

CTC++ integration

Abstract

This document describes the necessary steps to use the tool Testwell CTC++ (Test Coverage Analyzer for C/C++, from Verifysoft Technology GmbH) together with TESSY. The TESSY integration provides a ready-to-use data transfer from the target to the host machine which is required for acquiring the CTC++ coverage data.

Please refer to the CTC++ manual for details about how to use CTC++, its options and the resulting data and reports.

Table of contents

1	CTC++ host to target interface (HOTA)	2
2	Activation steps	2
2.1	Step 1: Add the CTC++ path	2
2.2	Step 2: Edit the makefile template	3
2.2.1	Include the predefined makefile	3
2.2.2	Add the object files to the link target	5
2.3	Step 3: Enable the CTC++ mode	6
3	Running tests	7
3.1	Instrumentation mode	7
3.2	Result files generated during test execution	8
3.3	Creating the CTC++ coverage report	9

1 CTC++ host to target interface (HOTA)

The CTC++ tool instruments arbitrary source files and collects various coverage data. When the test is executed on a target debugger or simulator (as with TESSY), the collected coverage data needs to be transferred back to the host machine in order to be analyzed and reported within coverage reports.

This transfer of data from the target to the host machine is readily prepared within TESSY. You need to do the settings described within the following sections in order to activate the CTC++ integration.

2 Activation steps

Please follow the steps below to enable the usage of CTC++ with TESSY.

2.1 Step 1: Add the CTC++ path

Because TESSY will call CTC++ when compiling/linking the test application, the path to the CTC++ installation directory needs to be added to the PATH environment variable managed by TESSY.

Please note: This is required even if the CTC++ installation path is already within the PATH environment variable on your computer, because TESSY manages its own PATH variable. It may also be necessary to add another environment variable LM_LICENSE_FILE in order to provide the license for CTC++.

Open the respective configuration file of TESSY (e.g. for version 4.3.4):

```
%appdata%\Razorcat\Tessy\4.3.4\config\tessy.conf
```

At the end of the file, there is a section **[Environment Variables]**. Please add the following line into this section (according to the installation path of CTC++ on your machine):

```
PATH+=C:\Testwell\CTC
```

```
35 [Environment Variables]
36 #=====
37 # All values defined here a added to the process
38 # environment
39 #=====
40
41 PATH+=C:\Testwell\CTC
42
```

I

Save the file and restart TESSY in order to load the new settings.

2.2 Step 2: Edit the makefile template

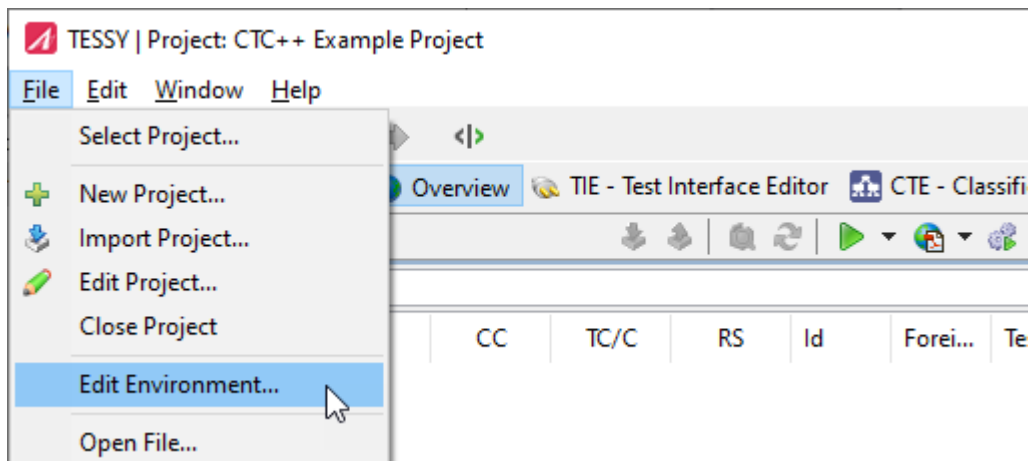
The control program of CTC++ will be called from within the makefile template of TESSY. You need to add the **ctc** command to the compiler and linker call and add some CTC++ source files to the test application.

