

Using Renesas GDB Server

Abstract

This document describes the usage of the Renesas GDB server as target debugger. At the time of writing this document only Renesas RX was supported. The latest update of this adaption required a split into simulator and emulator. So, please choose your TEE environment appropriately.

Important Note: You need a **functional** e2 studio development project which can successfully build a target binary and launch a debug session.

Please note: The Renesas GDB server adaption does not support interactive debugging features when executing tests with TESSY. (See 4 to learn how to debug interactively having your test data statically built into the target binary.)

Table of Contents

Abstract	1
1 Introduction.....	2
2 TESSY Environment Settings.....	2
3 Parallel Built and Parallel Execution	3
4 Interactive Debugging.....	3
5 Troubleshooting.....	6

1 Introduction

Since the latest update of the Renesas e2 studio, the procedure for the simulator and for the emulator differs. The simulator needs only the GDB client, while the emulator needs both, the GDB client and the GDB server. In order to utilize the Renesas GDB server, a functional e2 studio project is required, which will provide the command line arguments for the GDB server. When starting the debugger from within e2 studio, the GDB server command line is written into the console view of e2 studio. Please ensure that you can debug your development project with your installed version of e2 studio. This will assure that the connection to the hardware is properly set up.

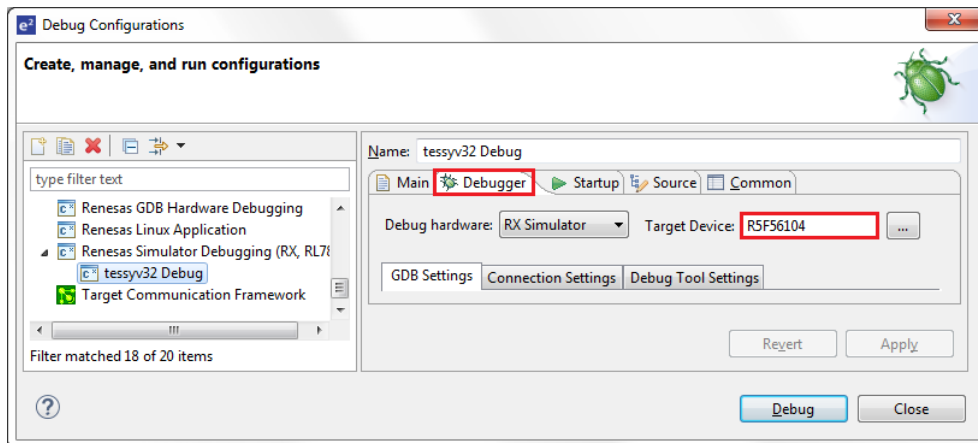
The communication between TESSY and the Renesas GDB server is based on GDB client and GDB server as backend. The Renesas GDB server is started by TESSY by executing the command line found in TEE attribute GDB Server Debugger. The port number is chosen by the Renesas GDB server itself, i.e. please do **not** use **GDB Server Port**. The Renesas GDB server outputs its port number on command line, which TESSY's master reads and uses to connect. TEE attribute **Port Trigger** holds the unique string, which precedes the port number directly. You should not need to change the value.

Furthermore, for the emulator you will have to adjust the contents of the GDB script file TEE attribute Init Script points to. Please alter the access width accordingly. You will find the correct values either from your hardware documentation or you may fetch them out of the **gdb trace** console window of the currently debugging e2 studio. TEE's default settings of the **Renesas Emulator** adaption can be utilized for an RX62N8 board carrying an R5F562N8BDFP processor.

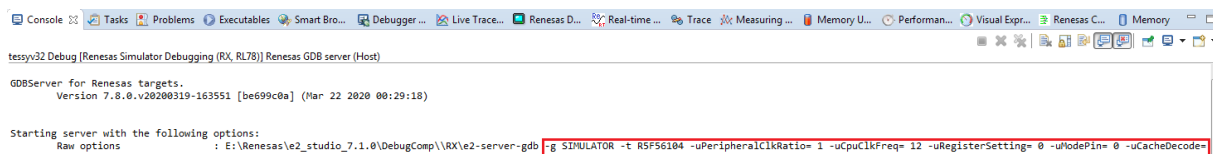
In order to debug the test application interactively with the test case values provided from TDE you need to rebuild the test application in a special mode, i.e. the input values will be compiled into the application. You may then download the test application using the e2 studio debugger and step through the test cases (see chapter 4). Alternatively, you could also setup a TEE environment for e2 studio (refer to application note *Using Renesas e2 studio Debugger*) and, if needed, switch to this TEE environment for the test object in question in order to debug interactively. However, please keep in mind, that the different environments are using different Makefile Templates. So, if you have setup your own Makefile Template for the **Renesas Simulator** or for the **Renesas Emulator**, you will have to setup up another one based on the Makefile Template of the e2 studio TEE environment.

