

RadiSys Hawk Debugger

Abstract

This document describes the integration of the RadiSys Hawk debugger.

Please note: *The GUI of the RadiSys Hawk debugger is not available when running tests with Tessy. There is a workaround provided to create a test application containing the test data built into the binary program.*

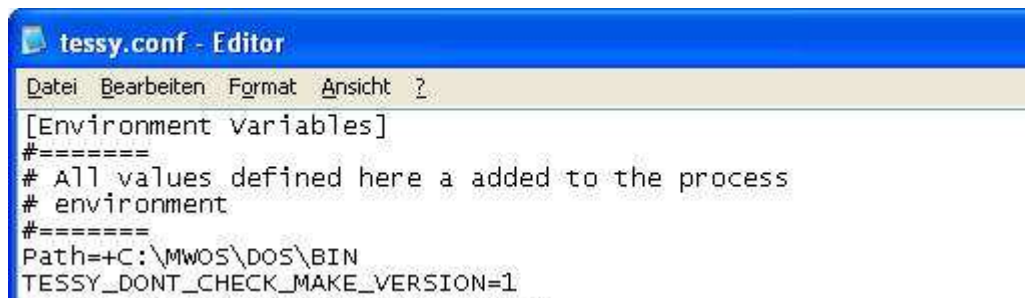
RadiSys Hawk Debugger	1
1 Introduction.....	2
2 Preparing Tessy	2
3 Executing a Test Run	2
4 Debugging using the Hawk Debugger GUI (with built-in test data).....	2
4.1 Step 1: Change the Module Properties	3
4.2 Step 2: Run the Test	3
4.3 Step 3: Load the Test Application with built-in Test Data	4
4.4 Step 4: Set a Breakpoint	4

1 Introduction

This document describes the steps needed to use Tessy in conjunction with RadiSys Hawk Debugger as target debugger.

2 Preparing Tessy

Tessy sets up its own environment variables. Since Microware Ultra C compiler tools need their binary path to be set you have to add that path to Tessy's Path variable. From Tessy's **Options** menu select **Edit Settings...** to open the configuration file. Add the binary path to the Path variable to the section **Environment Variables**.



```
tessy.conf - Editor
Datei Bearbeiten Format Ansicht ?
[Environment Variables]
#=====
# All values defined here are added to the process
# environment
#=====
Path+=C:\MWOS\DOS\BIN
TESSY_DONT_CHECK_MAKE_VERSION=1
```

3 Executing a Test Run

Tessy executes tests with the Hawk debugger using the DLL interface of the Hawk debugger without running the GUI of the Hawk debugger. Upon start of the test run, Tessy will launch the Tessy Target Handler automatically. It will also stop the handler appropriately.

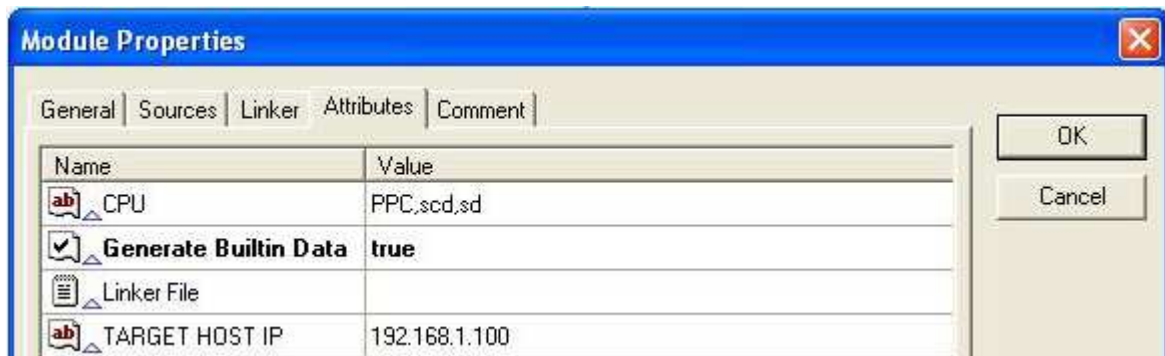
After the test execution, you will see the test results within Tessy. If some test cases failed, you may want to debug the test application. Please, follow the steps described within the next chapter in order to build and debug the test application with the Hawk debugger.