2.2.1 Include the predefined makefile

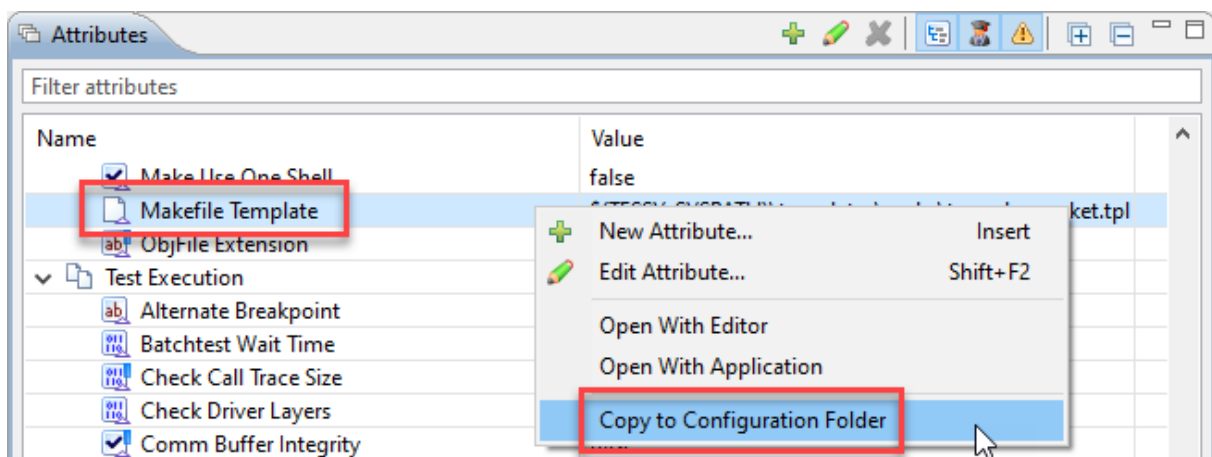
There is a ready-to-use makefile include within the TESSY installation directory, that contains the CTC++ calls to the compiler/linker call and defines the list of required additional CTC++ source files:

```
C:\Program Files\Razorcat\TESSY_4.3\sys\src\ctc\comm_ctc.mk
```

It is recommended to copy the makefile template into the config folder of your TESSY project before doing any changes. Open the TEE perspective:

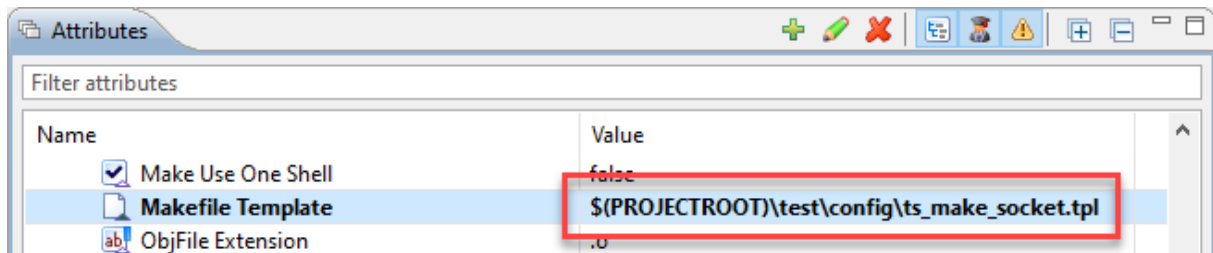


Within the attributes view of the TEE perspective, the “Copy to Configuration Folder” menu entry is the easiest way to do this:

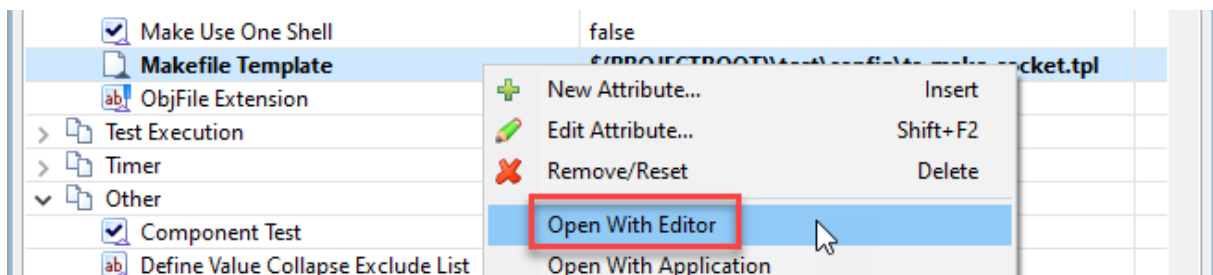


TESSY Application Notes

This will copy the makefile template from the installation folder into the configuration folder:



