

# GNU gdbserver Debugger

## Abstract

This document describes the usage of the GNU gdbserver debugger as target system running on an operating system. The minimum required version of the GNU gdbserver is 7.4.

## Table of Contents

Abstract .....	1
1 GNU gdbserver Debugger .....	2
2 TESSY Environment Settings.....	2
3 Interactive Debugging.....	3
4 Troubleshooting.....	5
4.1 Starting gdbserver is too slow .....	5

**Please note:** *The GNU gdbserver debugger adaption does not support interactive debugging features when executing tests with TESSY. (See 3 to learn how to debug interactively having your test data statically built into the target binary.)*

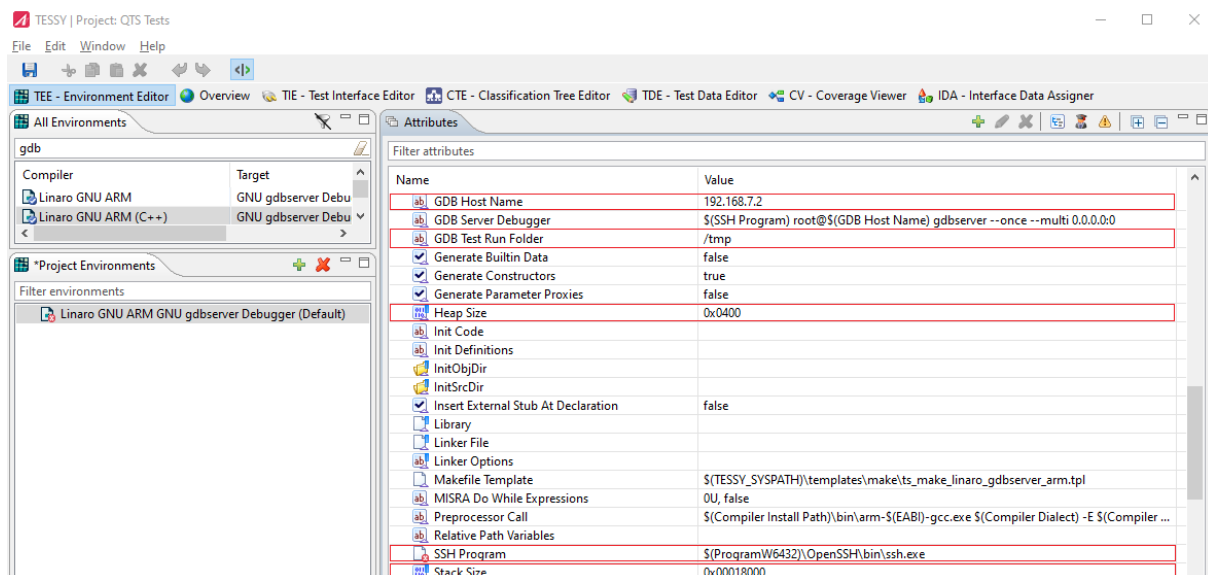
## 1 GNU gdbserver Debugger

The communication between TESSY and the target system is for this adaption based on GDB client and gdbserver as backend. gdbserver is started by TESSY by executing the command line found in TEE attribute **GDB Server Debugger**. By default, the attribute contains a command line that begins with a call to SSH which is followed by the destination host and user name and the commands needed to start gdbserver with its appropriate arguments. *You have to make sure that SSH can log in to the target operating system without entering a password!* There is no SSH version delivered by TESSY. You may install OpenSSH on your PC as for instance provided by <http://www.mls-software.com/opensshd.html>. You may also start gdbserver by any other means. It is though important to let the target operating system choose the port on which to start gdbserver by setting 0 as the initial port number. You are free to choose a fixed port number. But this will impede running tests in parallel. See chapter 4.1 if starting up gdbserver is too slow.

In order to debug the test application interactively with the test case values provided from within 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 respective GDB client debugger and step through the test cases (see chapter 3).

## 2 TESSY Environment Settings

There are several TEE attributes you have to pay attention to.



