

# TASKING VX and Classic Compiler

## Abstract

This document describes how to deal with the different TASKING VX and classic compilers. The TASKING Script Debugger is described by application note *074 Using TASKING Script Debugger*.

## Table of contents

Abstract .....	1
1 Introduction.....	2
2 Tricore VX Compiler .....	2
2.1 TEE attribute settings .....	2
2.2 Adapting the TASKING EDE settings.....	3
2.3 Adapting the TESSY makefile template .....	4
3 C166/XC166 VX.....	6
3.1 Adapting the default project settings .....	6
3.2 Adapting the TESSY makefile template .....	9
3.2.1 TEE settings.....	10
3.2.2 Compiler command line options.....	11
3.2.3 Linker command line options.....	12
4 C166/XC166 (Classic).....	13
4.1 Copy compiler/linker options .....	13
4.2 Specify the startup code.....	14
5 Flashing the test binary .....	14
6 Disable the watchdog .....	14

## 1 Introduction

The TASKING compilers are currently available within two different versions:

- The VX TASKING compiler which is a completely new product.
- The legacy “Classic” TASKING compiler (e.g. version 8.x for C166)

TESSY supports both versions of the compiler, but the handling is different and described within the following chapters.

Please note that older versions (e.g. V2.x) of the TASKING VX compiler for Tricore may require slightly different settings, because the behavior of the compiler and linker has been changed. The settings described in this application note are applicable for the current versions of the VX compiler (e.g. TASKING VX 3.5 and later).

## 2 Tricore VX Compiler



The TASKING VX Tricore V2.0 and later allows creating all settings required for compilation and linking within the EDE. There is a sample project within the TESSY installation:

```
...\sys\compiler\tasking\tricore_vx\sample.pjt
```


Open this project for all changes to the startup or memory layout configuration of your modules. The default module attributes within TESSY refer to the files generated from this TASKING EDE project. Nowadays, you may as well use your own TASKING EDE project.

### 2.1 TEE attribute settings

As startup code **cstart.o** and **sync\_on\_halt.o** in conjunction with the TASKING EDE project’s specific **linker file (lsl)** are normally required. Copy your project’s startup code, i.e. the linker and object files, into a subdirectory of your TESSY project’s **config** directory. Let TEE attribute **InitObjDir** point to this directory, keep attribute **InitSrcDir** empty, and set attribute **Use Board Files** to **true** as shown below.

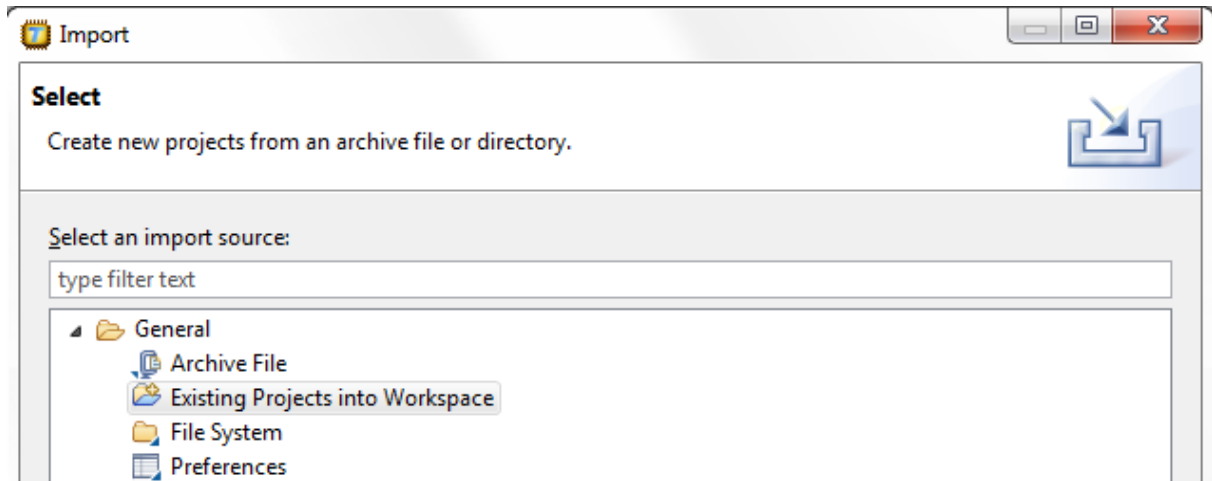
 InitObjDir	\$(PROJECTROOT)\tessy\config\startup
 InitSrcDir	
<input checked="" type="checkbox"/> Use Board Files	true

Let the linker file point to the copied TASKING EDE projects’ linker file (lsl).