2 TESSY Environment Settings

This chapter discusses the specific Renesas debugger TEE attributes only. Please refer to *chapter 2 of application note Using Renesas e2 studio Debugger* for the specific Renesas RX compiler TEE attributes. The best way to start is to set the **Compiler Install Path** and the **Target Install Path** and check what errors remain after toggling the **Show Errors/Warnings** toggle button, which is found in the **Attributes** view's toolbar. Please adjust all remaining unresolved paths being displayed. Next, adjust TEE attribute **Debugger Target**, which you can find within e2 studio's debug configuration



and check the **Renesas Debugger Options**, which you retrieved from the e2 studio console at debugger startup.



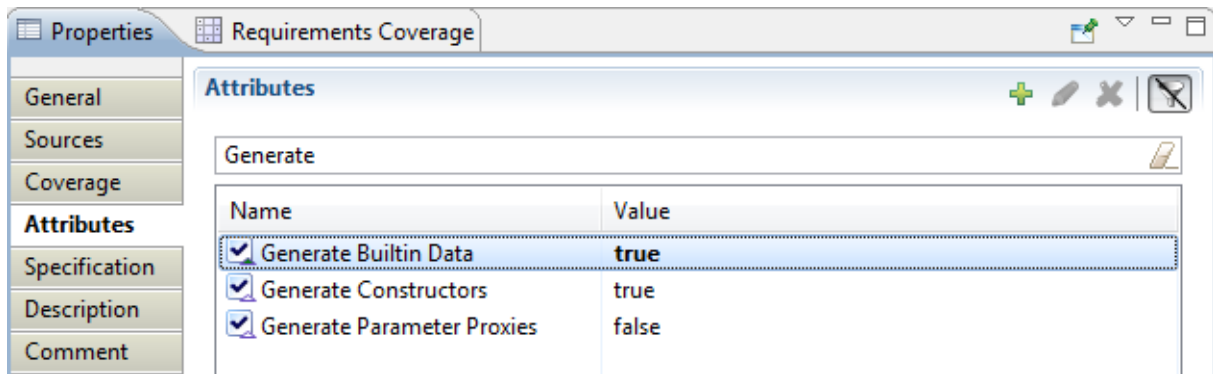
Finally, if you are using an emulator, save the I/O access width data, which you have collected as described in the introduction, into the GDB script pointed to by TEE attribute **Init Script**. Substitute the default values found there.