Now please open this copied file using the “Open With Editor” menu entry:



Search for both variables **S_CC** and **S_LINK** and place the following include statement after these lines:

```
include $(TESSY_SYS)\src\ctc\comm_ctc.mk
```

```

76 #
77 # SLAVE
78 #
79 S_CC      := gcc
80 S_LINK    := gcc
81 S_INCLUDES := -I$(TESSY_SYS)/include \
82             -I$(TESSY_SYS)/include/tessy \
83             -I$(TESSY_SYS)/include/tessy/comm \
84             -I$(TESSY_SYS)/include/tpt
85 S_STUB_OBJECT := $(MODULE_PATH)/ts_$(TESTOBJECT)_stubs.o
86 S_UC_OBJECT   := $(MODULE_PATH)/ts_$(TESTOBJECT)_usr.o
87 S_LIBRARIES   := -lpsapi -lws2_32
88 S_COMP_OPTIONS := -g $(COMPILER_DIALECT) -Wall -DTS_SLAVE -DTESSY -DTS_SOCKET -DTS_HAVE_INT64 -D
89 S_LINK_OPTIONS :=
90
91 include $(TESSY_SYS)\src\ctc\comm_ctc.mk
92

```

2.2.2 Add the object files to the link target

The CTC++ source files that need to be linked to the test application are already available within the following TESSY installation directory:

```
C:\Program Files\Razorcat\TESSY_4.3\sys\src\ctc
```

Please note: These files are derived from CTC++ version 9.1. You may need to update some files when using newer versions of CTC++. Please refer to the CTC++ manual for details.

The CTC++ files must be added to the link rule within the makefile template like shown below (e.g. for GNU/GCC):

```
177 #
178 # Objects to be delete after the build process.
179 #
180 TS_INTERMEDIATE_OBJS := $(BUILTIN_OBJ) $(MODULE_PATH)/ts_$(TESTOBJECT)_s.o $(S_SRC_OBJECTS) \
181 $(S_UC_OBJECT) $(S_STUB_OBJECT) $(S_TIMER_OBJECT) $(COMM_OBJECTS) \
182 $(S_TPT_OBJECTS) $(CTC_MODULES)
183 .INTERMEDIATE: $(TS_INTERMEDIATE_OBJS)
184
185 $(MODULE_PATH)/ts_$(TESTOBJECT)$(EVOL)_$(TARGET_FLAG).exe : $(TS_INTERMEDIATE_OBJS)
186 @echo *
187 @echo ***** Linking Slave *****
```

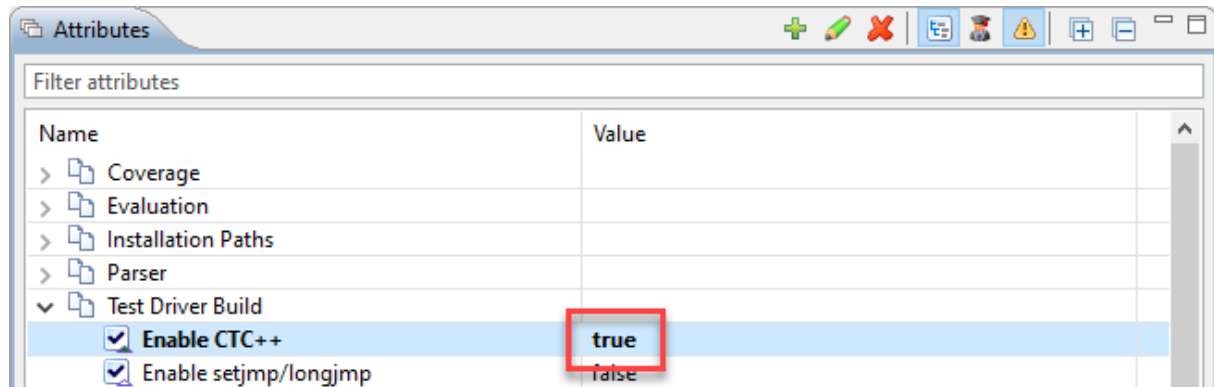
Add the following variable to the TS_INTERMEDIATE_OBJS variable (or both to the end of the linker command dependencies list as well as to the end of the linker command line):

```
$(CTC_MODULES)
```

If there are any errors during compilation or if you don't find the correct location, please contact Razorcat support and send the makefile template that you want to use.