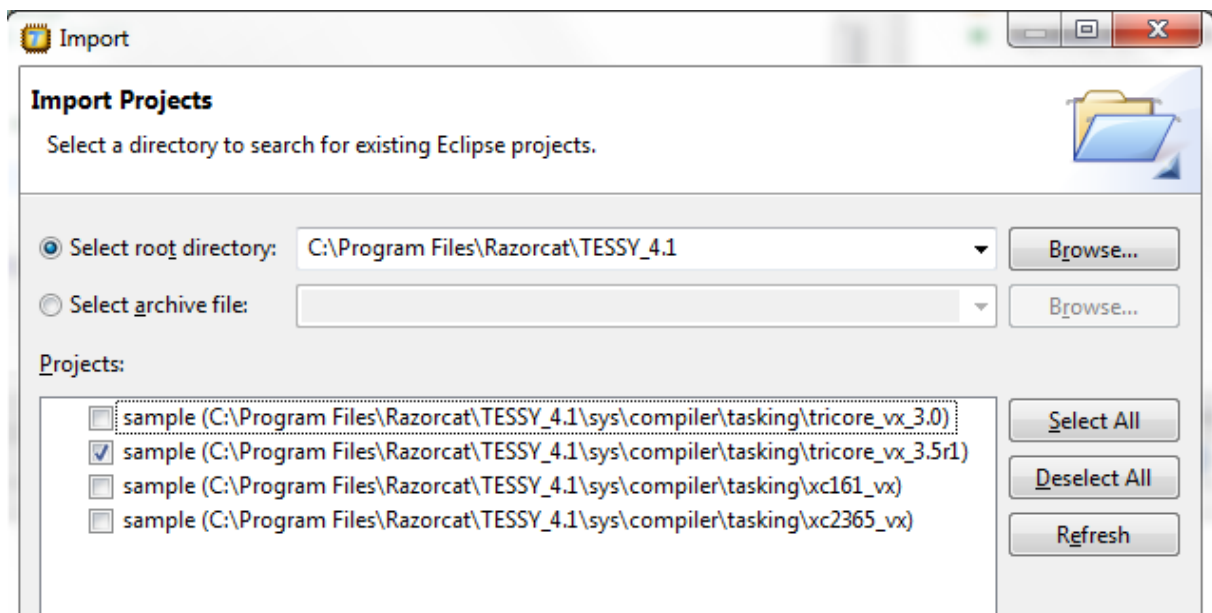
 Linker File	\$(InitObjDir)\_TC38x.lsl
-------------------------------------------------------------------------------------------------	---------------------------

## 2.2 Adapting the TASKING EDE settings

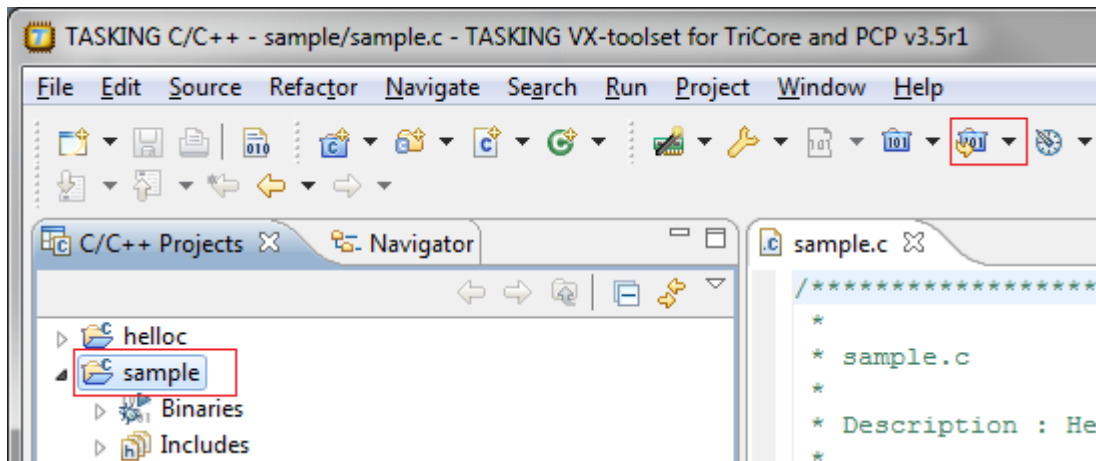
If you decided to use a TASKING EDE project from TESSY's installation directory, read this chapter. In order to change any settings within the TASKING EDE project, you need to import the project into an existing workspace of the TASKING EDE. From the **File** menu select **Import...** and import the desired TASKING EDE project.