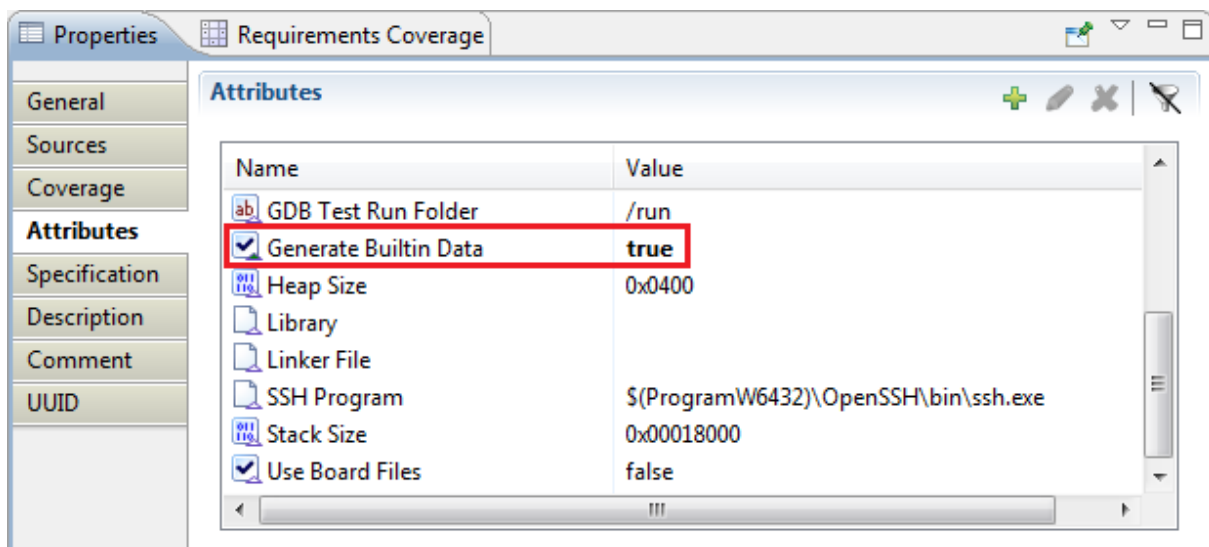
**GDB Host Name** contains the destination IP address where gdbserver is running on and listening for incoming connections. **GDB Test Run Folder** points to the location where gdbserver will temporarily save the target program to be tested. The folder `/run` usually points to memory on Linux systems and might therefore be faster but it is by default not writable by normal users. Thus, you have to start gdbserver with

administrator privileges, which is the default in attribute **GDB Server Debugger**, or leave the default value for this attribute at `/tmp`. **SSH Program** points to the SSH client to be used to connect to the target system and to start gdbserver. You may also use another command to start gdbserver. But if you do so, please adjust the **GDB Server Debugger** attribute accordingly and leave gdbserver's port number at 0.