4 Debugging using the Hawk Debugger GUI (with built-in test data)

Since the Hawk debugger does not support interactive debugging controlled from outside (i.e. from Tessy when executing tests), there is an option that allows creating a standalone test application with the test data already built-in. This scenario is available by switching a Tessy module attribute as described below. You may then load this test application into the Hawk debugger and step through the code.

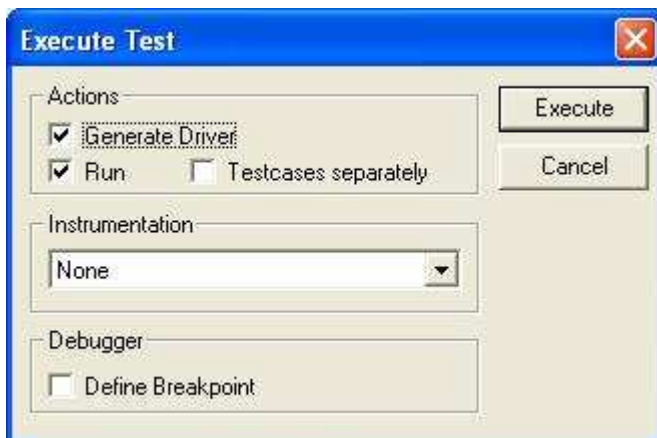
4.1 Step 1: Change the Module Properties

In order to generate a test application with built-in data, select your test module and choose **Properties...** from the context menu. This will open the **Module Properties** dialog. Now select the **Attributes** tab and change the setting of the module attribute **Generate Builtin Data** to true. Press **Ok** to save the changes and close the dialog.



4.2 Step 2: Run the Test

Start the test within Tessy with the **Generate Driver** option selected within the Execute Test dialog as shown below.



Tessy will generate an executable with test data built-in. The name and path of the generated test application will be shown within the **Messages** window of Tessy.

```
Messages
-----
Compile/Link Driver
-----
*
***** Compiling Builtin Data *****
*
***** Compiling Slave *****
*
***** Compiling Communication Modules *****
*
***** Linking Slave *****
*
* Generated target binary 'C:\tessy\samples\2_9_17\Tessy\Sample\_Sample\IsValueInRange\ts_is_value_in_range_d' containing builtin data.
*
```

Now copy the file name of the test application.

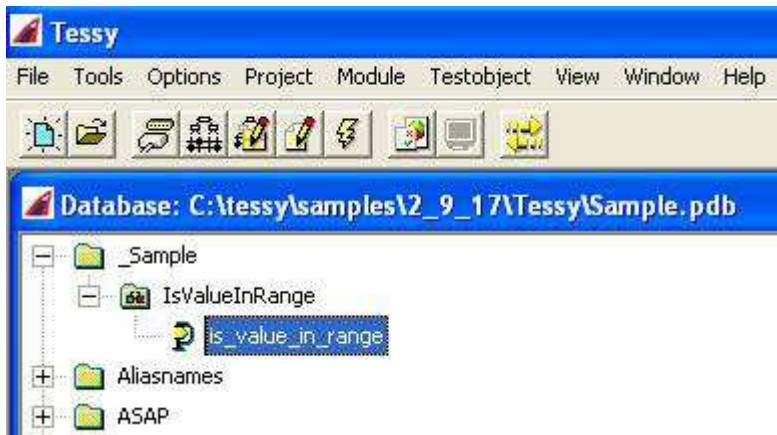
4.3 Step 3: Load the Test Application with built-in Test Data

Load the generated target binary into the Hawk Debugger using **Connect...** from the **Debug** menu by pasting the path into the **Program** field.