Within the second page of the import dialog select the installation directory of TESSY. The available projects for TASKING VX compilers will be shown. Select the most recent TriCore sample project.



Choose **Rebuild Project** to generate the object and LSL files:



The object and LSL files for this project will be generated from within the TASKING EDE. The following files will be used by TESSY to compile and link the test driver.

- ...\\TESSY\_4.1\\sys\\compiler\\tasking\\tricore\_vx\_3.5r1\\Debug\\cstart.o
- ...\\TESSY\_4.1\\sys\\compiler\\tasking\\tricore\_vx\_3.5r1\\sample.lsl

You can change the processor settings according to your needs and rebuild the project again.

**Please note:** Some TriCore processors have special memory access restrictions. You need to make sure that the variables used by the TESSY communication files are correctly placed in memory (e.g. for TC1775B, the variables need to be placed into **byte accessible** RAM memory, e.g. 0xD0000000).

Since there are many possibilities on how to configure the TriCore CPU, you will probably use the EDE project that you are also using for development. In this case, simply change the module attributes as described above to point to the files of your own TASKING EDE project.

## 2.3 Adapting the TESSY makefile template

TESSY uses a makefile template to compile and link the test driver application. The makefile template for TASKING Tricore is located within the TESSY installation directory:

- ...\\TESSY\_4.1\\sys\\templates\\make\\ts\_make\_taskingvx\_tasking\_tricore.tpl

The compiler and linker options used by the TASKING EDE are stored within several files of the “debug” subdirectory of the TASKING EDE project:

- .sample.elf.opt contains the linker options
- .cstart.o.opt contains the compiler options

Please check whether the options within the makefile template are the same as within the above options files. The relevant template variables are shown below.

```
109 # CPU = "TC1796B"
110 TASKING_EDE_OPT_CC := -C"$(CPU)" -t -Wa-g&HLS --emit-locals=-equ,-symbols -Wa-Ogs \
111     -Wa--error-limit=42 --iso=99 --language=-gcc,-volatile,+strings \
112     --switch=auto --align=0 --default-near-size=0 --default-a0-size=0 \
113     --default-a1-size=0 -O2 --tradeoff=4 --compact-max-size=200 -g --source
114
115 TASKING_EDE_OPT_ASM := -C"$(CPU)" -t -Wa-g&HLS --emit-locals=-equ,-symbols -Wa-Ogs -Wa--error-limit=42
116
117 TASKING_EDE_OPT_LC := -C"$(CPU)" -t "$(LINKER_FILE)" -Wl-OtxycL --no-map-file \
118     -Wl--map-file="$(MODULE_PATH_DOS)\ts_$(TESTOBJECT).mapxml":XML \
119     -Wl-mcrfik15mNoduQ -Wl--error-limit=42 -g
```

These options normally do not change when changing the CPU type so that in most cases there are no adaptations necessary. Just make sure to change the **CPU** attribute within the TEE attribute settings as described above.

### 3 C166/XC166 VX

The TASKING VX compiler version for the C166 device family supports the following microcontrollers:

