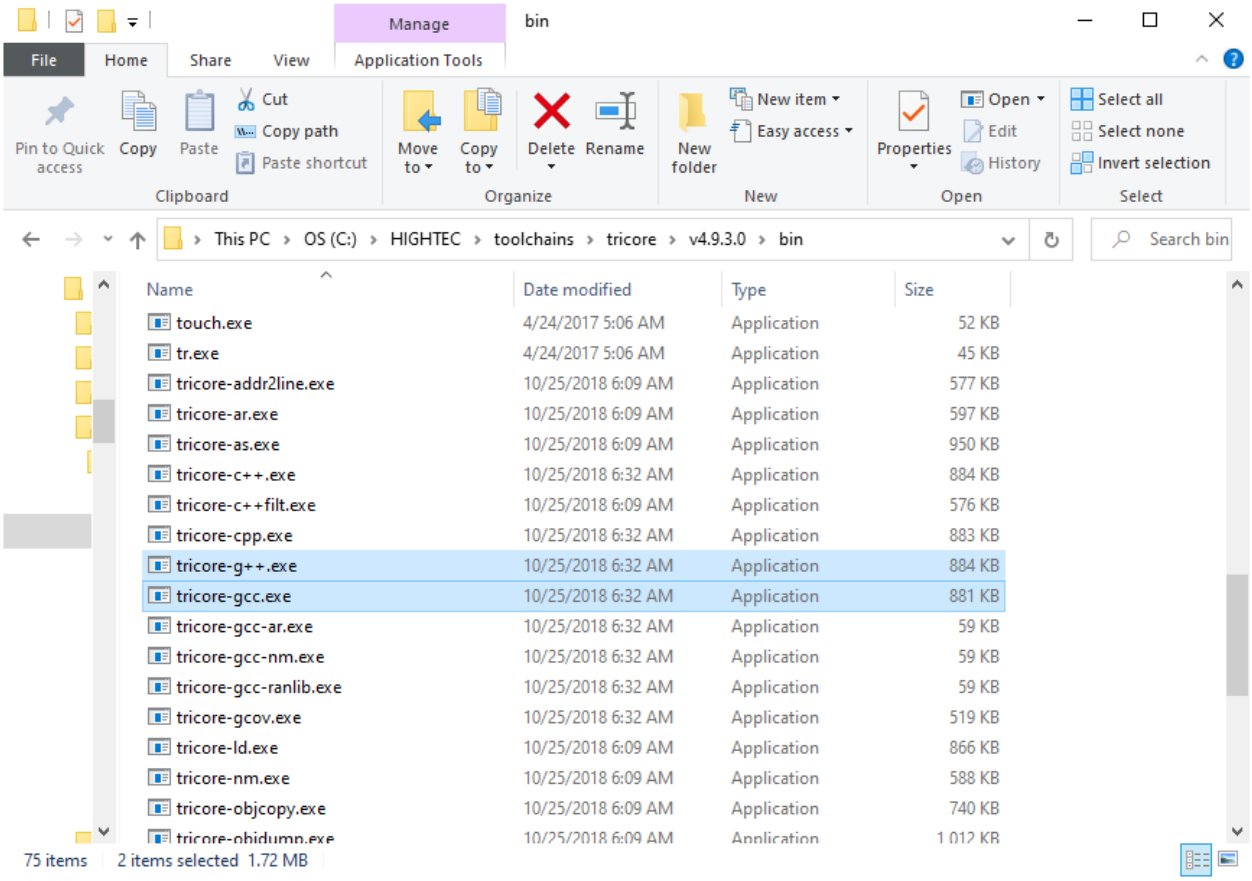
The contents of `configsim.t32` contains configuration information for the simulator. `configsim.t32` contains:

1. Setting for configuring t32mtc to use the simulator
2. Setting for the Trace32 IDE to appear minimized
3. If executing a VectorCAST test case in debug mode, the Trace32 IDE will appear maximized

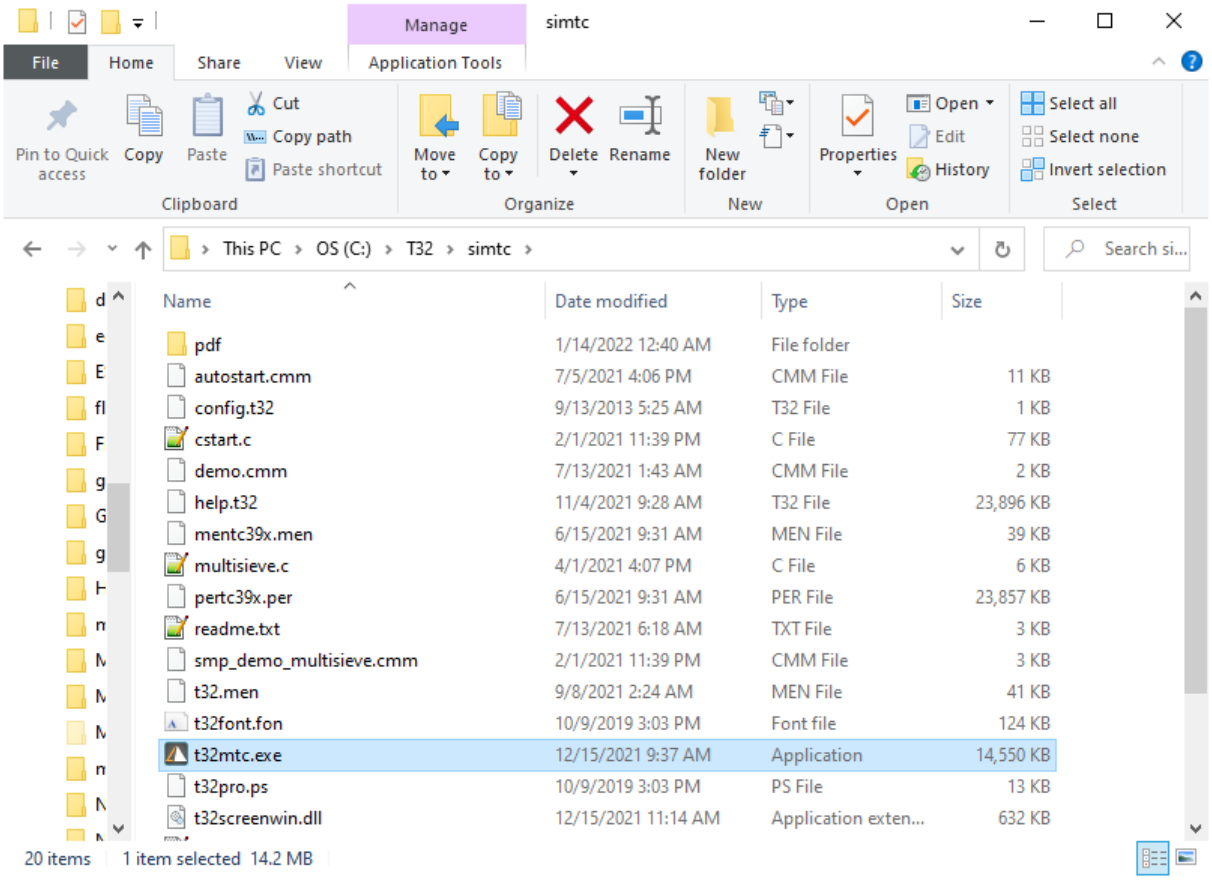
2.2.3 Workflow

The following steps show an example of setting up a VectorCAST Unit Test Environment with the “Hightec Tricore tc39xx Trace32 Simulator C++” RSP.