2.3 Step 3: Enable the CTC++ mode

The TESSY test driver generator needs to add the call to the data transfer function of CTC++ into the test application code. In order to enable this functionality, please change the value of the **Enable CTC++** attribute to **true**:

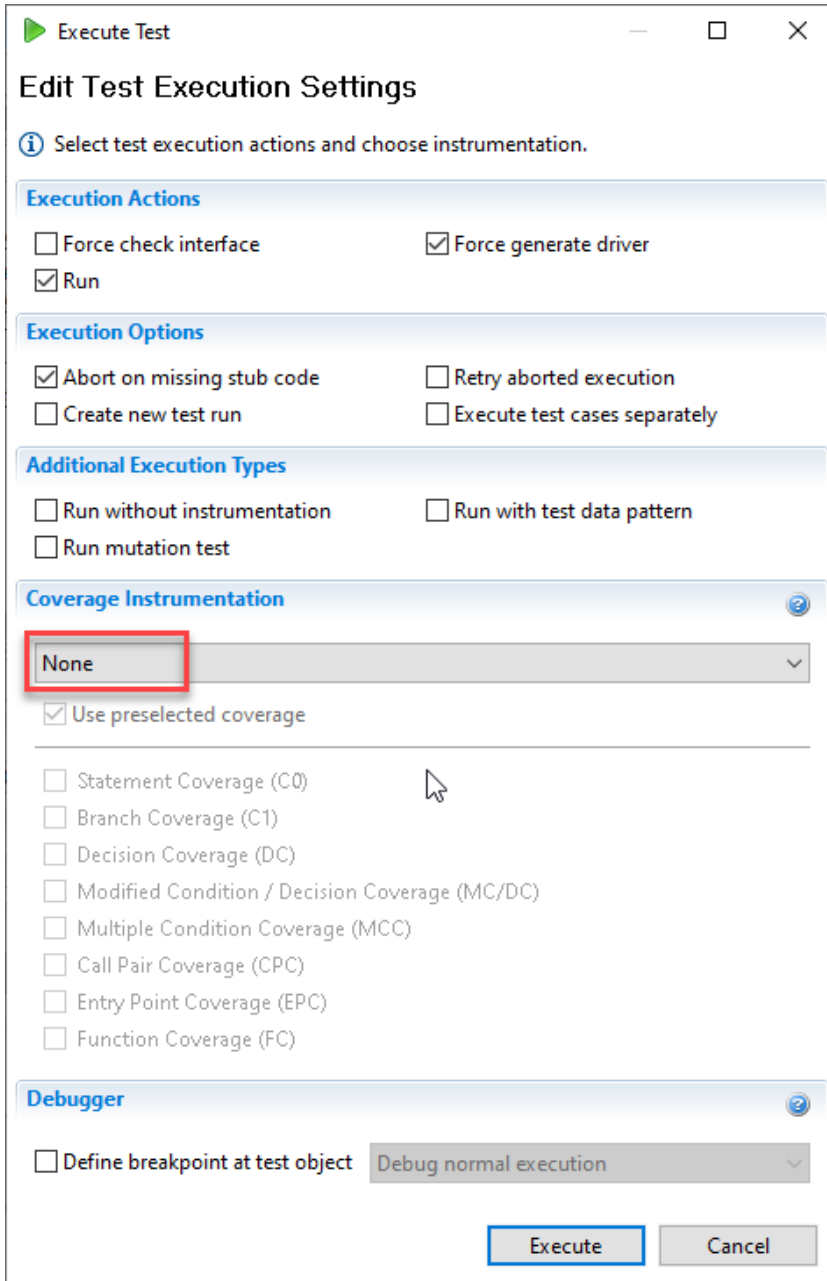


Save your changes and switch back to the overview perspective.

3 Running tests

3.1 Instrumentation mode

When running tests, you should not select any coverage measurement within TESSY, because the TESSY coverage instrumentation may interfere with the CTC++ instrumentation. Use the following **Instrumentation** mode within the TESSY test execution dialog:



Execute Test

Edit Test Execution Settings

Select test execution actions and choose instrumentation.