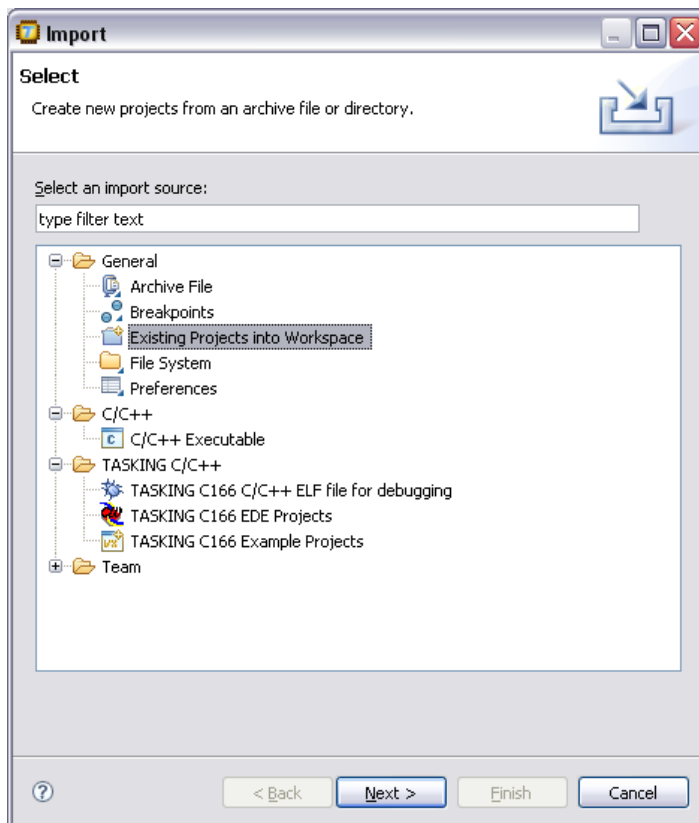
- C166/XC166
- XC2000
- XE166
- and other derivatives

Please refer to the TASKING EDE manual for further details. The default setting for TESSY uses the XC161 device.

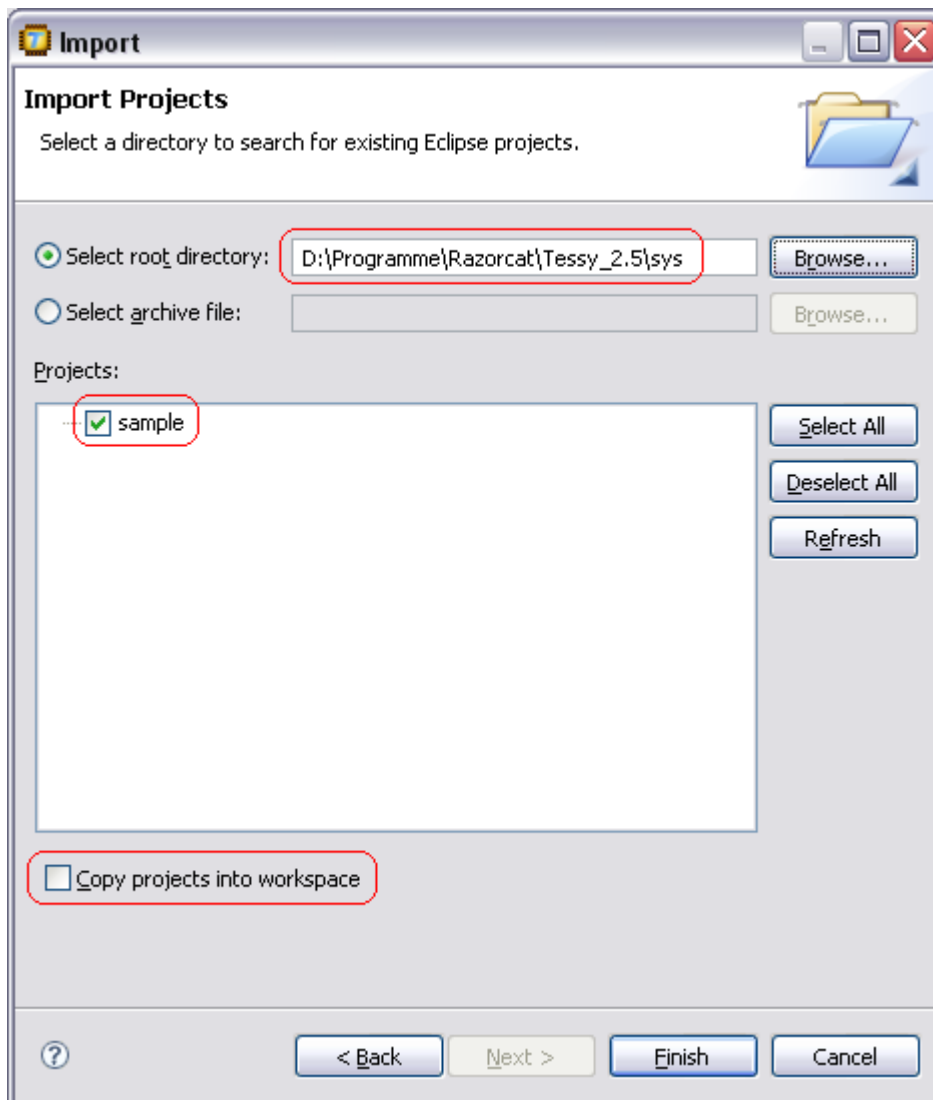
#### 3.1 Adapting the default project settings