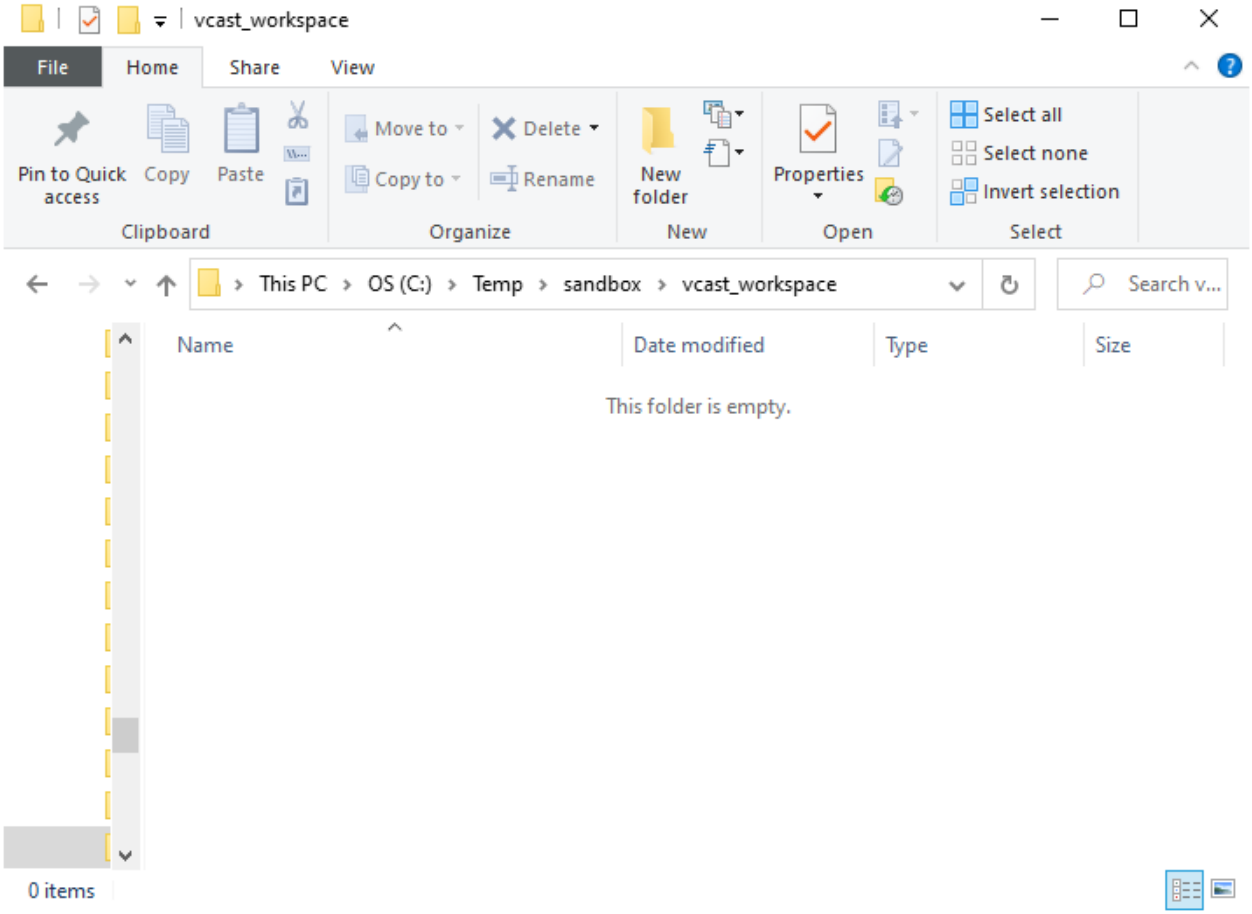
1. Verify HighTec compiler directory and file exist.



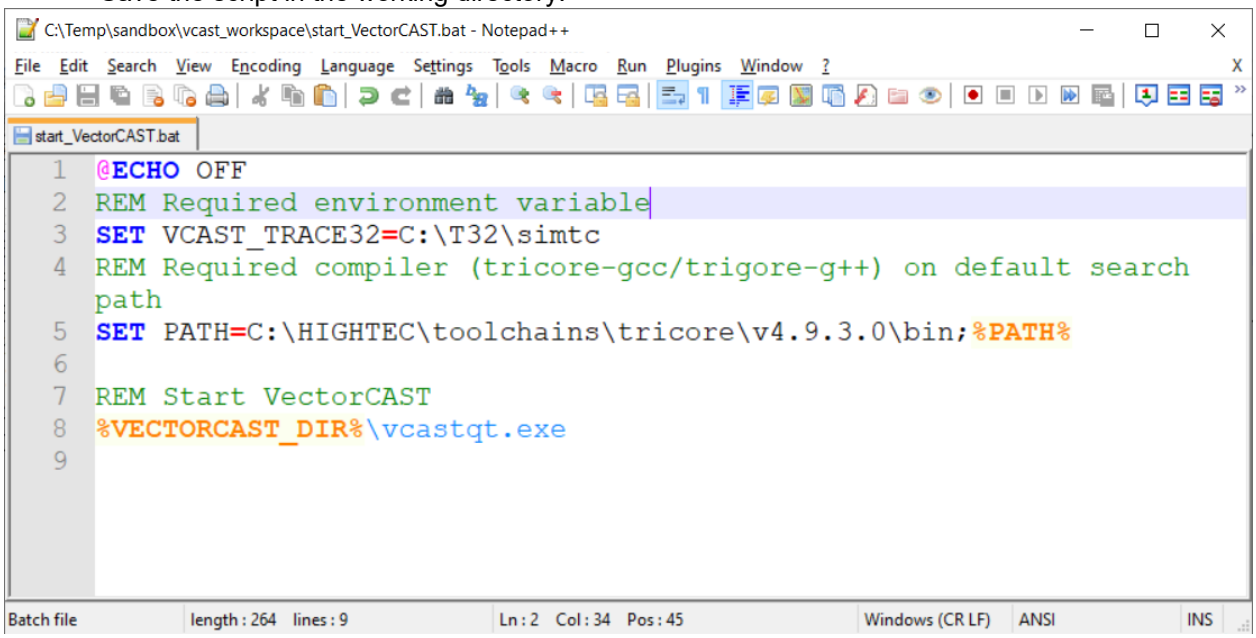
2. Verify Lauterbach Trace32 simulator directory and file exist.



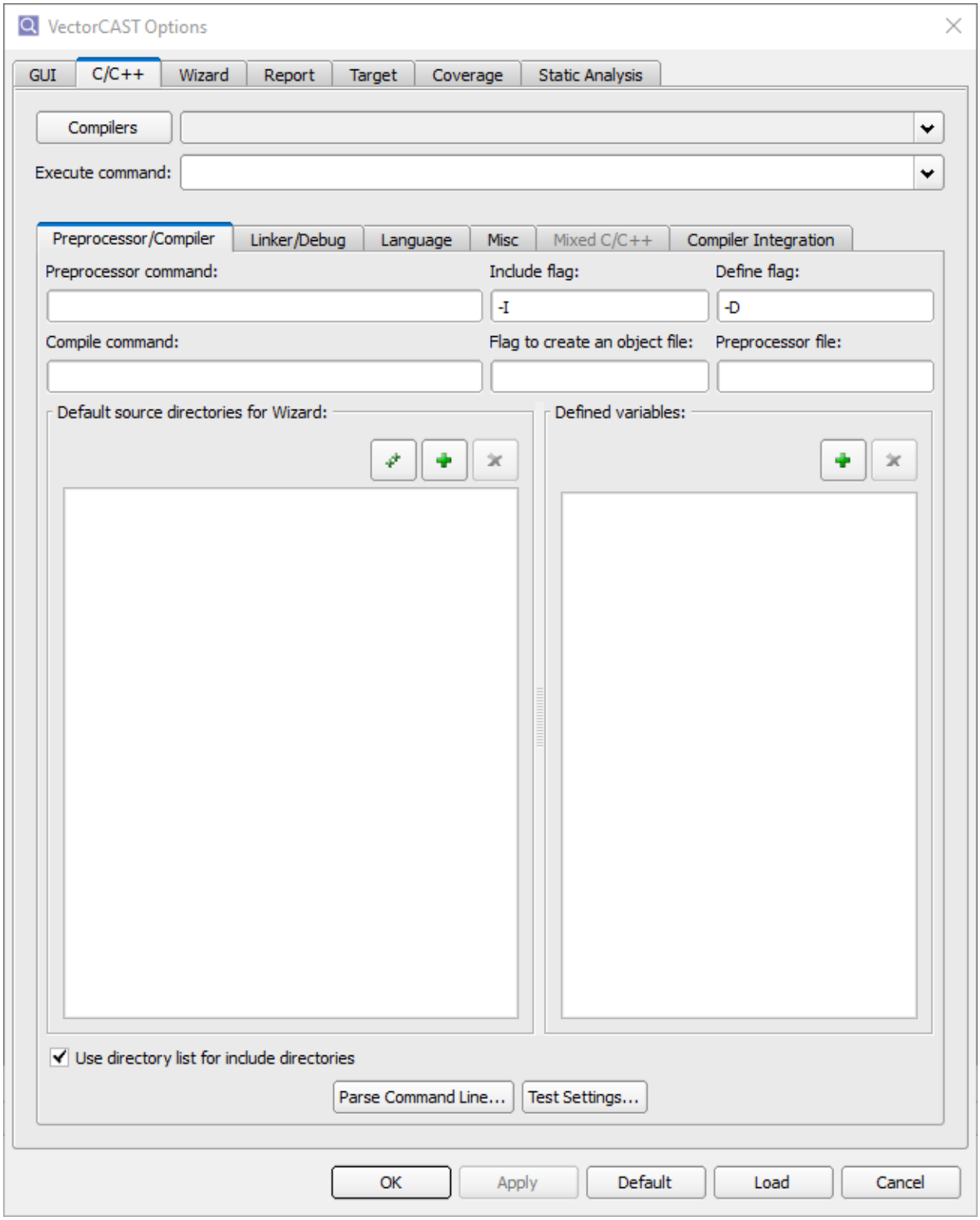
3. Create a working directory to be used as a workspace for VectorCAST.



