

# Legacy Tools

## Abstract

This application note describes how to use legacy compilers and debuggers.

## Table of contents

Abstract .....	1
1 Introduction.....	2
2 Obsolete Combinations .....	2
3 Legacy Makefile Templates.....	3
4 Online Samples .....	4

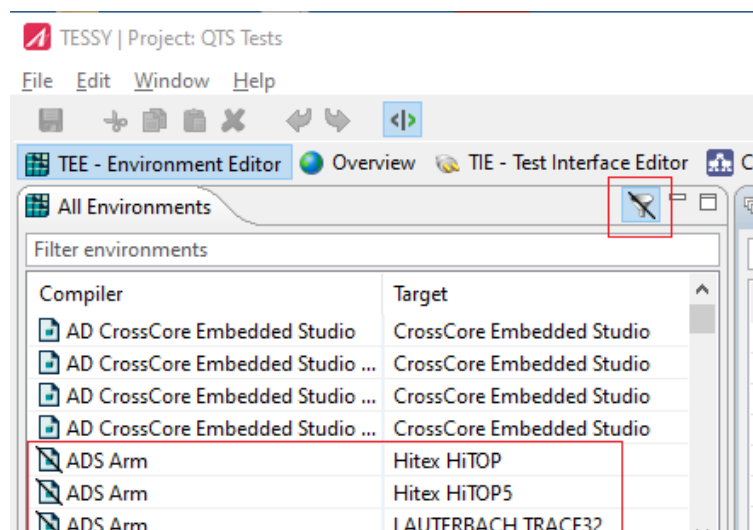
## 1 Introduction

This application note will guide you through the different types of approaches to activate and utilize so-called legacy tools. *Legacy tool* means *not* that we adapt TESSY to an obsolete compiler or debugger which has not been supported by now. Legacy tools denote compilers, debuggers, and combinations to which TESSY has been fully adapted to. The difference to non-legacy tools is their today's accepted usage in the embedded world. Most often there are newer versions of the tools available. This usually means that the tools have different command line arguments or even different program names and thus are different programs.

Legacy tools are supported in several different ways. Their activation and usage will be represented in the following chapters step by step. The second chapter handles legacy compiler / debugger combinations which is associated with TESSY's environment editor (TEE). So, it might be a good idea to consult chapter 6.5 *TEE: Configuring the test environment* from TESSY's user manual if you have not done by now. The TESSY installation folder `sys\compilers` and `sys\targets` contain several compiler respectively debugger specific files in their subfolders which originate mostly from third party products and might be obsolete. Especially as far as linker files and startup code files are concerned, you should always try to use newer versions of these files from your own development project. See application note 063 *Startup Code of the Test Program* for further hints on the integration of startup code and linker files. The third chapter handles legacy compilers to be used which is associated with make file templates. The fourth chapter handles legacy projects.