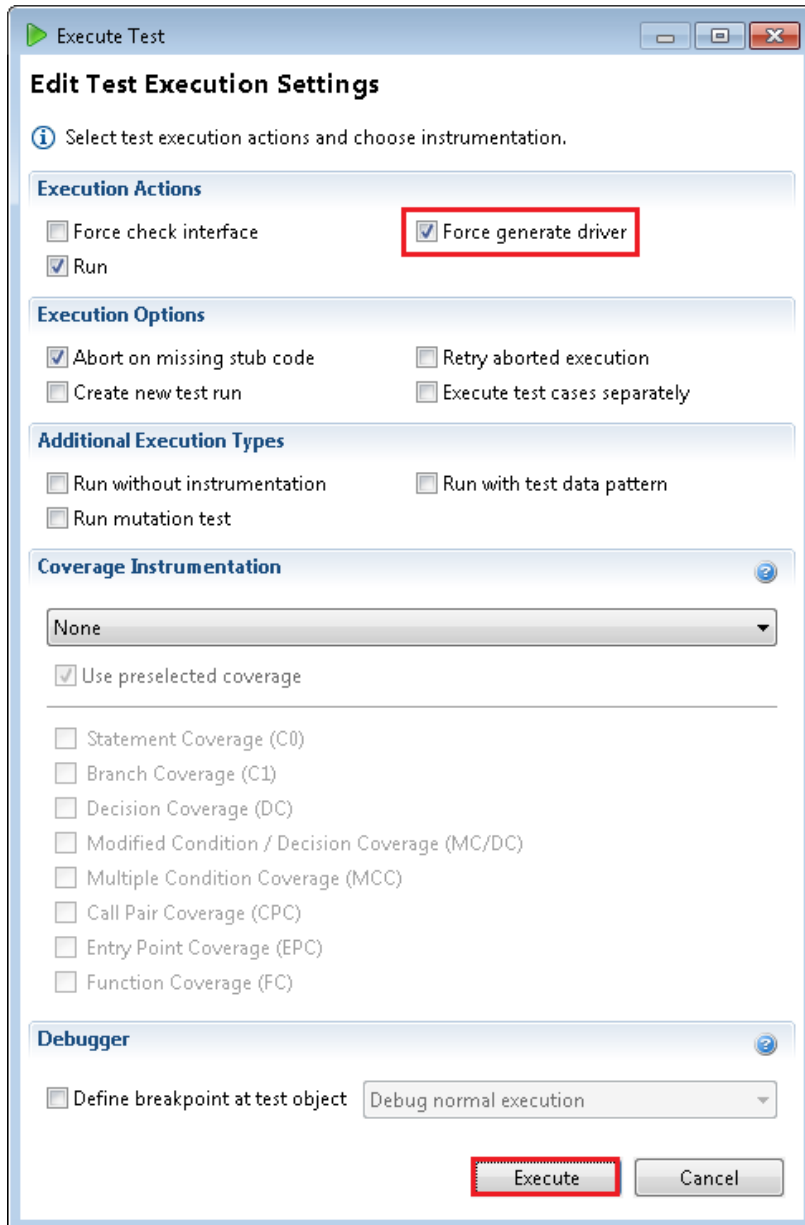
Using Linaro as cross compiler you may also check the attributes **Heap Size** and **Stack Size** and make sure they fit to your requirements.

### 3 Interactive Debugging

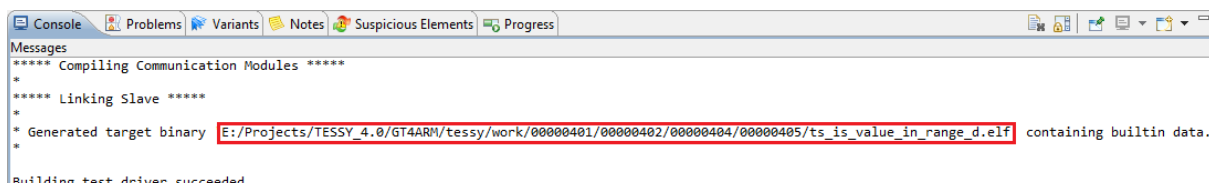
The TESSY gdbserver debugger adaption does not support interactive debugging during a test run. But it is possible to debug your test object interactively having the test data built-in which might be useful in case of errors during a test run. So, in order to debug the test object interactively TESSY provides the **Generate Builtin Data** attribute. 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.



You can copy the path to the generated target binary from TESSY's **Console** view and use the corresponding GDB client to debug the application. Please, consult the GNU GDB documentation found on the internet on how to debug the application.

## 4 Troubleshooting

### 4.1 Starting gdbserver is too slow

If it takes more than a few milliseconds to establish the SSH connection, it is recommended to edit file `/etc/ssh/sshd_config` on your target operating system and add

#### **UseDNS no**

on a single line. You may also start the SSH server with option **-u0** which is done in file `/etc/default/ssh` or in file `/etc/init.d/sshd` on most Linux systems. Both measures prevent the SSH server from resolving the clients host name. Furthermore, it might help to use OpenSSH to connect to a Linux system.