VECTOR

VectorCAST DDCI/Deos Integration

Version 1.1

2023-07-19

Application Note AN-ACT-1-103

Author Craig Pedersen

Restrictions Public Document

Abstract Information on DDC-I/Deos 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 describes the integration between VectorCAST and the DDC-I Deos/QEMU Simulator. It describes how to configure an OpenArbor Project that contains the QEMU Simulator for the Deos Operating System, along with a place holder for the VectorCAST test harness. In addition, it describes how to build the Test Harness in VectorCAST and download it to the QEMU Simulator.

2 Instructions

2.1 Notes

- It is assumed that the reader is familiar with VectorCAST and has at least some experience with building and executing test cases in general.
- Knowledge of the DDC-I/Deos environment, the QEMU simulator, and the Open Arbor IDE is not required, but would certainly help. It is suggested to get familiar with this environment and tools outside of VectorCAST before integrating with VectorCAST. The instructions here are written for an initial project creation and should work as described. However, when re-opening the project or changing the project, knowledge of the Deos development environment will be beneficial.
- The instructions are written for a Windows host, but would be similar for Linux
- Official Deos support was introduced in VectorCAST 2021sp6. Prior versions will not work. This app note was developed using VectorCAST 2023 but should basically be the same for 2021 or 2022.
- The QEMU Simulator initializes and creates its own subnet, which could potentially cause networking problems on your workstation:
  - If you are using a VPN, you may lose your wireless connection when the simulator starts up. Some VPNs have a setting to disable WiFi when a LAN is connected. If this happens, disable this setting.
  - In addition, you may need to disable Firewall Protection to connect to the Simulator.

2.2 Setup

- It is required that the DDC-I/Deos package, OpenArbor, and VectorCAST are fully installed and licensed.
- VectorCAST needs to be launched from a DDCI shell and requires some environment variables to be set before invocation. The following example batch file can be used. Note that the compiler version can be found here: `%VCAST_DDCI_INSTALL%\usr\local\cross-compilers\`. Also note that comments are contained within brackets and should be removed if you are copying this into a batch file:

  If you have difficultly with this batch file, please contact Vector Customer Support or your local Field Application Engineer (FAE) (see the section 5 below).

```
rem set VECTORCAST_DIR=C:\VCAST\ [This should be done outside of this batch file]
set DESK_IP_ADDR=192.168.19.101  [IP Address of the QEMU Simulator]
set VCAST_DDCI_INSTALL=c:\ddc-I  [Location of the DDCI installation]
set DESKHOME=%VCAST_DDCI_INSTALL%\desk
set VCAST_DDCI_COMP_VER=7.3.0    [DDCI Compiler Version]
[path for the gdb debugger.  Take out “rem” if you need the debugger]
```

```
rem PATH=%VCAST_DDCI_INSTALL%\usr\local\deos-debuggers\gdb-8.0.1\bin;%PATH%
[Command to start up VectorCAST from the DDC-I shell.  The command should all be
on one line:]
start/min %VCAST_DDCI_INSTALL%\bin\env.exe /bin/python /desk/desk-environment.py
%VECTORCAST_DIR%\vcastqt
```

2.3 Overview of the Deos/QEMU Environment

The execution platform for this integration is the QEMU Simulator. QEMU supports a variety of microprocessors and here we will be using PPC. DDC-I has created an OpenArbor example platform that includes the Deos operating system along with an example ”UDP-vs-TCP” application, which runs on the QEMU PPC simulator. This platform will be used as a container to run the test harness. To get this to work, the name of the test harness executable needs to be the same as the name of the example application created in OpenArbor. As you will see in the Configuration Instructions below, we've chosen to rename the executable to “vcast”, just for clarity. However, we will still refer to the application downloaded from OpenArbor as “UDP-vs-TCP” and the executable downloaded from VectorCAST as the “test harness”, just to distinguish the difference. The only requirement is the executable names need to match (i.e., “vcast.exe”).

Note that when the example platform is created in OpenArbor, there is separation between the OS (Deos) and the UDP-vs-TCP application. They are combined and downloaded when starting up QEMU through the Target Manager. For the VectorCAST integration, that is all that needs to be done. After the harness is built in VectorCAST, it will get downloaded to Deos and replace the UDP-vs-TCP Application.