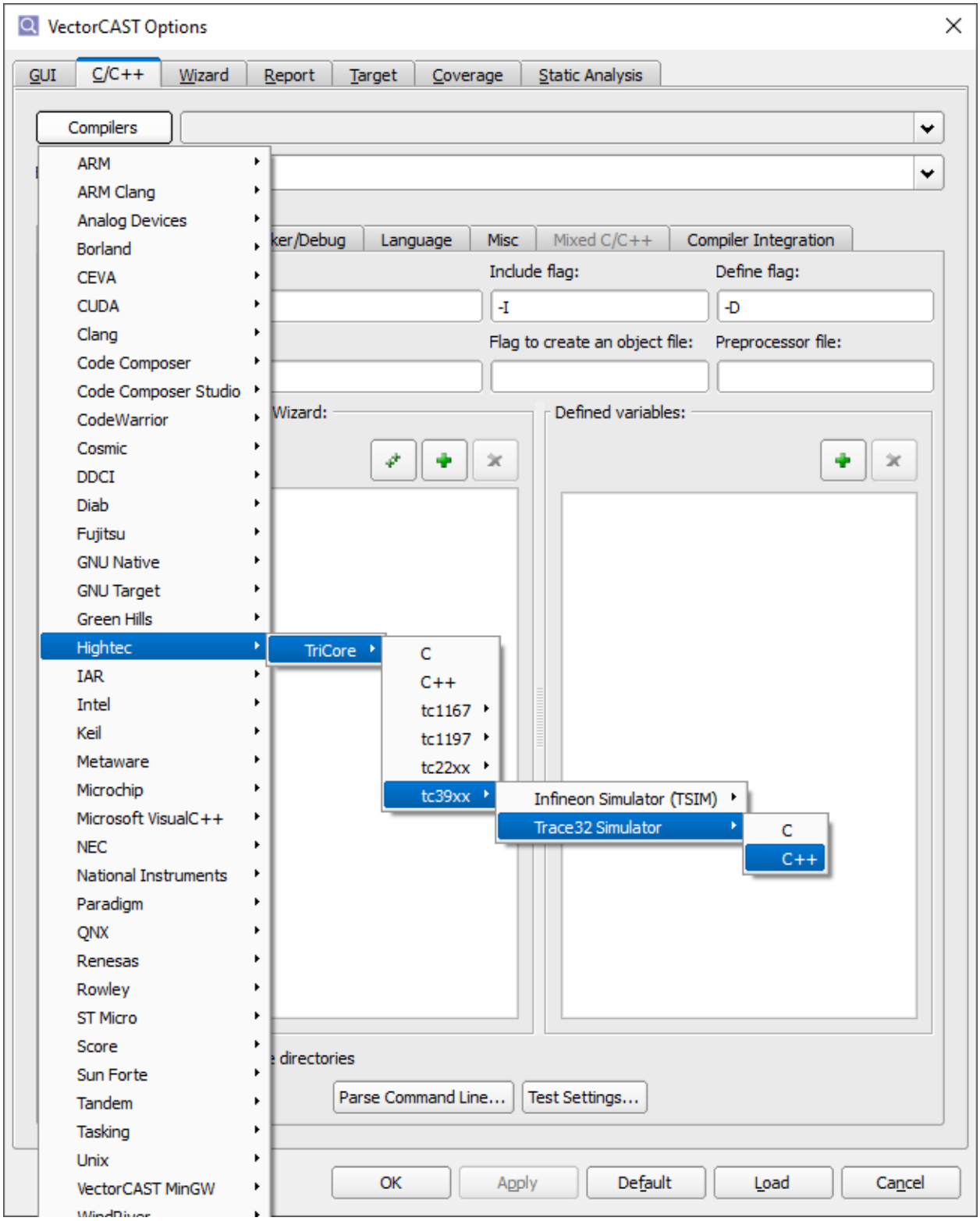
4. Open a text editor. Create a script to initialize environment variables and starts VectorCAST. Save the script in the working directory.



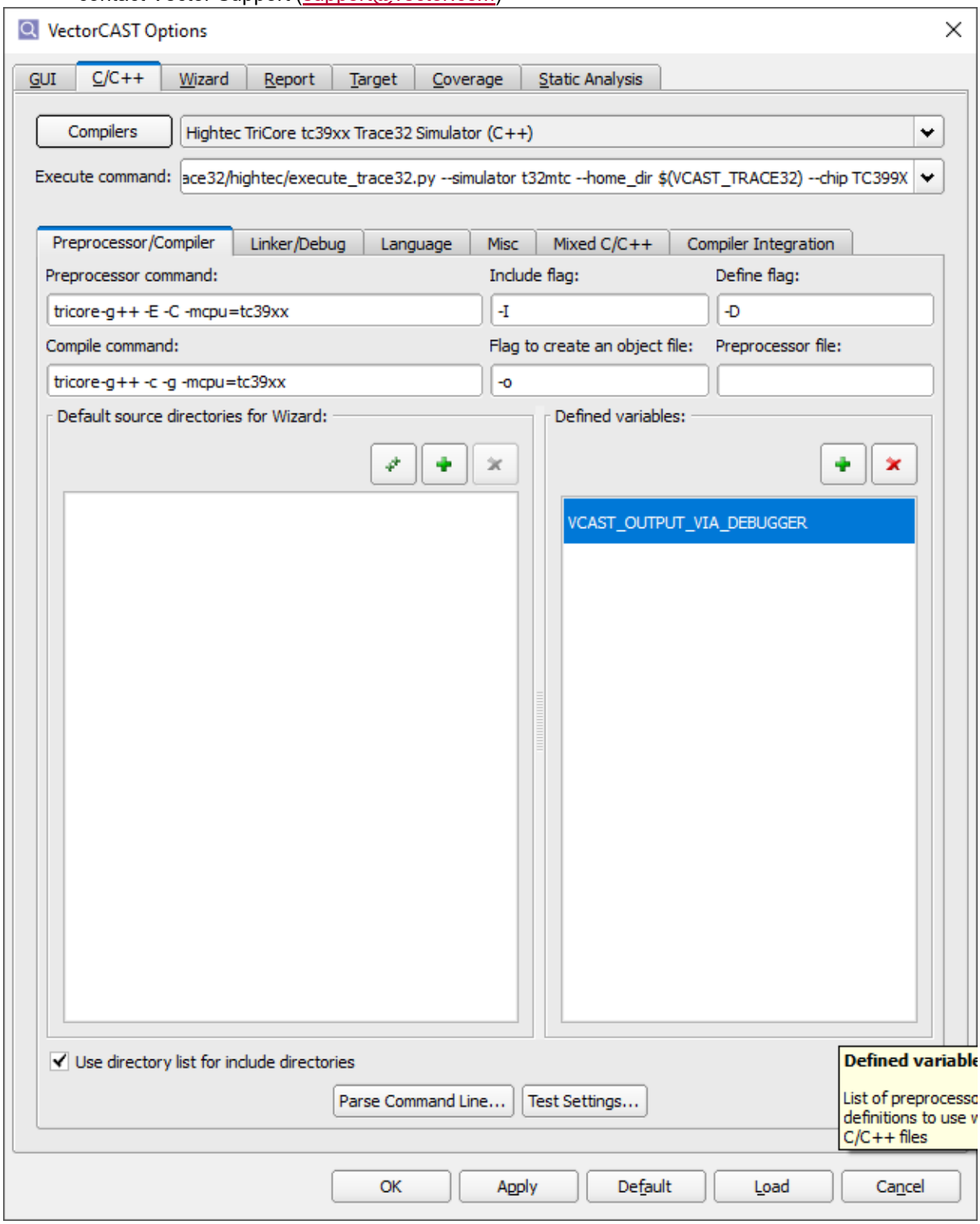
5. Use the newly created script to launch VectorCAST. Open the VectorCAST Options dialog by selecting Tools→Options. Click the C/C++ tab.