The TESSY installation contains a sample TASKING EDE project that holds the compiler and linker settings used within the corresponding makefile template of TESSY. The sample project is configured for an XC161CS device.

Choose **File->Import ...** to edit the default project delivered by TESSY. This will add the project to your TASKING EDE workspace. Choose the **Existing Project into Workspace** option within the **Import** dialog like shown below:

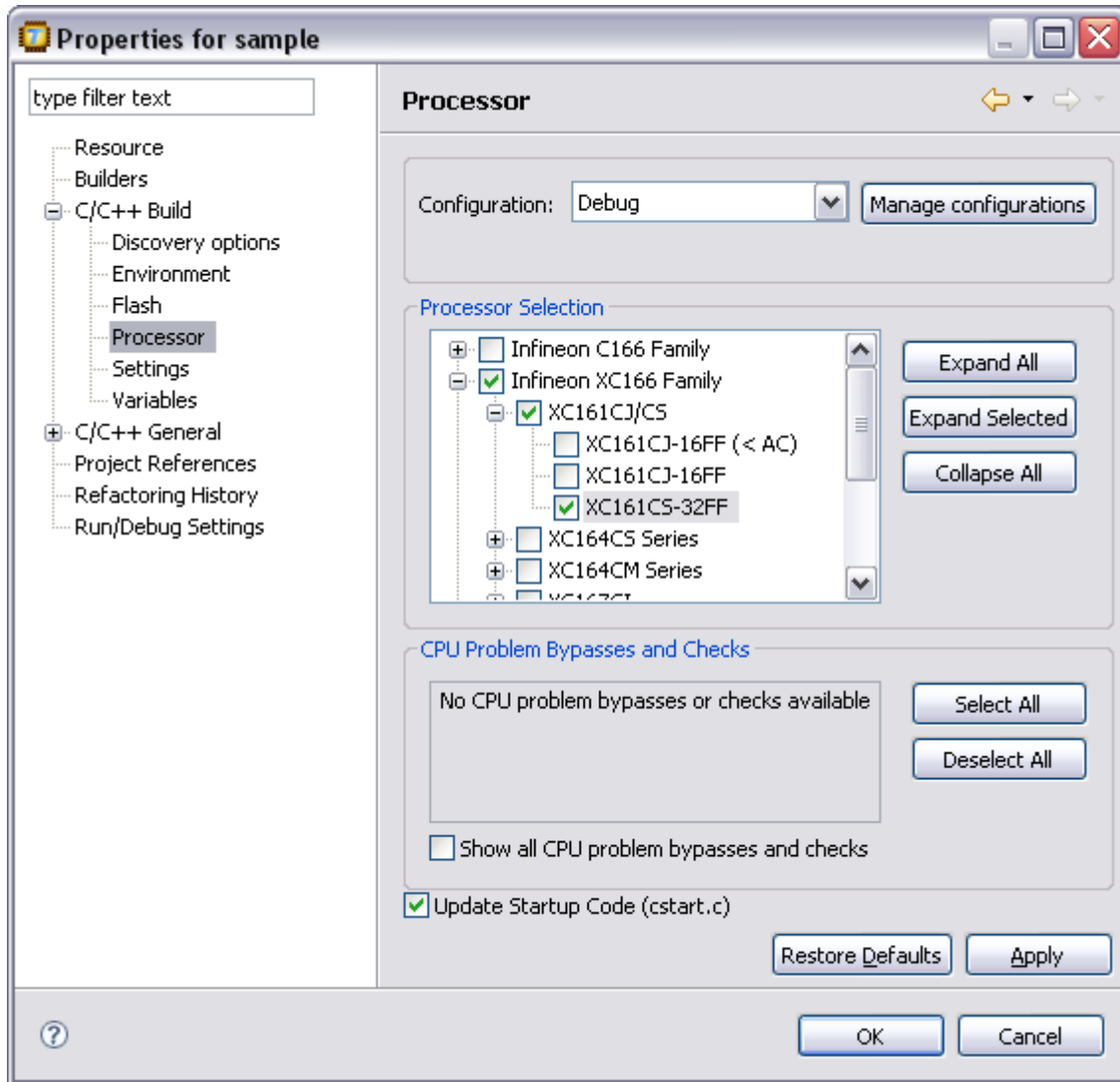


Pressing the **Next** button will show the following dialog. Please enter the TESSY installation path as root directory. This will cause the wizard to search for projects within this directory and if everything works fine, the “sample” project will be listed. Uncheck the **Copy projects into workspace** checkbox, since we want to change the settings within the given project location.



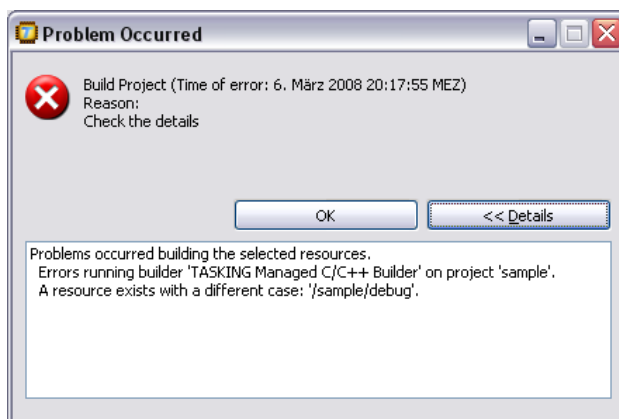