2.4 Configuration Instructions

The configuration steps are as follows:

2.4.1 Creation of a Deos Kernel & Application Project using OpenArbor

- Invoke the OpenArbor IDE
- Create a new Workspace that will contain all your work
  - File->SwitchWorkspace->Other
  - Dismiss the Welcome window as it may obscure all other windows
- Do File->New->DDC-I Deos Quickstart
  - Select qemu-ppc as the Deos reference platform and keep “qemu-ppc” as the Project Base Name. Note: Do not change this name! Otherwise, the target server will not launch properly.
  - Make note of the IP address or change it to match the DESK_IP_ADDR, which is noted above for the start-up batch file.
  - Select Next
  - Expand socket
  - Select udp-vs-tcp
  - Select Finish
- By default, the project is built automatically in the background. You can disable that from the Project pulldown window if you desire
- Add udp-vs-tcp as a dependency of qemu-ppc, as follows:
  - In the DDC-I Project Window, expand qemu-ppc->Deos Component and right click on Dependencies and select AddDependency and then select udp-vs-tcp in the pull-down list. Only load the dependency to the platreg register.
  - Select OK

2.4.2 Change the executable name

VectorCAST will try to download and replace an executable application named “vcast.exe”. However, the name that we created above is “udp-vs-tcp”. This name appears in 2 places and needs to be changed as follows:

- In the Project Window, right click on the “udp-vs-tcp” node and rename it to “vcast”.
- Open the vcast/config/udp-vs-tcp.pd.xml file and find ProcessTemplate->executableName and change that name to vcast.exe
- If you need to debug the vcast executable, change processTemplate > breakAtStartup to yes
- Save the changes
- In the vcast/config folder there may be alternate udp-vs-tcp.pd.xml files ending in ~1~ and ~2~. Those files can be removed

2.4.3 Add the Library Dependencies

The VectorCAST test harness requires certain shared libraries when it executes. Adding the following dependencies will result in those libraries being loaded with Deos when the QEMU emulator boots up.

- In the Project Window, go to vcast->DeosComponent->Dependencies and add:
  - math
  - gnu-language

2.4.4 Add app_main.cpp to the udp-vs-tcp project

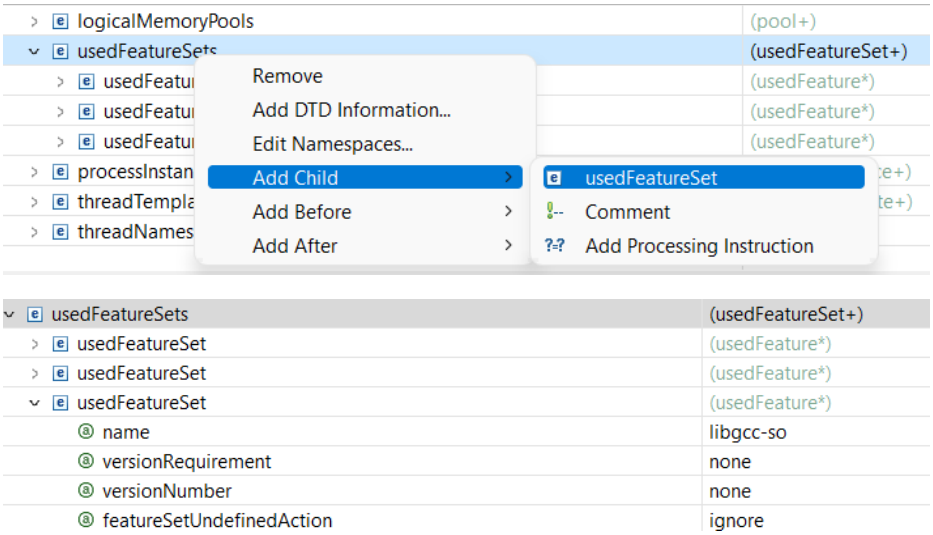
The VectorCAST test harness has a main() function with supporting functions to allow the harness to operate correctly. Replace udp-vs-tcp.cpp with $VECTORCAST_DIR/DATA/ddci/deos/app_main.cpp. This file should compile and run with the existing configuration.