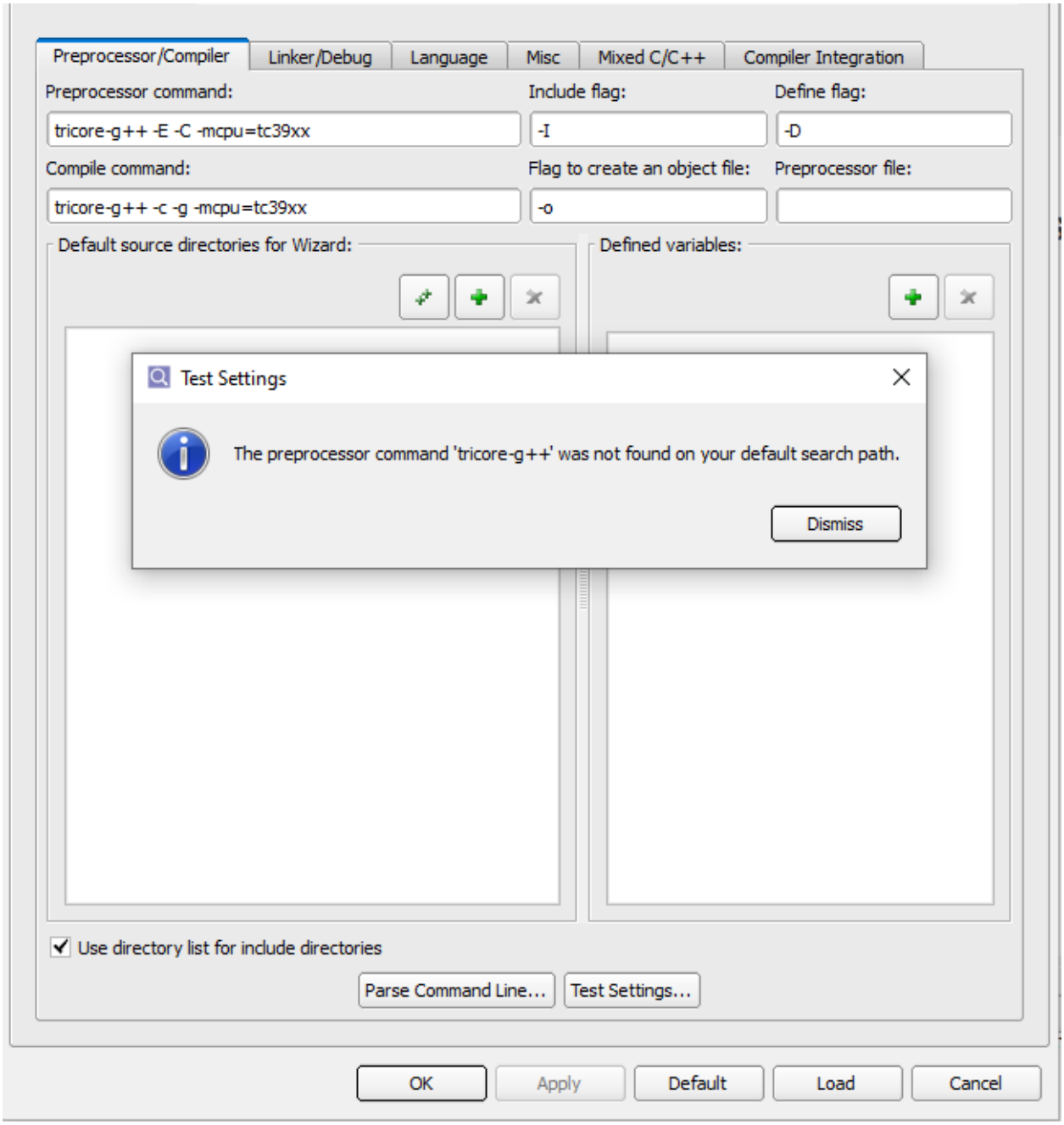
6. Choose the “Hightec Tricore tc39xx Trace32 Simulator C++” RSP by selecting Compilers→Hightec→Tricore→tc39xx→Trace32 Simulator→C++



7. The VectorCAST Options dialog should have updated fields. Click Apply. If an error occurred, contact Vector Support (support@vector.com)



8. Verify the compiler is set up by clicking the Test Settings… button. If an error appears, verify the compiler is on the default search path.



9. Click OK to close the VectorCAST Options dialog.
10. To create a Unit Test Environment, refer to the VectorCAST/C++ User's Guide section "Creating a New Environment." If errors appear while creating the environment, contact Vector Support (support@vector.com).

2.3 Use Case: Infineon Simulator

As part of the HighTec toolchain, the Infineon Simulator (TSIM) binaries are installed. The binary names are as follows:

- tsim
- tsim1311
- tsim16
- tsim16p_e

2.3.1 Setup

Based on the '-mcpu' flag of the compiler, the VectorCAST RSP automatically detects the architecture and the appropriate simulator binary to use. The environment variable VCAST_HIGHTEC_TRICORE_INSTALL_BASE must be set to the installation directory of the HighTec toolchain. It is used to find the following files TSIM configuration files:

<VCAST_HIGHTEC_TRICORE_INSTALL_BASE>/bsp/tricore/common/tsim/<arch>/DCONFIG

<VCAST_HIGHTEC_TRICORE_INSTALL_BASE>/bsp/tricore/common/tsim/<arch>/MCONFIG

<VCAST_HIGHTEC_TRICORE_INSTALL_BASE>/bsp/tricore/common/tsim/<arch>/ICONFIG

<VCAST_HIGHTEC_TRICORE_INSTALL_BASE>/bsp/tricore/common/tsim/<arch>/PCONFIG

The HighTec compiler "tricore-gcc/tricore-g++" must be on the default search path.

2.3.2 Execution Method

VectorCAST uses a Python script to control the Infineon (TSIM) Simulator. As part of the RSP, VectorCAST is configured to capture the coverage and test case results by capturing the standard output (stdout) and writing it to a file.