3 Parallel Built and Parallel Execution

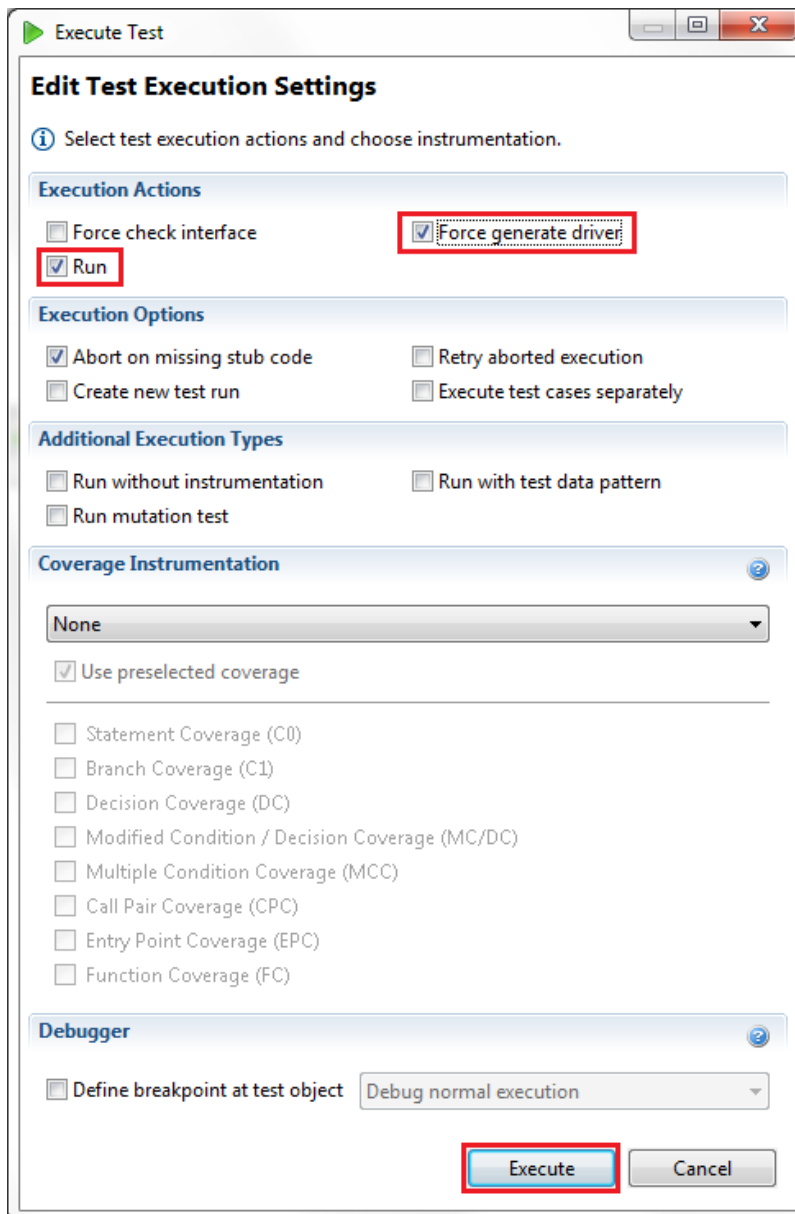
The amount of test objects built in parallel is given by **Compiler Concurrency**, which is by default set to 4, while 20 source files per test object are compiled in parallel. So, no more than 80 processes run in parallel for the built. You can disable the parallel built by setting TEE attribute **Compiler Concurrency** to 1 and by clearing the TEE attribute **Make Options**. The parallel execution is set to 10 by default. Too many compiler or debugger processes in parallel may slow down the test execution. It depends on your system. So, feel free to test different values for **Compiler Concurrency** and **Target Concurrency**. So, if you are using one emulator, please keep TEE attribute **Target Concurrency** at 1.

4 Interactive Debugging

The TESSY Renesas GDB server adaption does not support interactive debugging during a test run. But it is possible to debug your test object interactively having the test data built-in which might be useful in case of errors during a test run. So, in order to debug the test object interactively TESSY provides the TEE attribute **Generate Builtin Data**. The attribute is of type Boolean and, if set to *true*, TESSY will rebuilt your target binary during the next test run having the selected test data built-in, i.e. TESSY will not actually perform the test run but instead create the target binary with test data built-in to it. To disable this feature, you have to set the attribute to *false*.



Open the **Execute Test** dialog and make sure **Force Generate Driver** is selected.