Pressing the **Finish** button will add this project to your workspace. You may now review and change the project settings within the TASKING EDE.

Using the **Properties** menu entry of the context menu will display the **Properties** dialog as shown below.

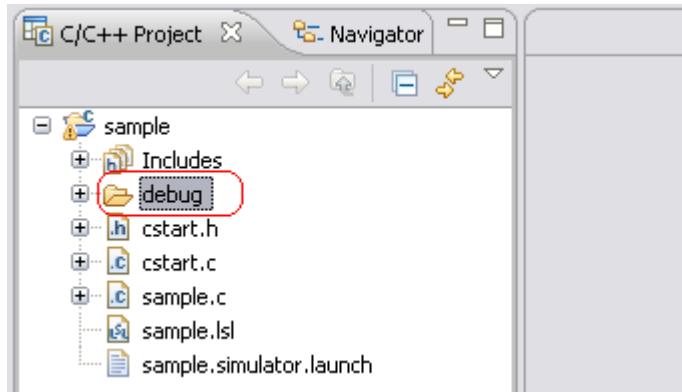


You may change and save any settings and rebuild the project. This will generate a number of files within the project directory. The next section provides a description on how to enter the necessary compiler and linker settings into the TESSY makefile template.

When building the project, the following error may occur within the TASKING EDE:



In this case, delete the debug node (refer to the figure below) from your project and rebuild the project.



The debug node will be created again during the rebuild process of your project.

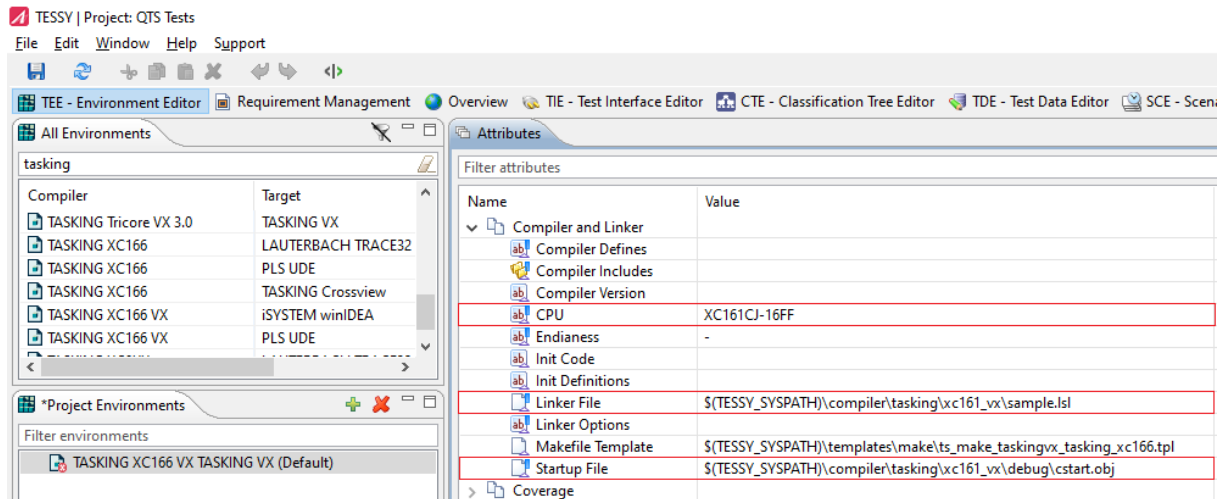
### 3.2 Adapting the TESSY makefile template