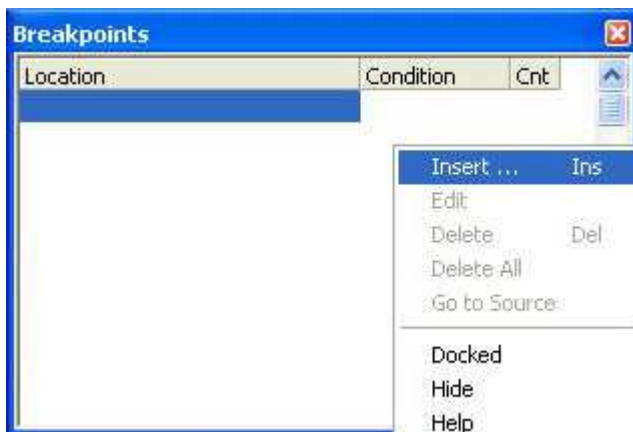


4.4 Step 4: Set a Breakpoint

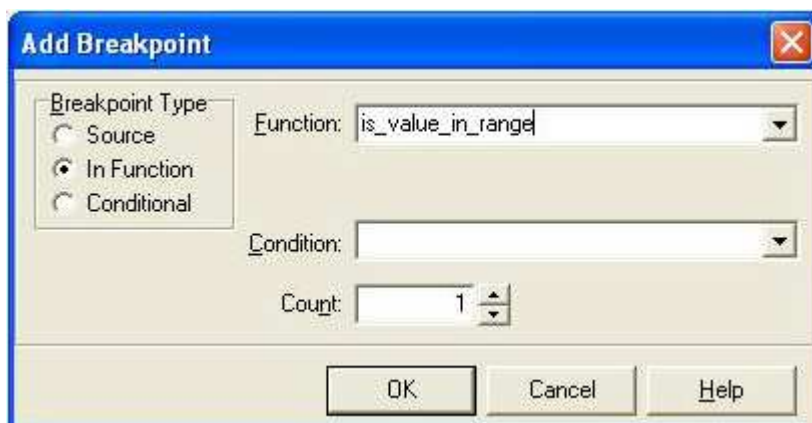
In order to stop at your test object you need to set a breakpoint. The easiest way to do this is to switch back to Tessy and select the test object.



Now press Ctrl-C to copy the name of your test object and switch back to the Hawk Debugger. From the **Breakpoints** view context menu choose **Insert ...** to insert a breakpoint.



In the **Add Breakpoint** view paste the module name into the **Function** field.

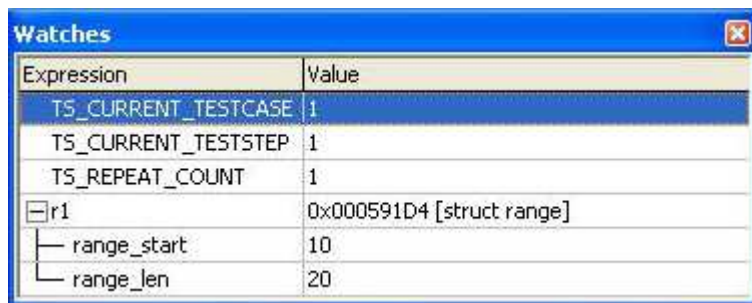


Tessy Application Notes

Now run the target binary by pressing the **Run** button in the tool bar menu.



The test execution will stop at your test object. To keep track of the test cases and test steps you may add the variables **TS_CURRENT_TESTCASE** and **TS_CURRENT_TESTSTEP** to the Watches view.



Expression	Value
TS_CURRENT_TESTCASE	1
TS_CURRENT_TESTSTEP	1
TS_REPEAT_COUNT	1
r1	0x000591D4 [struct range]
range_start	10
range_len	20