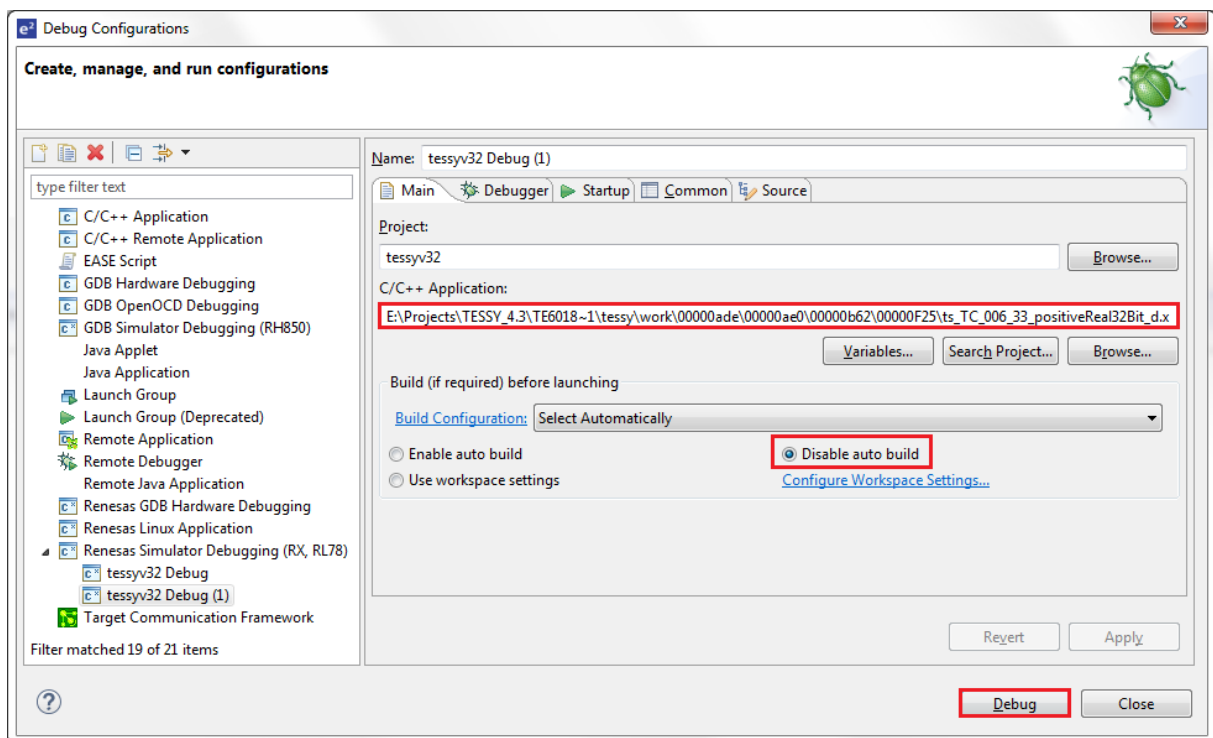


Now execute the test by pressing the **Execute** button. TESSY displays the path to the generated built-in target binary in the **Console** view.

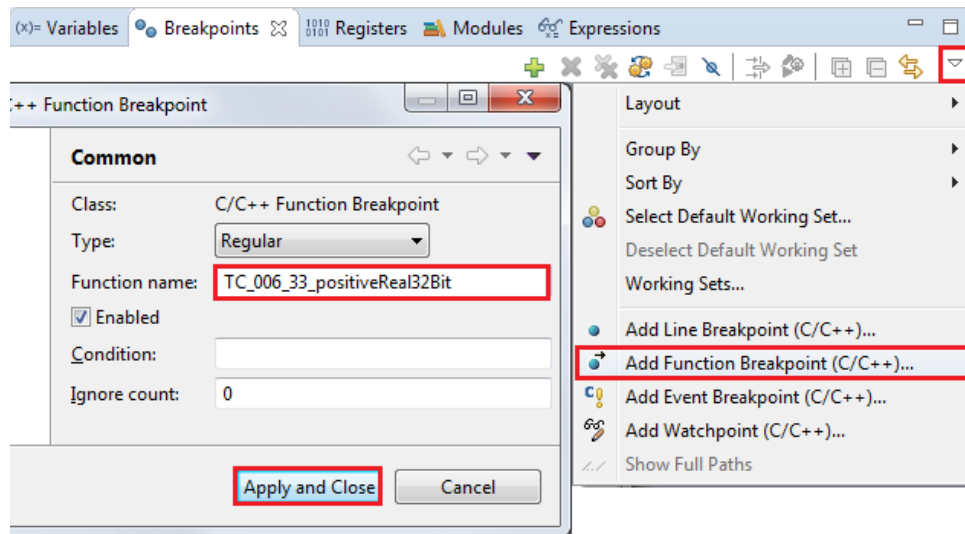
```

Console | Problems | Variants | Notes | Suspicious Elements
-----
Messages
***** Linking Slave *****
*
* Generated target binary 'E:\Projects\TESSY_4.3\TE6018~1\tessy\work\00000ade\00000ae0\00000b62\00000F25\ts_TC_006_33_positiveReal32Bit_d.x' containing builtin data.
Library Generator Completed
*
  
```

You can copy the path of the generated target binary from TESSY's **Console** view and load it into the e2 studio debugger by duplicating your original debug configuration and pasting the path into the C/C++ Application field of the duplicate.



Start the debugger by clicking **Debug**. Now, go back to TESSY and copy the name of your test object. When the debug perspective is up and running, select the **Breakpoints** view and add a function breakpoint. Paste the test object name into the **Function name** field.



Finally, click Resume twice to reach your test object.

Please note: Alternatively, you can set up the corresponding e2 studio TEE environment and switch to that environment for the test object in question so that you can also interactively debug the test object.

5 Troubleshooting

Generally, use the Renesas Simulator environment to run tests with the simulator, i.e. without hardware. Use Renesas Emulator, if you want to run the tests on actual hardware. Ensure to set the correct debugger target and the correct I/O access width data.

By now, there are no known problems. Please contact support@razorcat.com if you encounter any unsolvable problems.