

Using setjmp/longjmp

Abstract

This document describes the usage of the setjmp/longjmp mechanism with Tessy. Some restrictions apply, if your software uses this feature of the C language, so that there may be situations, which are not supported by the proposed solution. Please make sure, that your target compiler supports this feature like described in this document.

Table of Contents

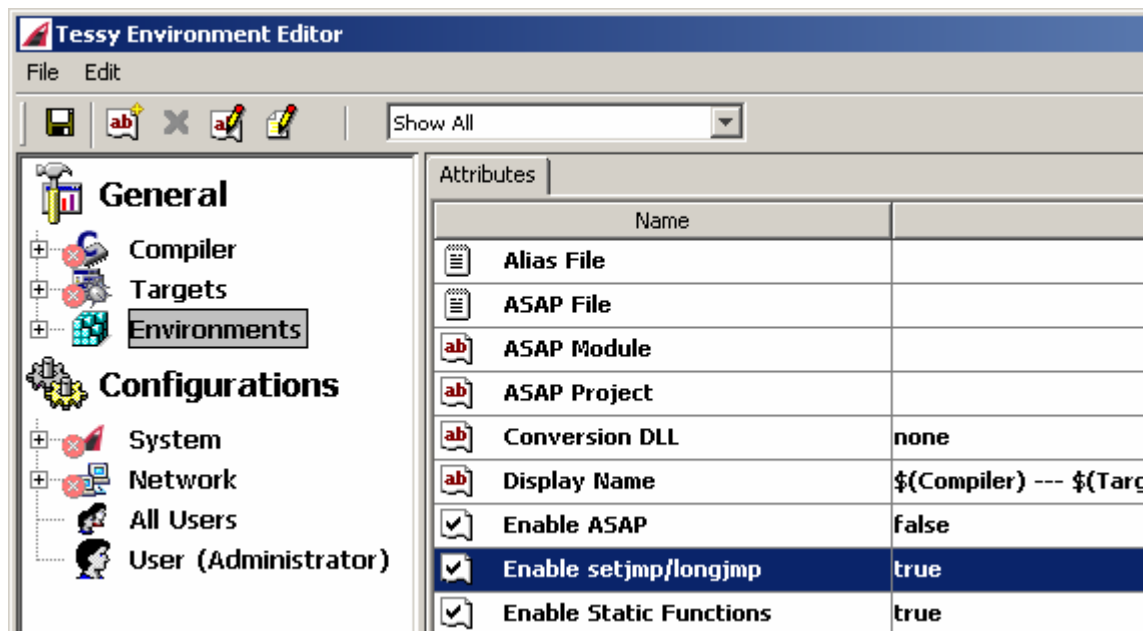
Abstract	1
Introduction.....	2
Enable the setjmp/longjmp Mechanism	2
Settings within UCE	3
Generated Test Driver Code.....	5

Introduction

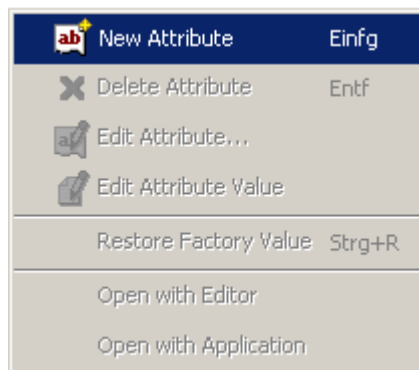
The setjmp/longjmp mechanism requires a special jmp_buf structure to be filled, in order to mark the jump target. This structure needs to be passed to the longjmp() call. Since your software will probably use another jmp_buf structure, this needs to be adapted for usage with Tessy. The solution described below hides the longjmp() call within an external function, that may be replaced by a stub function within Tessy.

Enable the setjmp/longjmp Mechanism

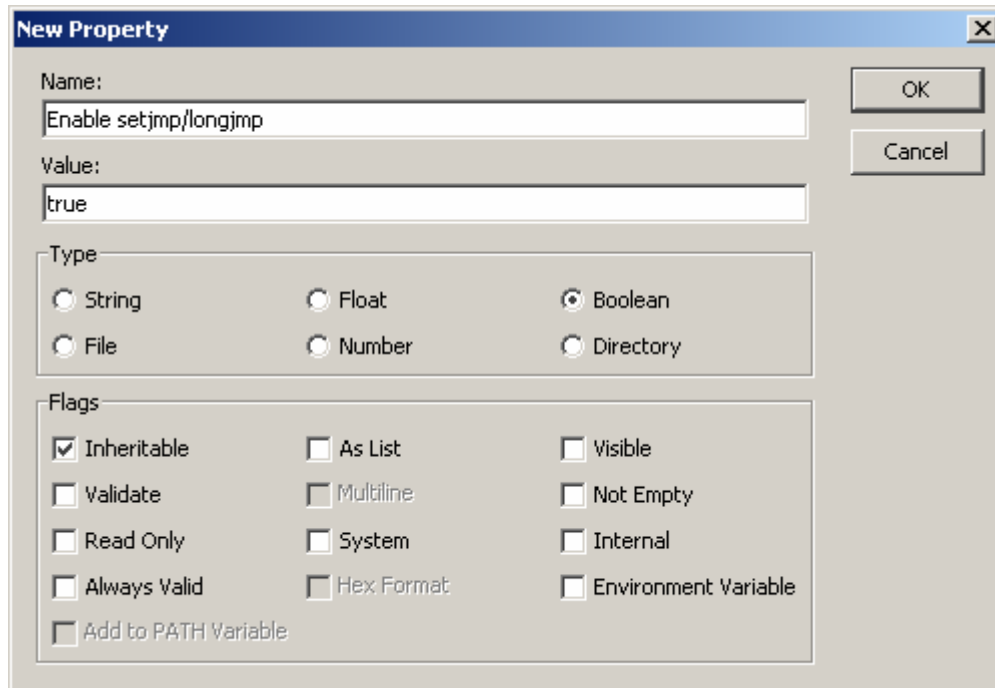
A special test driver needs to be created for usage with setjmp/longjmp. You need to enable Tessy to generate this special driver code within the **Environment Editor** as shown below:



Create a **new** attribute of type **Boolean** within the respective compiler section or within the global environments section as shown above. Choose **New Attribute** from the context menu:

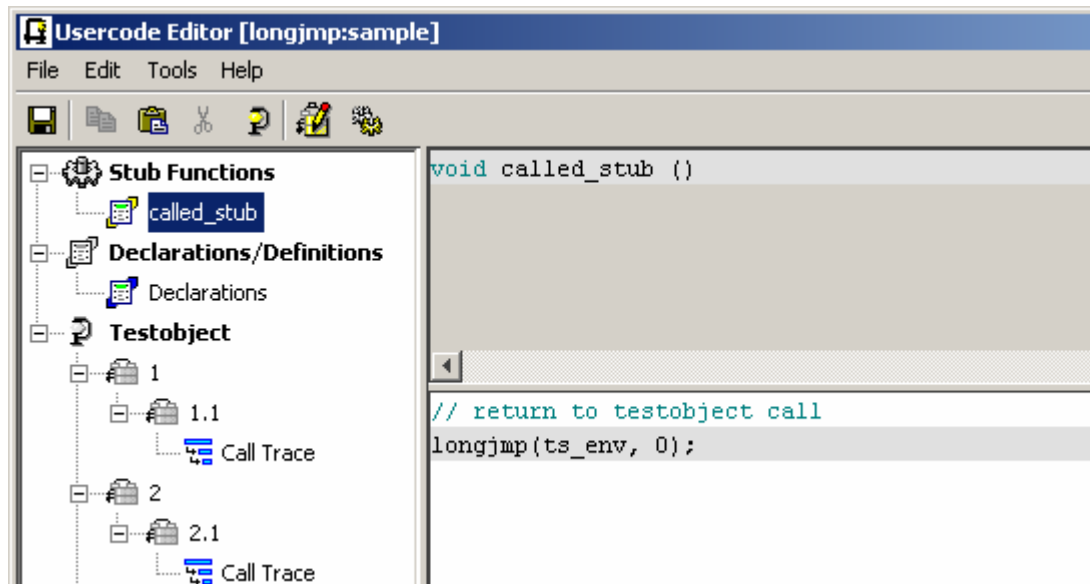


The following dialog shows the necessary settings:



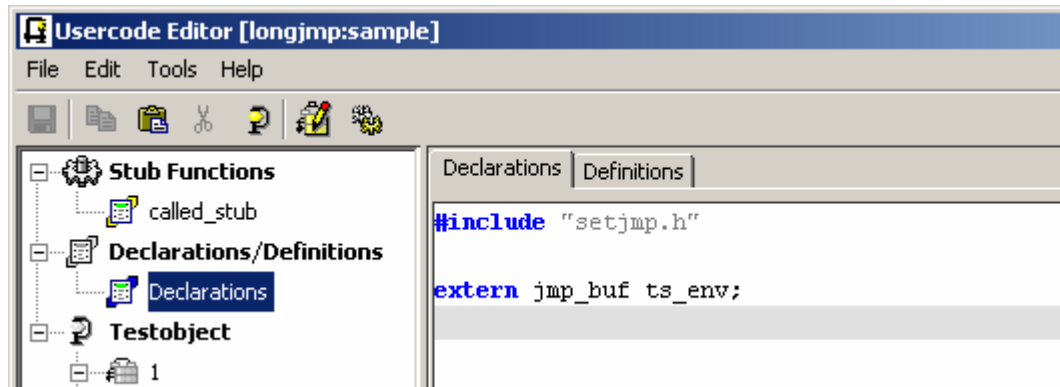
Settings within UCE

The global variable `ts_env` will be available for usage with `longjmp()` somewhere within your software or within a stub function. The following settings within the **Usercode Editor UCE** are required to use the `longjmp()` function.



```
longjmp(ts_env, 0);
```

The variable `ts_env` needs to be declared within the declarations section of the Usercode Editor UCE like shown below:



```
#include "setjmp.h"  
extern jmp_buf ts_env;
```

Generated Test Driver Code

The following code will be generated within the test driver in order to set the longjmp() return address:

```
176 void tessy_execute_task(void)
177 {
178     if (ts_mode == TS_MODE_TESTSTEP) {
179         while (TS_REPEAT_COUNT) {
180             ts_reset_stubcounter();
181             if (!setjmp(ts_env)) {
182                 sample_testobject();
183             }
184             TS_REPEAT_COUNT--;
185         }
186     }
187 }
```

When the testobject or any stub function calls the longjmp() function, the program execution will continue after the call to the testobject, resulting in a behavior as if the testobject was exited normally.