## 2 Obsolete Combinations

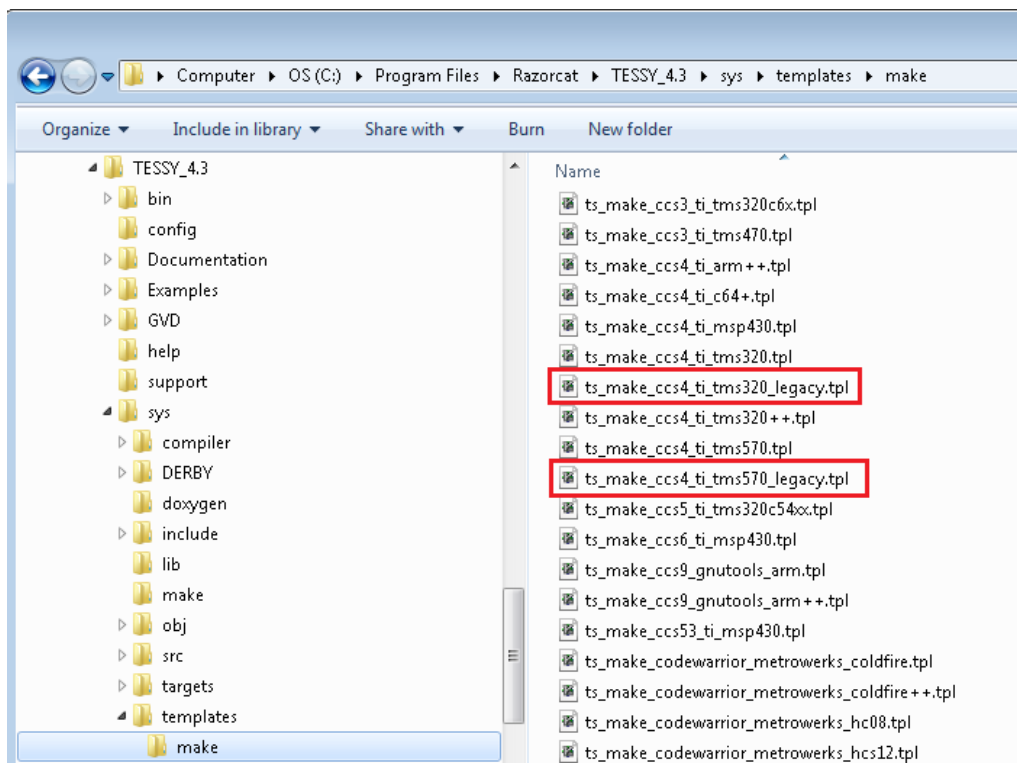
As mentioned in the introduction over time compilers and debuggers become obsolete. So, in order not to confuse you with all versions of compiler / debugger combinations TESSY separates obsolete compiler / debugger combinations from the latest supported ones. This is simply done by hiding all legacy compilers and debuggers. Start the TEE and select **Show Obsolete** from the menu by pressing the filter symbol to disable the filter.



Finally, the obsolete compilers and debuggers are displayed while their first symbol is struck through in order to separate them visually from non-obsolete compilers and debuggers. Nevertheless, they are fully functional. But keep in mind that oftentimes even the compiler and debugger vendors do not support these obsolete ones any more. So, problems could occur that can finally only be solved by using the newer versions of the respective compiler or debugger.

### 3 Legacy Makefile Templates

As long as only the compiler arguments or name is concerned obsolete versions might still be available and activatable by choosing a different version of the Makefile template. Those Makefile templates that support a legacy compiler version are named *\*\_legacy.tpl* where *\** denotes the normal Makefile template's base name.



