

Metrowerks Compiler

Abstract

This document describes tips and tricks for the different Metrowerks compilers (HCS08/HCS12). Also refer to the application note concerning the respective target debugger.

Table of Contents

Abstract	1
1 Introduction.....	2
1.1 HCS12.....	2
1.1.1 TEE Settings	2
1.1.2 Reserving Memory Locations for Absolute Pointer Access	3

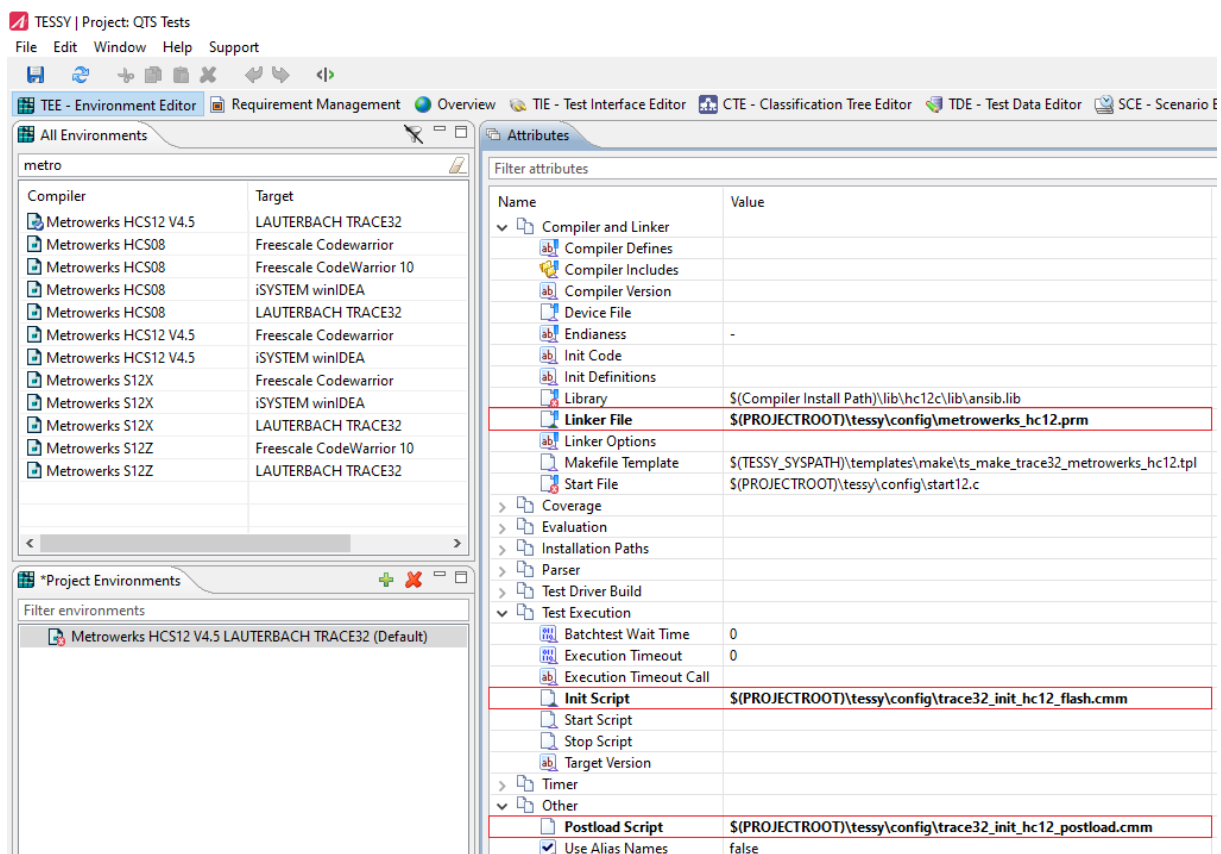
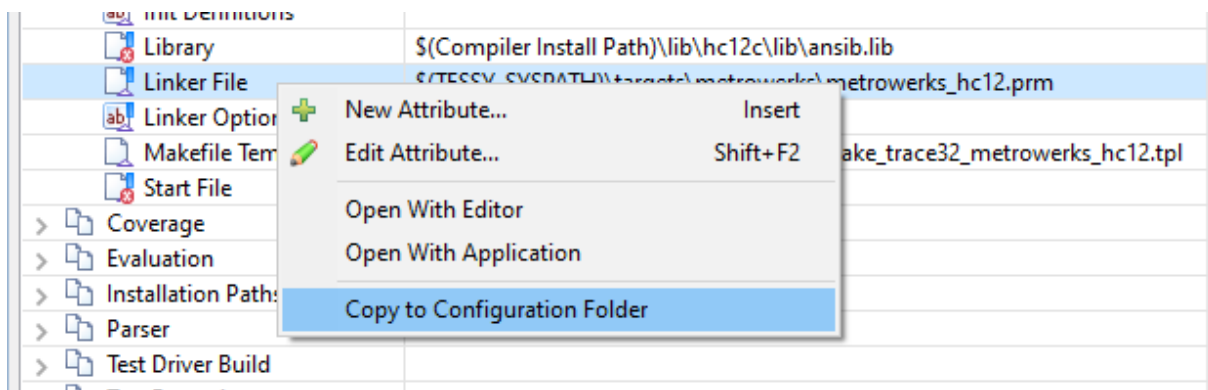
1 Introduction

1.1 HCS12

The Metrowerks compiler requires a linker file to be specified containing your section definitions and memory layout. The default linker file can be adapted or replaced by a linker file from your development project.