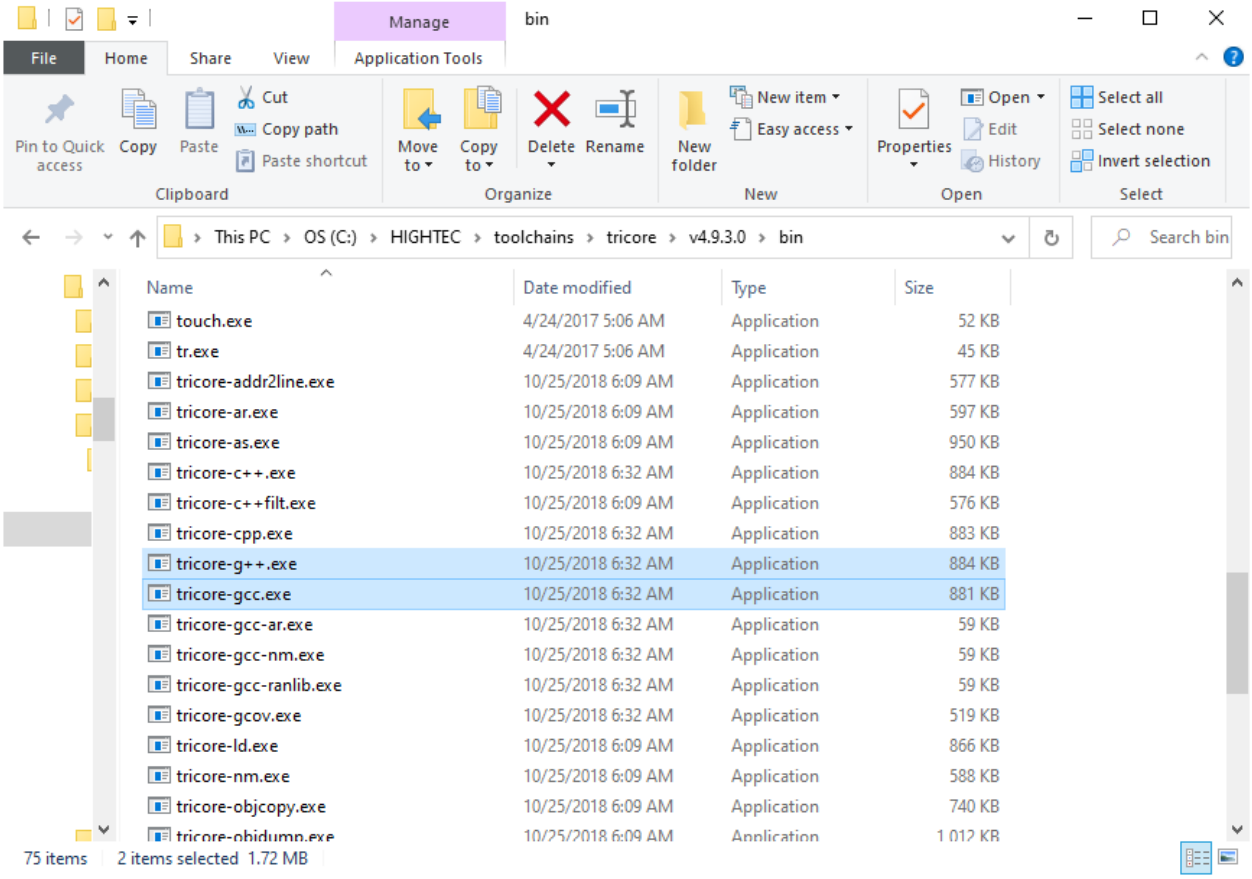
The Python script performs the following tasks:

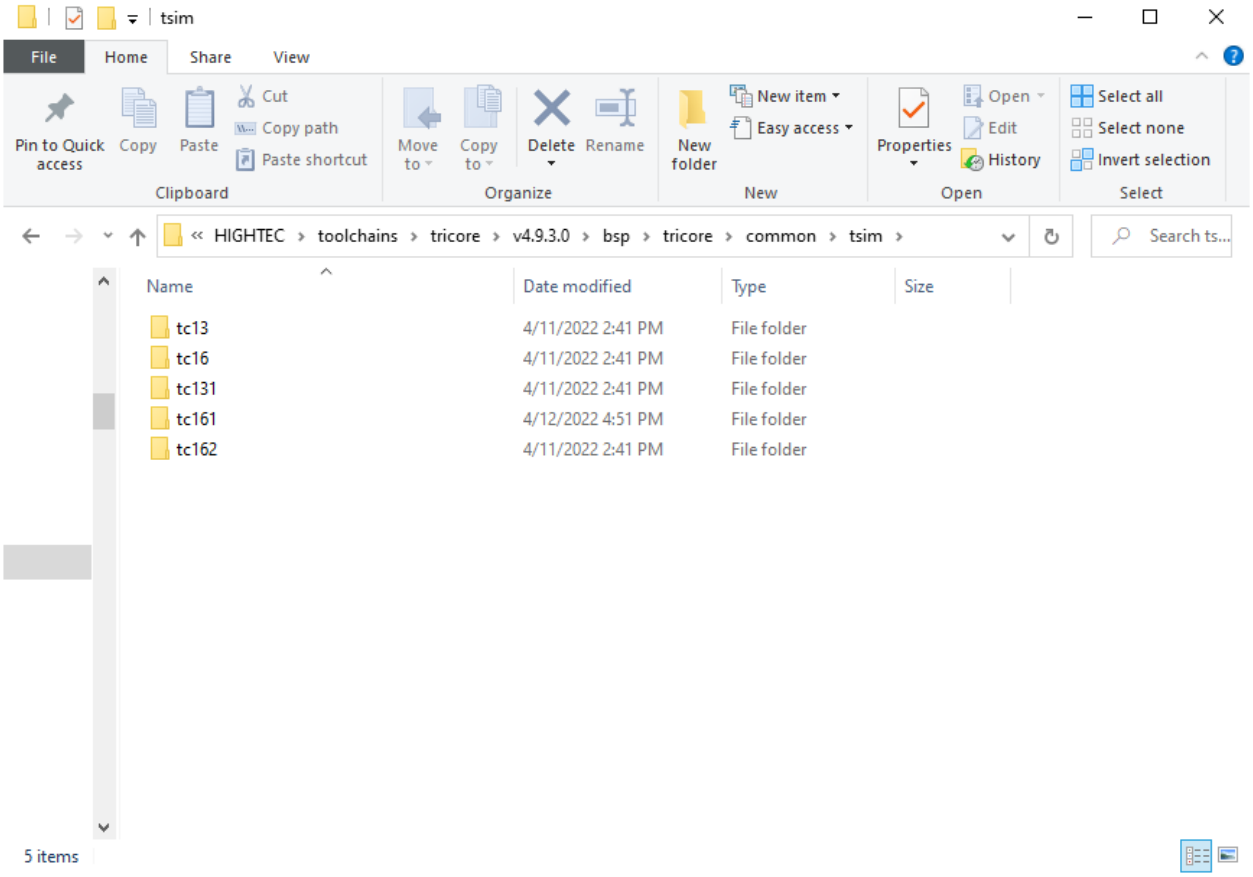
1. Compose a command-line string that uses the TSIM configuration file directories for tsim
2. Execute tsim with the following flags:
   Dconfig, MConfig, IConfig, PConfig: Configuration files for the Simulator
   h: Complete histogram output
   s: Standalone execution
   H: Enable syscall handling for HighTec tools
   disable-watchdog: Disable the running of the watchdog timer
   log-file: Name of the log file
   x: Override the max number of cycles. Value of zero (0) runs forever
   o: Specify input file name