TESSY | Project: QTS Tests

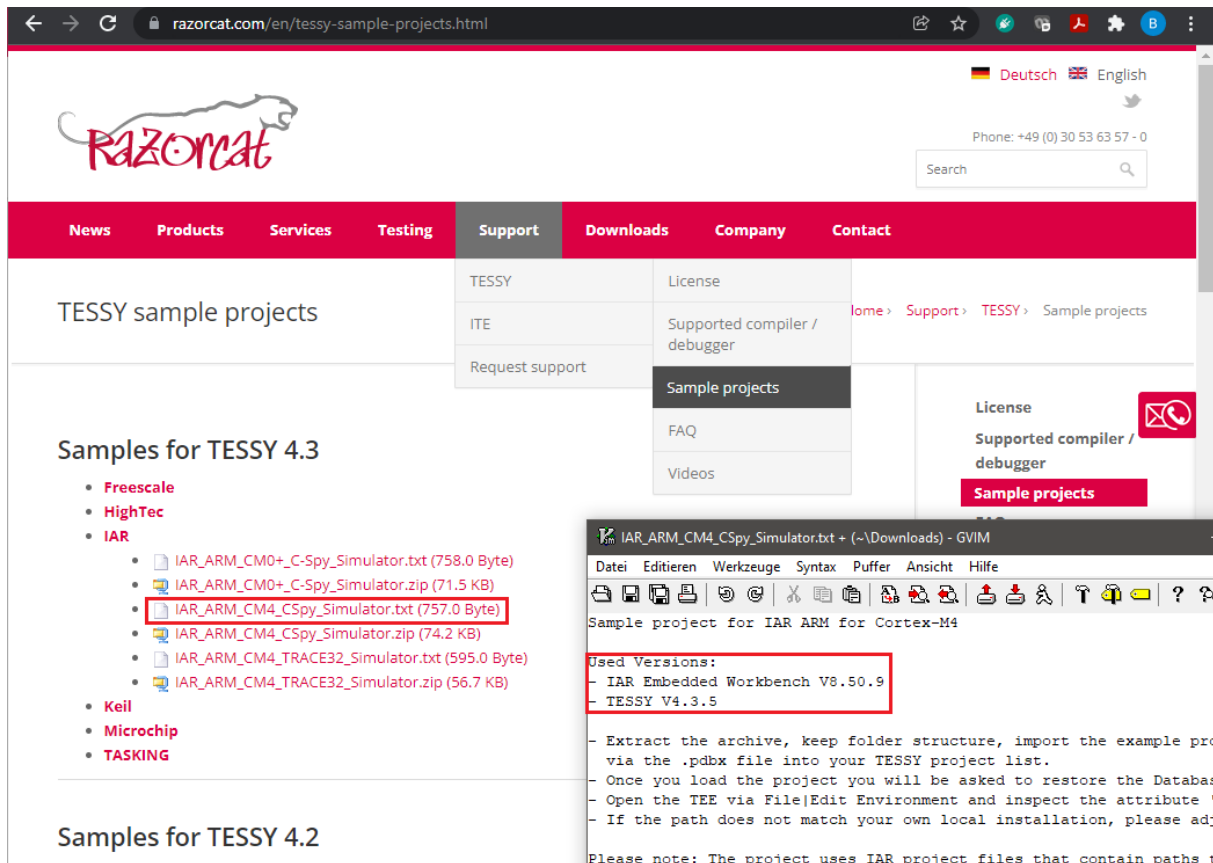
File Edit Window Help

TEE - Environment Editor Overview TIE - Test Interface Editor CTE - Classification Tree Editor TDE - Test Data Editor CV - Coverage Viewer

Name	Value
Heap Size	160
HW MPY	none
Init Code	WDTCCTL = WDTPW + WDTHOLD; _EINT0;
Init Definitions	#include <msp430fr2033.h>
InitObjDir	
InitSrcDir	
Insert External Stub At Declaration	false
Library	\$(Compiler Install Path)\lib\libc.a
Linker File	\$(TESSY_SYSPATH)\compiler\ti\msp430\lnk_msp430fr2033.cmd
Linker Options	
Makefile Template	\$(TESSY_SYSPATH)\templates\make\ts_make_ccs6_ti_msp430.tpl
MISRA Do While Expressions	0U, false
Relative Path Variables	

## 4 Online Samples

On our website you can find several TESSY projects prepared for different TESSY versions as well as different compiler and debugger versions. These projects have been successfully tested. Each project is accompanied by a text file which contains information about the tool used for the test run including its version.



The screenshot shows the Razorcat website's 'TESSY sample projects' page. The page has a navigation menu with items like News, Products, Services, Testing, Support, Downloads, Company, and Contact. A dropdown menu is open under 'Support', showing options like TESSY, License, ITE, Supported compiler / debugger, Request support, Sample projects, FAQ, and Videos. The 'Sample projects' option is highlighted. Below the navigation, there's a search bar and a phone number: +49 (0) 30 53 63 57 - 0. The main content area is titled 'TESSY sample projects' and lists 'Samples for TESSY 4.3'. Under the 'IAR' category, several files are listed, including 'IAR\_ARM\_CM4\_CSpy\_Simulator.txt (757.0 Byte)', which is highlighted with a red box. An inset window shows the content of this file, titled 'IAR\_ARM\_CM4\_CSpy\_Simulator.txt + (~\Downloads) - GVIM'. The file content includes 'Used Versions:' followed by '- IAR Embedded Workbench V8.50.9' and '- TESSY V4.3.5', both highlighted with a red box. Below this, there are instructions on how to use the project files.

If you find a project that corresponds to your compiler and debugger, you may download the package and unzip it into a folder. It is strongly recommended to avoid too long paths and also to avoid spaces in the path name because several target compilers do not support them. Finally, import the project as described in chapter 4.1.2 *Creating, importing, cloning, editing, deleting a project* of TESSY's user manual.