The following files within the TASKING EDE project directory are relevant for compilation and linking of the TESSY test driver application:

File	Description
sample.lsl	Contains the linker relevant information for memory layout etc. This file is referenced by the <b>Linker File</b> attribute within TEE.
debug\cstart.obj	Contains the compiled startup code for the device. This file is referenced by the <b>Startup File</b> attribute within TEE.
debug\makefile	Contains command line options to be passed to the linker. The settings within this file need to be copied into the respective TESSY makefile template.
debug\subdir.mk	Contains command line options to be passed to the compiler. The settings within this file need to be copied into the respective TESSY makefile template.

### 3.2.1 TEE settings

You may change the **CPU**, **Linker File** and **Startup File** attributes in order to compile and link with another TASKING EDE project located outside the TESSY installation directory. The picture below shows the default settings within TEE:



**Please note:** The **CPU** setting must be the same as for the processor type selected within the TASKING EDE project (refer to section 3.1). Have a look into the generated files described below for information about the exact microcontroller name that need to be provided within the **CPU** attribute (This name will be added as variable into the generated makefile).

### 3.2.2 Compiler command line options

The following red marked part of the command line is relevant for adaptation of the default TESSY settings.

The first part starting with “-C” contains the name of the selected microcontroller. This name needs to be specified as CPU attribute within TEE. The other options need to be added to the compiler options within the TESSY makefile template.

```

1 #####
2 # Automatically-generated file. Do not edit!
3 #####
4
5 C_FILES += "../cstart.c"
6
7 "cstart.d" : "../cstart.c"
8 "cstart.obj" : "../cstart.c"
9     @echo Compiling ${<F}
10     ["C:\Programme\TASKING\C166-VX v2.2r1\bin\cc166" -c -o "$@" "$<" -Cxc161cs -t -Wa-gAHLs] --e
11

```

**TASKING EDE file “debug\subdir.mk”**

Add these options to the makefile template into the makefile variable **EDE\_COMP\_OPTIONS** like shown below:

```

39 # SLAVE
40 # (You may change these settings to use your own compiler/linker options)
41 #
42
43 # Options to be copied from the Tasking EDE generated makefile
44 #
45 EDE_COMP_OPTIONS := -C$(CPU) -t -Wa-gAHLs --emit-locals=-equ,-symbols
46                 -Wa--error-limit=42 -Mn --iso=99 --language=+cmp1
47                 --compact-max-size=200 --max-call-depth=-1 -g

```

**TESSY makefile template**

### 3.2.3 Linker command line options

Do the same with the command line options of the linker within the file “debug\makefile”. Do not miss the options at the end of the very long linker command line as shown below (with word-wrapping activated).

```

22 # All Target
23 all : "sample.elf"
24
25 "sample.elf" : $(OBJ_FILES) $(EXT_LIBRARIES) $(LSL_FILES)
26 @echo Linking to $(@F)
27 @"C:\Programme\TASKING\C166-VX v2.2r1\bin\cc166" -o "$@" $(OBJ_FILES) $(EXT_LIBRARIES) -Cxc161c -t -Wl-OtZxycL -
28 file -Wl--map-file="$(PROJ).mapxml":XML (-mcrfik1SmNuQ) -Wl--error-limit=42 "-./$(PROJ).lsl" -g -Mn
29
30 -include "../makefile.targets"