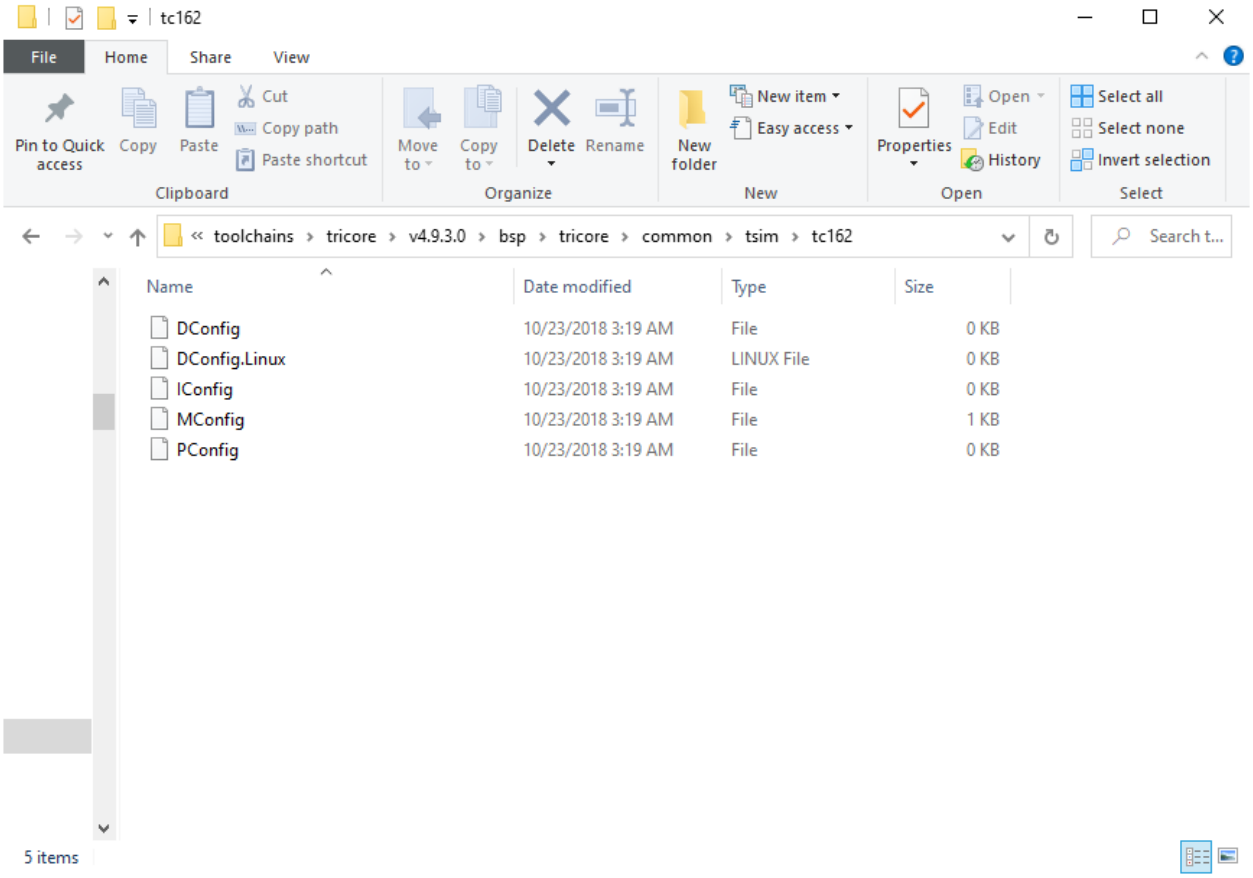
2.3.3 Workflow

The following steps show an example of setting up a VectorCAST Unit Test Environment with the “Hightec Tricore tc39xx Infineon Simulator (TSIM) C++” RSP.

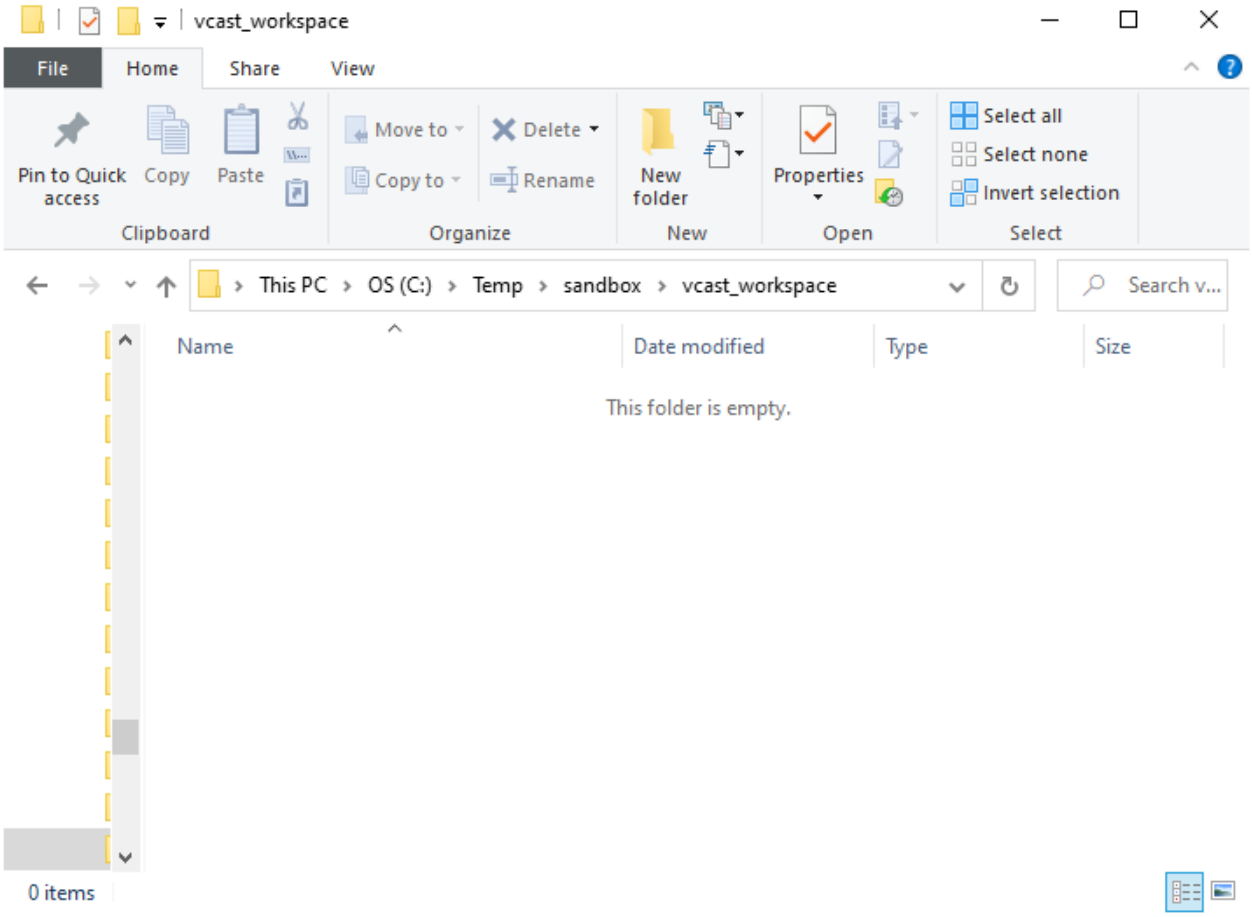
1. Verify HighTec compiler, TSIM configuration directories and files exist.