- RightMouse on vcast/code and do New->File and add this file: $VECTORCAST_DIR/DATA/ddci/deos/app_main.cpp
- RightMouse on vcast/code/udp-vs-tcp.cpp and select Delete

2.4.5 Project Configuration.

From the vcast node in OpenArbor, open the config/udp-vs-tcp.pd.xml file and make the following changes:

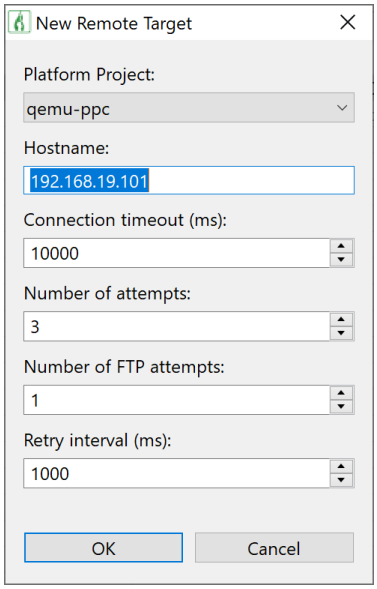
- `processTemplate:`
  - `argumentQuota: 30`
- `logicalMemoryPools:`
  - `pool->pagesNeeded: 128`
- `usedFeatureSets: - Add libgcc-so to a new usedFeatureSet from (design view)`



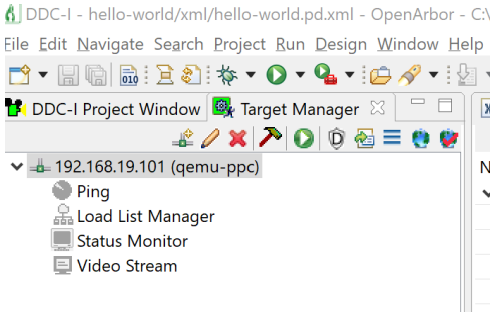
- `processInstances->processInstance`
  - `envpCommandLineParameters: MALLOC_PAGES=100`
- `threadTemplates > threadTemplate name=”mainThreadTemplate”`
  - `stackSizeInPages: 16`
- Save the changes

2.4.6 Target Server Configuration

- Now select the Target Manager Tab (if that Tab is not present, go to Window->ShowView->Other->DDC-I Views->TargetManager). In the Target Manager window, you should see the qemu-ppc simulator node. If it is not present, right click in the Target Manage panel and do New RemoteTarget. Verify the IP address and click ok.



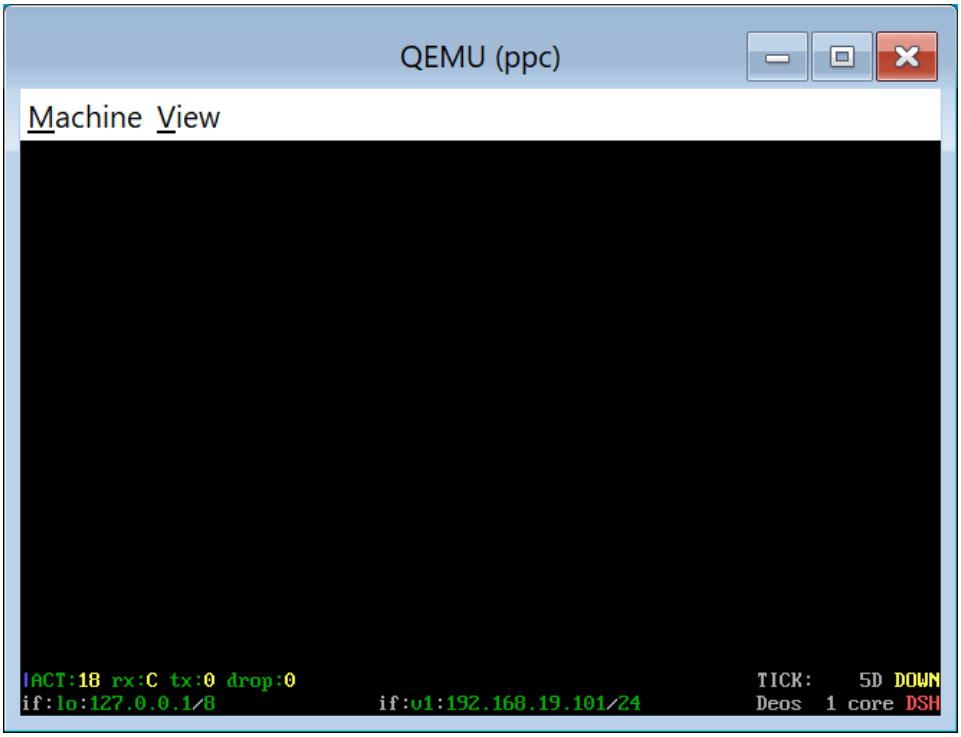
> The result should look like this:



> Important Note: When you launch the simulator in the next step, it will trigger a rebuild of the project if there are any outstanding changes. The build details are printed in the DDC-I-Console window. That window is typically in the bottom right pane. If you do not see that window, you can reopen it from Window->ShowView->DDC-I-Console. Because of auto-building, it is sometimes possible to miss a build error. When an error occurs, the system may revert to the previous build. If for any reason things don't appear to be working properly, you may want to clear the window and then manually build the "qemu-ppc" and "vcast" nodes in the project window, just to make sure you did not miss any errors.

> Go to the Target Manager window and right click on the qemu-ppc node and select Launch Emulator

- This may take a minute as the project may need to get rebuilt.
- Eventually you will see the QEMU (ppc) console, which should look like this:



> The Kernel and the UDP-vs-TCP App should now be downloaded, but the app will not automatically start. Go back to the Target Manager window and right click on the qemu-ppc node and select UpdateTargetLoad. This will start up the UDP-vs-TCP application. If you do not see any output, you may have a networking problem. Review the Notes section 2.1 above. If everything is done correctly, the emulator window should now look like this screen shot. If not, VectorCAST will probably not work.

2.4.6.1 Testing connection using telnet

Once you have the replaced the udp-vs-tcp.cpp file with app_main.cpp, you can test the project using telnet from a DOS command shellQEMU:

- telnet 192.168.19.101
- enter any character
- The output should look something like this:

Go ahead and kill the target server by simply closing the window. You should now be ready to run tests within VectorCAST.

2.4.7 Creating a VectorCAST Test Environment

For Deos support, VectorCAST needs to be launched from the DDC-I shell (using the batch file as described in the Setup section 2.2 above). From VectorCAST, set a new working directory to contain your work. Note that this working directory is different than the workspace you created for OpenArbor. Now create a test environment using the DDCI->Deos->PPC->C++ compiler template with a known C/C++ module (we suggest %VECTORCAST_DIR%/tutorial/cpp/manager.cpp). Note that the C++ template supports both C & C++ code.

Next, we need to relaunch the target server and also press the UpdateTargetLoad button. Once the simulator is running, you should now be able to run a test from VectorCAST. In the QEMU shell, you should see some different output, indicating that the VectorCAST test harness is running. The test will take a few seconds to initiate, so be patient.

Once the test is complete, you should then have control back in VectorCAST and you will hopefully see that the test completed successfully. If this is all successful, you should now be ready to build an environment against your proprietary code and start running tests.

2.5 Subsequent Usage

The configuration above should be a one-time effort. Certainly, you can go back and reconfigure anything as needed, but for the most part, the configuration is static. After you have run a successful test case and then want to go back at a later point in time, here is all that needs to be done:

- Open up OpenArbor with your prior Workspace
- Open the Target Manager window
- Right click on the qemu-ppc node & select Launch Emulator
- Select the TargetUpdateLoad button
- Open up VectorCAST with your prior Test Environment
- Run the tests
- Note that the target server does not need to be restarted between tests. It should run in perpetuity.

3 Additional Resources

https://www.ddci.com/products_deos_do_178c_arinc_653/

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/.