Execution Actions

- Force check interface
- Force generate driver
- Run

Execution Options

- Abort on missing stub code
- Retry aborted execution
- Create new test run
- Execute test cases separately

Additional Execution Types

- Run without instrumentation
- Run with test data pattern
- Run mutation test

Coverage Instrumentation

None

- Use preselected coverage
- Statement Coverage (C0)
- Branch Coverage (C1)
- Decision Coverage (DC)
- Modified Condition / Decision Coverage (MC/DC)
- Multiple Condition Coverage (MCC)
- Call Pair Coverage (CPC)
- Entry Point Coverage (EPC)
- Function Coverage (FC)

Debugger

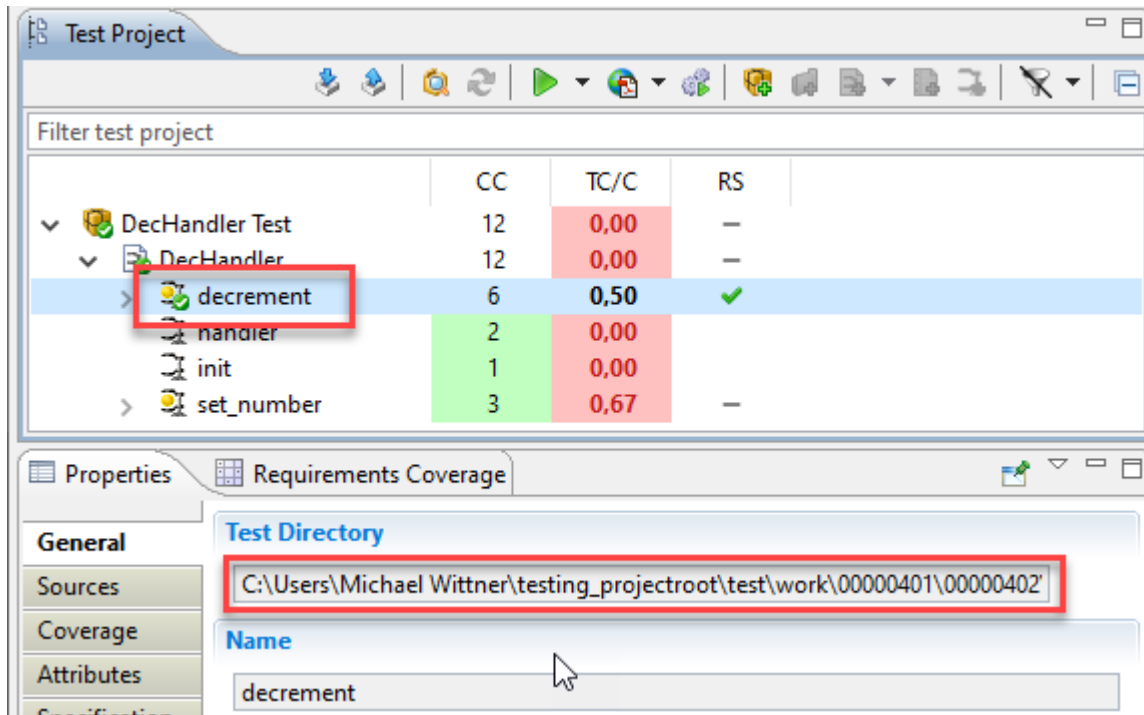
- Define breakpoint at test object

Debug normal execution

Execute **Cancel**

3.2 Result files generated during test execution

After each test execution, you will find some additional files within the test object work folder. You can open the work folder of your test object by selecting the test object and using ALT + double-click on the **Test Directory** field within the **General** tab of the properties view:



The following files will be generated when running the test with the CTC++ integration enabled:

File name	Description
MON.sym	Symbol file generated by CTC++ during instrumentation of the source file(s)
ts_<test object name>_stdout.txt	Output file containing the coverage data of CTC++ in raw format. Needs to be converted using the CTC++ utilities

3.3 Creating the CTC++ coverage report

There is a DOS batch script available that converts the raw data files into CTC++ format files and creates the coverage report. You can find this file within the following directory of the TESSY installation:

```
C:\Program Files\Razorcat\TESSY_4.3\bin\tessy2ctc.bat
```

Call this script from the DOS command line or from another DOS script with the appropriate options:

1. Test directory name of the test object
2. Test object name

This will convert the files, create and show a coverage report.