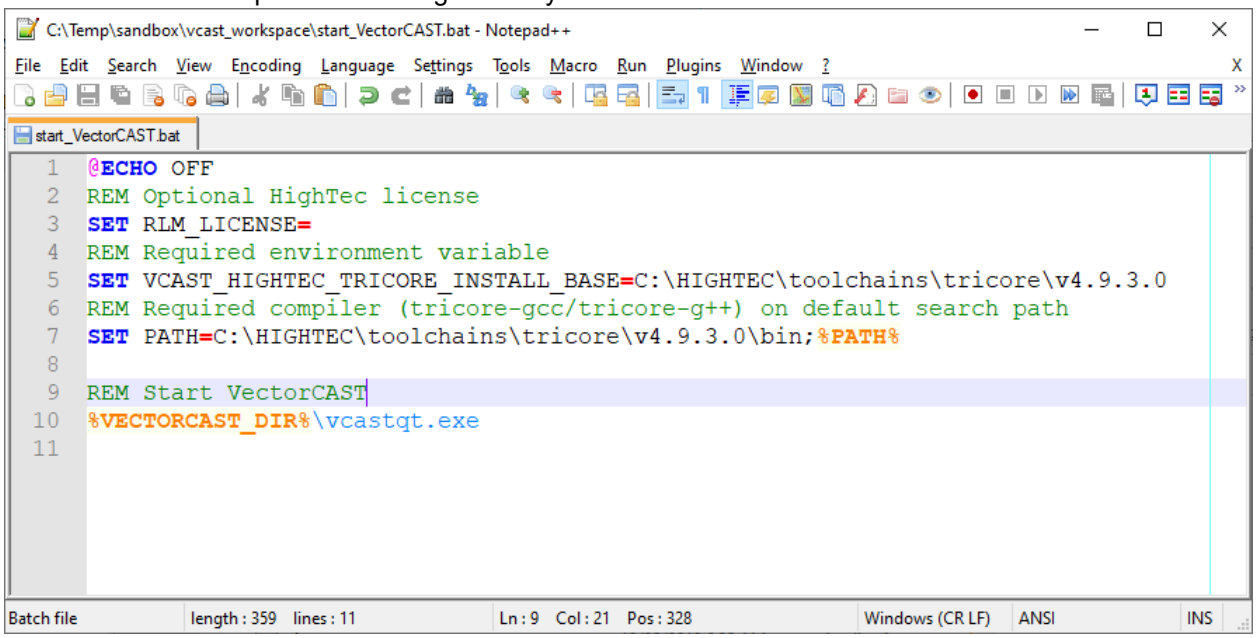




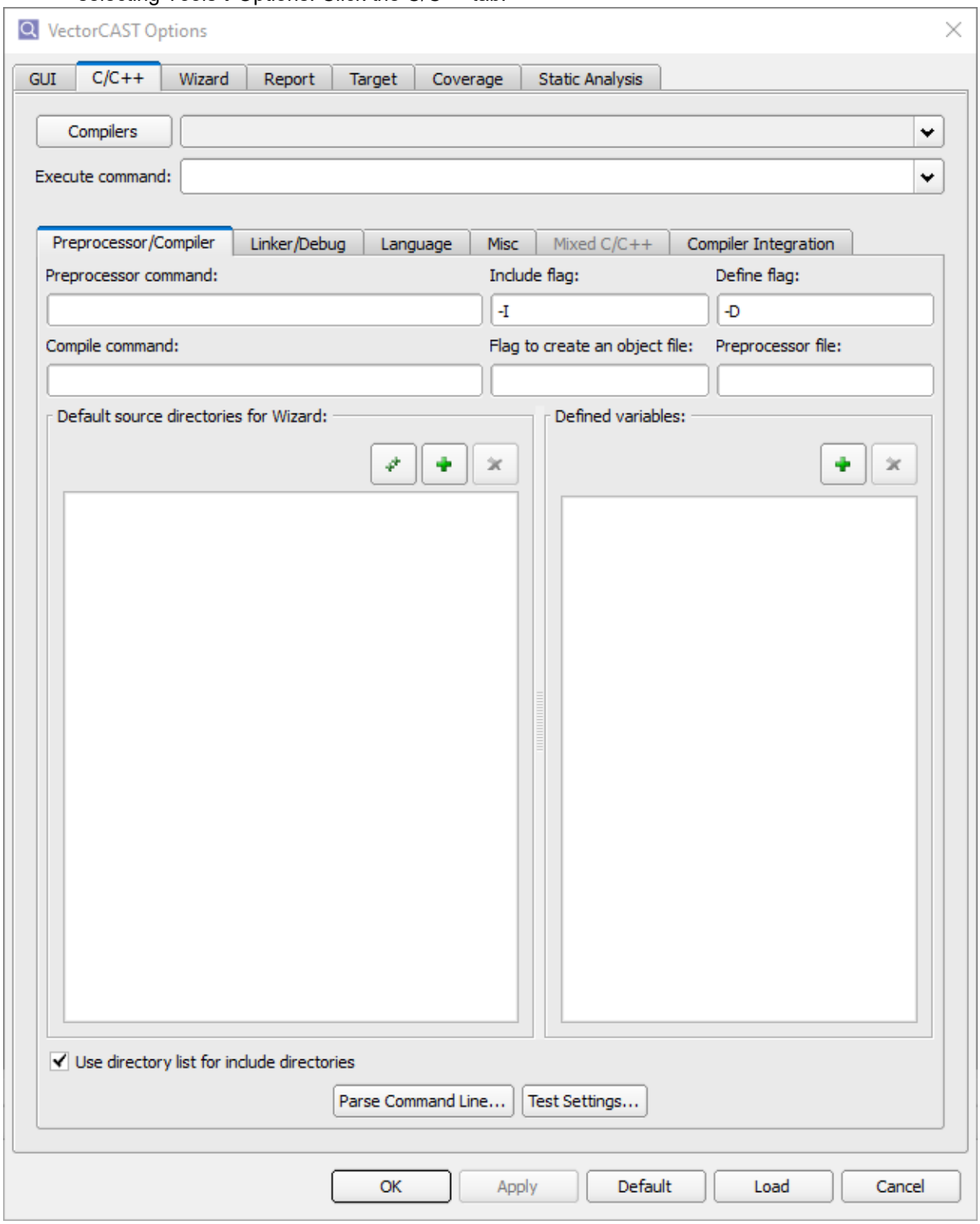
2. Create a working directory to be used as a workspace for VectorCAST.



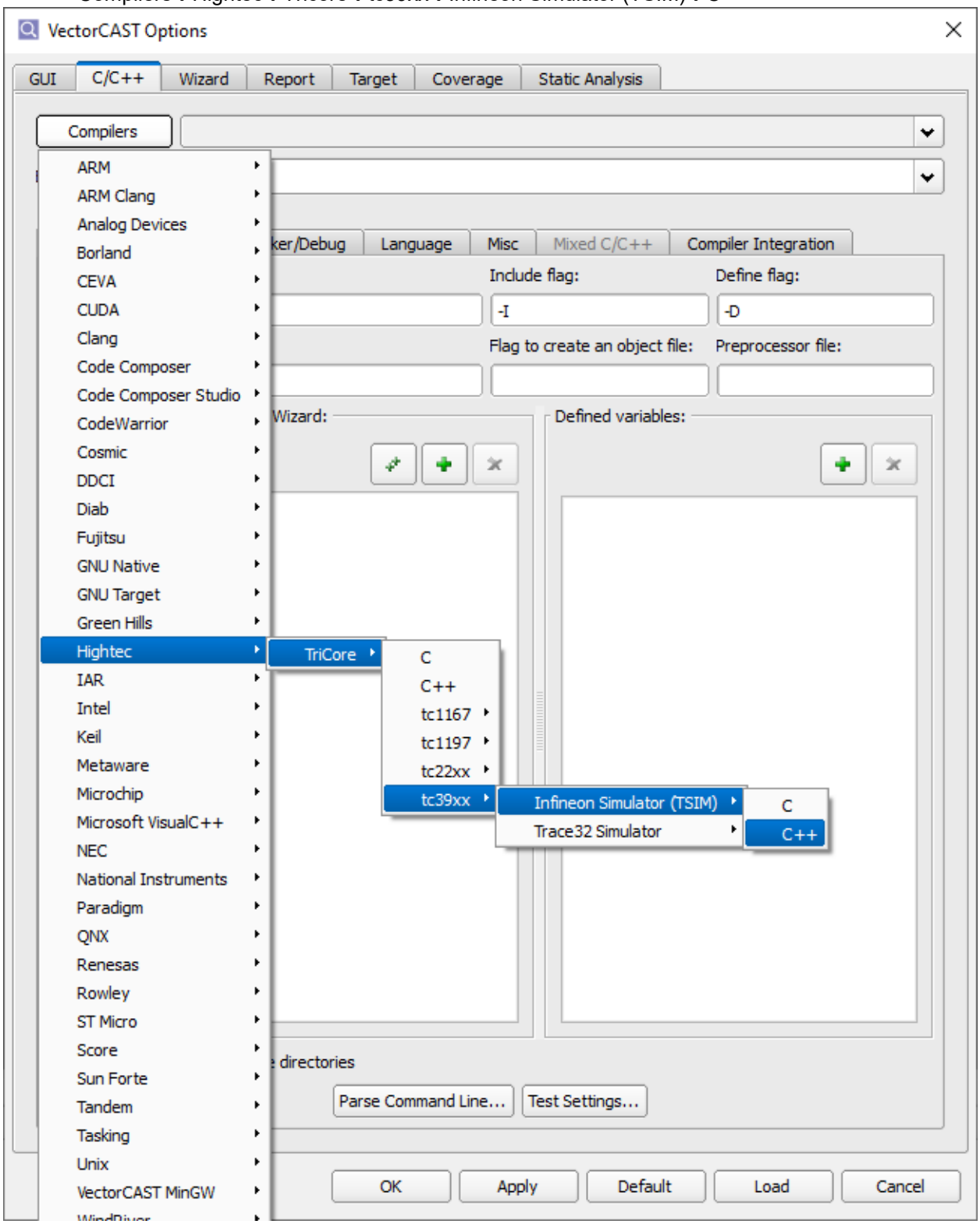
3. Open a text editor. Create a script to initialize environment variables and starts VectorCAST. Save the script in the working directory.