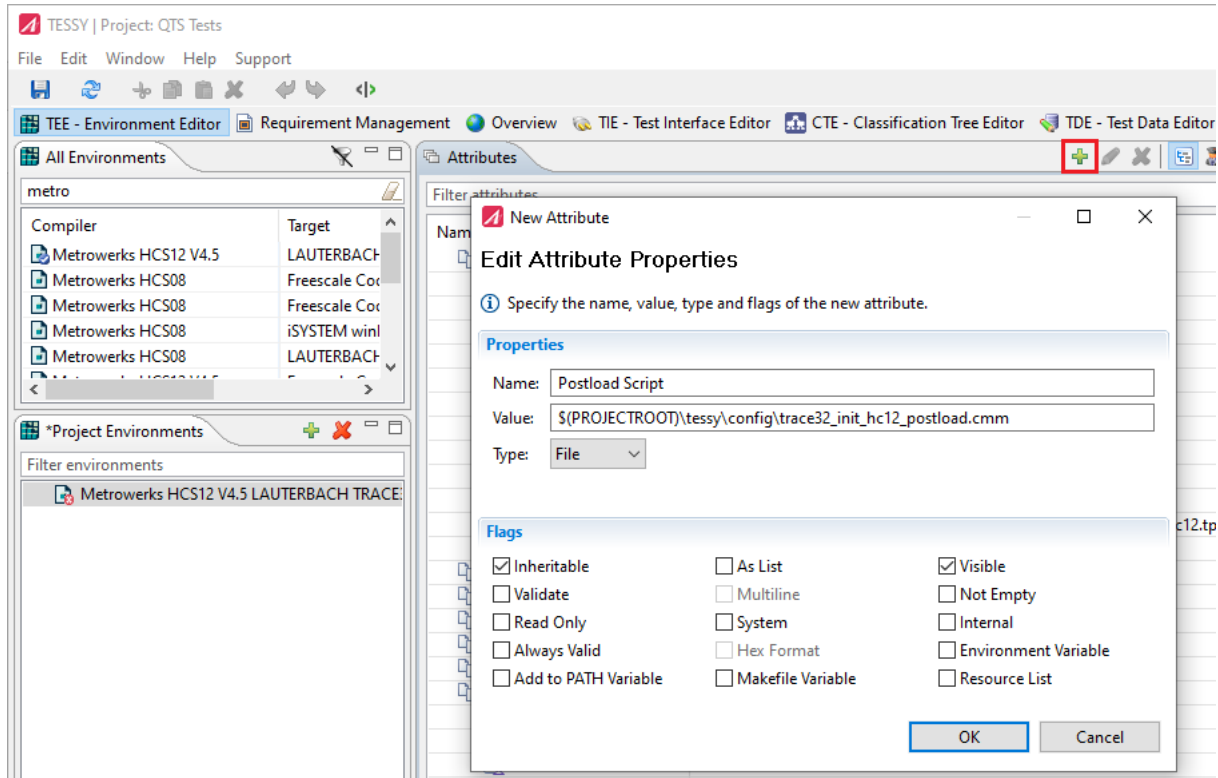
1.1.1 TEE Settings

The default attributes refer to the standard linker file. We would suggest to change the default settings and move your linker file into your test project file structure.



The above example shows a configuration for TRACE32 using scripts for flashing and the linker file located within the project directory structure.

The attribute **Postload Script** has to be added with the New Attribute button.



1.1.2 Reserving Memory Locations for Absolute Pointer Access

It is sometimes necessary to have a specific memory area available, that is located within a defined address range. In the example below, the target address space and the address value itself are both input values that have an effect on the result.

```

// XOR given value and address value
// and store the result into address cell
void store_value(unsigned short value,
                unsigned short *address)
{
    // XOR value and address
    value = value ^ (unsigned short)address;

    // Store result
    *address = value;
}

```

The address value is combined with the input value using an XOR operation. The result of this operation is stored at the location, the pointer points to. Therefore, the address pointer provides the location of the result and influences the result value.

A suitable input would be like follows:



This implies, that the address range starting from 0x0400 need to be reserved from the available linker memory. You may do this by modifying the linker file (Refer to the module attributes) like below:

```
SECTIONS
MY_RAM      = READ_WRITE 0x0450 TO 0x0BFF;
MY_ROM      = READ_ONLY  0x0C00 TO 0x1FFF;
```

This allows memory space beginning from 0x0400 up to 0x0450 to be used without conflicts. The test report below shows the input values and the evaluation of the result using an eval macro.

Testcase 1

Teststep 1.1 ✔			
Name	Input Value		
value	0x1122		
address	0x0400		
Name	Actual Value	Eval Expected Value	Result
XOR value	5410	== 5410	✔

Teststep C Epilog: TESSY_EVAL_U16("XOR value", *(unsigned short *)0x0400, ==, 0x0400 ^ 0x1122);

You may enter pointer address values directly within TDE. But be aware of the correct memory locations and pointer initializations to prevent erroneous memory accesses.