
Subject: Incredible interaction between graphics and DLM: SOS!

Posted by [fabio.tosetti](#) on Tue, 05 Aug 2008 15:27:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have a very simple file set (see below), with only one function, `idl_test()`, written just to isolate the problem.

The following instruction simply sleep for 20 seconds and correctly returns:

```
IDL> print, idl_test()
```

The following sequence cause the `idl_test()` to return BEFORE the expected time:

```
IDL> err = dialog_message("A dialog")
IDL> print, idl_test()
```

The `idl_test()` returns after about 3-4 seconds.

The same happens if I plot a graphic before calling `idl_test()`.

Of course in my application I don't want to `sleep()`, but to call some Python code that do remote communication (Pyro), but the result is exactly the same, and it's very frustrating :-(...

THE EXAMPLE FILES ARE:

1) `idl_test.dlm`:

```
MODULE idl_test
DESCRIPTION idl test library
VERSION 1.0
SOURCE LB
BUILD_DATE AUG 05 2008
```

```
FUNCTION IDL_TEST 0 0
```

2) `idl_test.h`:

```
#ifndef IDL_TEST_H_INCLUDE
#define IDL_TEST_H_INCLUDE

extern "C"{
    int IDL_Load(void);
}

#endif /*IDL_TEST_H_INCLUDE*/
```

2) idl_test.cpp file:

```
#include "idl_test.h"

#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

extern "C"{
    #include "idl_export.h"
}

IDL_VPTR idl_test(int lArgc, IDL_VPTR Argv[]) {

    // TEST
    int sec = 20;
    printf("Sleeping for %d seconds... will timeout earlier!\n", sec);
    sleep(sec);
    return IDL_StrToSTRING("idl_test returned before 20 seconds!!!");
}

int IDL_Load(void) {

    // These tables contain information on the functions and procedures
    // that make up the TESTMODULE DLM. The information contained in
    // these
    // tables must be identical to that contained in testmodule.dlm.
    //
    static IDL_SYSFUN_DEF2 function_addr[] = {

        { (IDL_SYSRTN_GENERIC) idl_test, "IDL_TEST", 0, 0, 0, 0},

    };

    // Register my routines: the routines must be specified exactly
    // the same
    // as in .dlm.
    return IDL_SysRtnAdd(function_addr, TRUE,
        IDL_CARRAYELTS(function_addr));
}
```

4) Makefile:

all: idl_test.so

ITT had the great idea of changing the name of
the installation directory since idl6.3

So if you have idl < 6.4 use

#IDLDIR = /usr/local/rsi/idl/external/include

else use

IDLDIR = /usr/local/itt/idl/external/include

CPPFLAGS = -Wall -W -Wreturn-type -Wunused -D_GNU_SOURCE

MORE_INCLUDE = -I\$(IDLDIR) -I/usr/include/python

%.o:%.h

.cpp.o:

g++ \$(CPPFLAGS) \$(MORE_INCLUDE) -c \$< -o \$@

idl_test.so: idl_test.o \$(IDLDIR)/idl_export.h

g++ \$(CPPFLAGS) -shared -o idl_test.so idl_test.o -lstdc++

clean:

rm -f *.o *.so