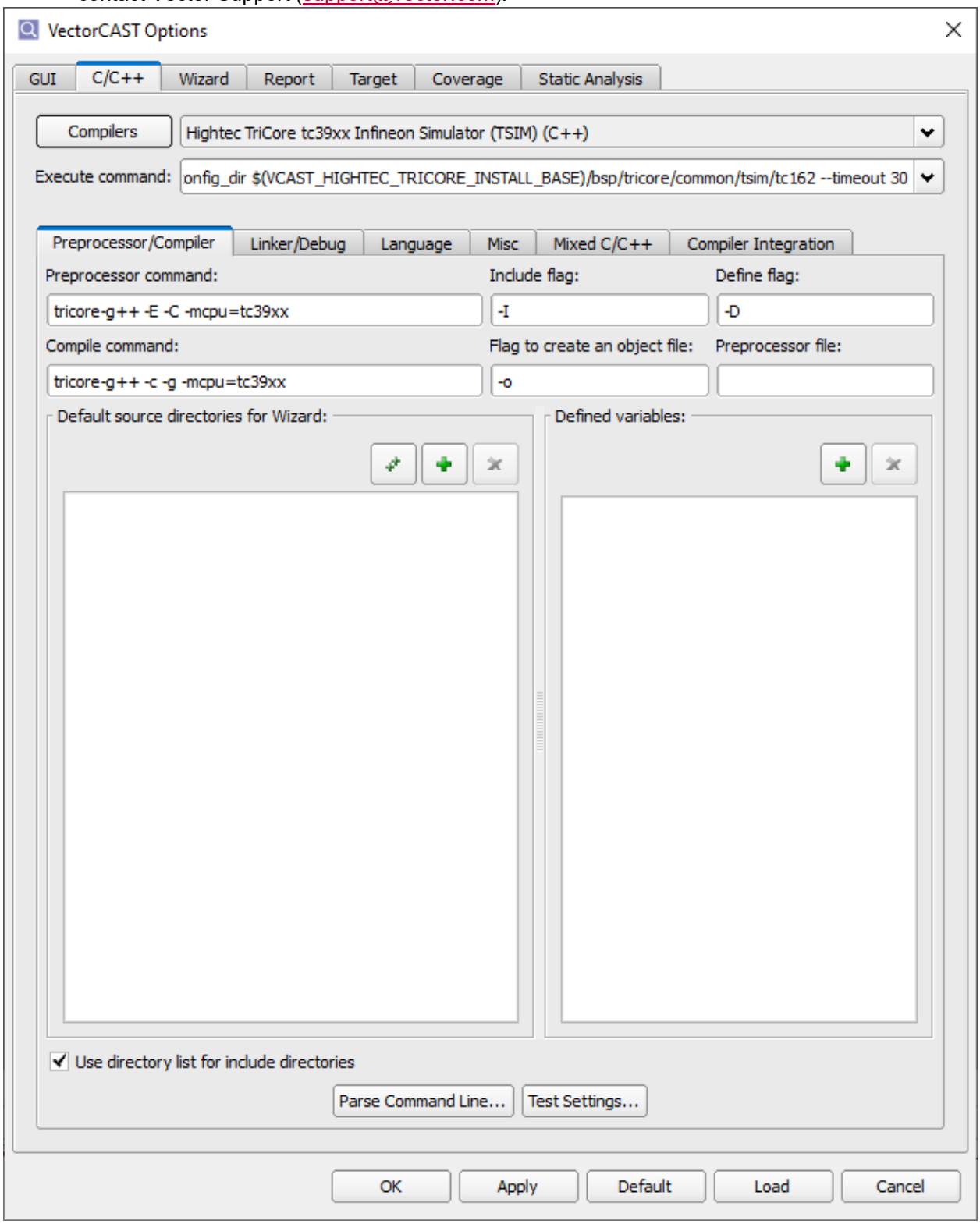
4. Use the newly created script to launch VectorCAST. Open the VectorCAST Options dialog by selecting Tools→Options. Click the C/C++ tab.



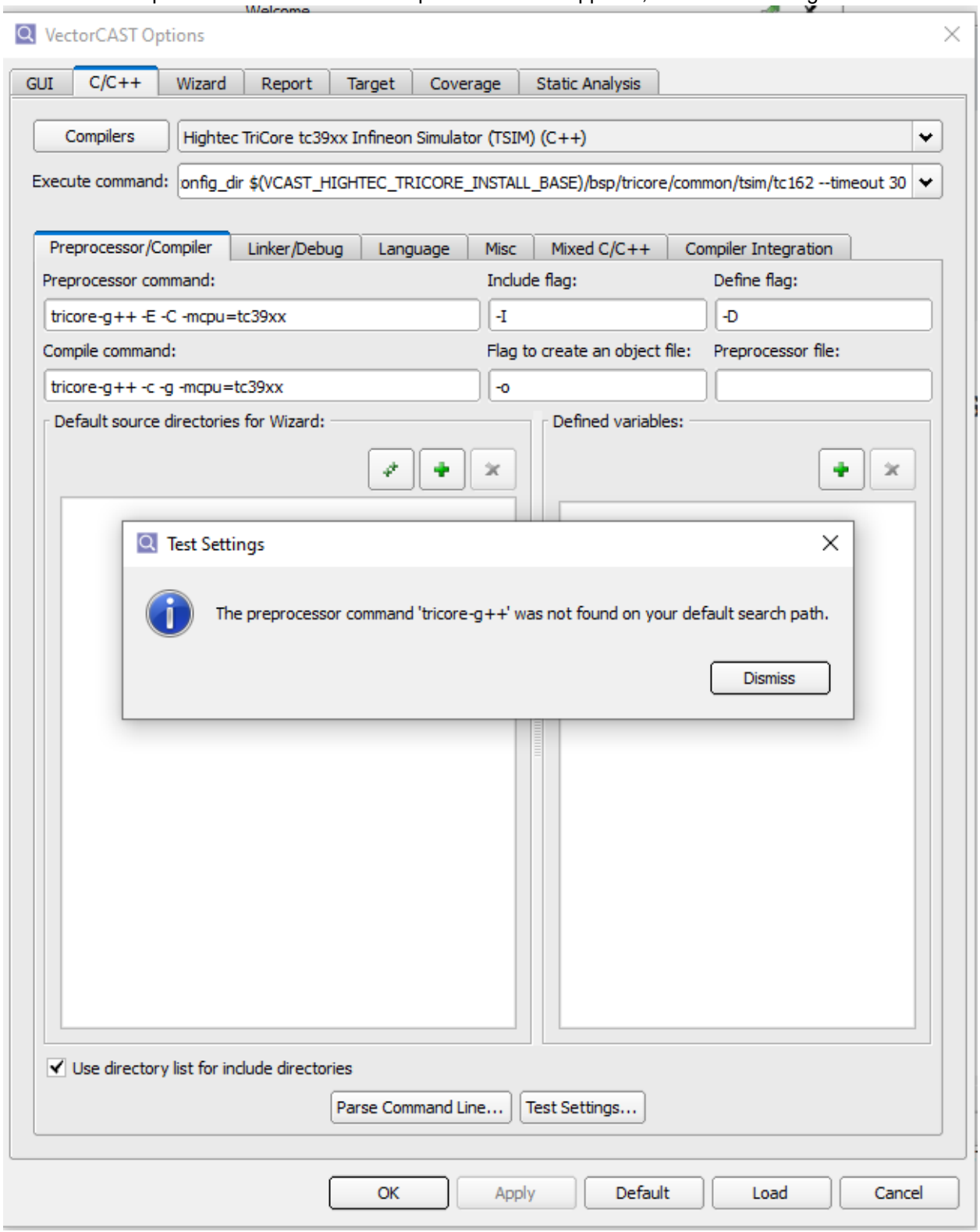
5. Choose the “Hightec Tricore tc39xx Infineon Simulator (TSIM) C++” RSP by selecting Compilers→Hightec→Tricore→tc39xx→Infineon Simulator (TSIM)→C++



6. The VectorCAST Options dialog should have updated fields. Click Apply. If an error occurred, contact Vector Support (support@vector.com).



7. Verify the compiler is set up by clicking the Test Settings… button. If an error appears, verify the compiler is on the default search path. If no error appears, dismiss the dialog



8. Click OK to close the VectorCAST Options dialog.
9. To create a Unit Test Environment, refer to the VectorCAST/C++ User Guide section “Creating a New Environment.” If errors appear while creating the environment, contact Vector Support (support@vector.com).

3 Additional Resources

- https://hightec-rt.com/en/
- https://www.lauterbach.com/frames.html?home.html

4 Trademarks

All mentioned names are either registered or unregistered trademarks of their respective owners.

5 Contacts

For a full list of all Vector locations and addresses worldwide, please visit http://vector.com/contact/.