```

#### TASKING EDE file “debug\makefile” (word-wrapped)

Add these options to the makefile template into the makefile variable **EDE\_LINK\_OPTIONS** like shown below:

```

37 # SLAVE
38 # (You may change these settings to use your own compiler/linker options)
39 #
40
41 # Options to be copied from the Tasking EDE generated makefile
42 #
43 EDE_COMP_OPTIONS      := -Cxc161cj -t -Wa-g&HLS --emit-locals=-equ,-symbols -Wa-
44                        -Wa--error-limit=42 -Mn --iso=99 --language=+cmp14,+di
45                        --compact-max-size=200 --max-call-depth=-1 -g
46
47 EDE_LINK_OPTIONS      := -Cxc161cj -t -Wl-OtZxycL --no-map-file -Wl--map-file="
48                        -mcrfik1SmNuQ) -Wl--error-limit=42 -g -Mn "$(LSL_FILE)"
49

```

#### TESSY makefile template

## 4 C166/XC166 (Classic)

The TASKING C166 release 7.5 r5 and later allows creating all settings required for compilation and linking within the EDE. There is a sample project within the TESSY installation (prepared for XC166):

```
... \sys\compiler\tasking\xc166\sample.pjt
```

Open this project for all changes to the processor, startup or memory layout configuration of your modules. Upon changes to the configuration, simply rebuild the project and copy the settings into the appropriate locations like described below.

### 4.1 Copy compiler/linker options

If you change the compiler settings, you need to copy these settings from the file

```
<project name>.mak
```

generated by the TASKING EDE (the makefile variables OPT\_CC and OPT\_LC).

```
PROJ      = sample
PROJDIR   = d:\tmp\tasking_c166_sample
PRODDIR   = c:\programme\c166v75
OUTDIR    =

OPT_CC    = -Wcp-I"c:\programme\c166v75\include" -Wc-I"c:\program
OPT_MPP   = @'_sample.inc' WA(1)
OPT_LC    = "_sample.ilo" -cf -trap -libfmtiol -Ms -xmifp2 EXTEND
```

Both these variables need to be copied into the corresponding makefile template (e.g. ... \sys\templates\make\ts\_make\_hitop\_tasking\_xc166.tpl).

The variables TASKING\_EDE\_OPT\_CC and TASKING\_EDE\_OPT\_LC shall contain these EDE settings.

```
# Options to be copied from the Tasking EDE generated makefile
#
TASKING_EDE_OPT_CC := -Wc-x2midfp -DCPUTYPE=0x1662 -Ms -Wc-FSC -noc++ -Wc-OB
TASKING_EDE_OPT_LC := "${TESSY_SYS_DOS}\compiler\tasking\xc166\_sample.ilo"
```

You need to add the path to the `_project name.ilo` entry within the TASKING\_EDE\_OPT\_LC variable.

## 4.2 Specify the startup code

Another file used from the EDE generated project will be the startup code. If you want to use another startup file, either change the EDE project or change the makefile template (variable `S_OBJECTS`) to use the respective file.

```
S_CC      := $(TASKING_HOME)\bin\cc166
S_LINK    := $(TASKING_HOME)\bin\cc166
S_OBJECTS := $(TESSY_SYS_DOS)\compiler\tasking\xc166\start.obj \
             $(MODULE_PATH_DOS)\tslows_hitop.obj \
             $(MODULE_PATH_DOS)\tstcomm.obj \
             $(MODULE_PATH_DOS)\ts_timer_c166.obj \
             $(MODULE_PATH_DOS)\tscom1.obj
```

## 5 Flashing the test binary

If you need to flash your test binary set the TEE attribute **Flash Program** to **true**.

<input checked="" type="checkbox"/> Flash Program	true
---------------------------------------------------	------

## 6 Disable the watchdog

The `MConfig` file of your TASKING VX project contains the essential watchdog setup for the TASKING VX debugger. This file has to be located in the same path as where the TASKING VX debugger is launched from. Thus the TEE attribute **Debugger Launch Path** was introduced to function as the launch path for the TASKING VX debugger. So **Debugger Launch Path** should point to the directory where the TASKING VX project's `MConfig` file is found. The default value is `$(PROJECTROOT)\tessy\config`.