

Visual Fortran/C++ 混合言語プログラミング

注意：本文書は、Visual C++ 6.0 と Visual Fortran 6.0/6.5 に基づくものです。

1. C++ → Fortran 呼び出し

C++では、例えば

```
extern "C" void _stdcall DGESV(int *N); // 必ず大文字
```

と宣言し、Fortran のルーチン

```
subroutine DGESV(N)
```

```
integer N
```

を呼び出す。「**extern "C"**」は、C++による名前の修飾（関数のオーバーロードを区別するため、名前に文字列を追加すること）を避けるために必須である（他の処理系も、多分同じ）。

リンカーの使う名前は、「**_DGESV@4**」である。ここで、4 は、スタックに積むバイト数。

2. C++ → CXML 呼び出し

上記呼び出し規則に従うとともに、**CXMLDLL.LIB** をリンクする。[Build]-[Project Setting](Alt-F7)の[Fortran]-[Library]-[Use CXML]をチェックしても、意味がない。VC のDeveloping Studio からでも、(VF をインストールしていれば)利用できる。

3. C++ → IMSL 呼び出し

上記呼び出し規則に従うとともに、**IMSL.LIB IMSLS_ERR.LIB IMSLMPISTUB.LIB**（最後はMPIを使用する場合のみ必要）をリンクする（方法は、CXML の場合と同じ）。

IMSL は、Fortran の内部ルーチンを使用し、**dfor.lib** と**LIBC.lib** で使用しているシンボル**__matherr**の多重定義を避けるため、以下の手順を踏むこと。

- (1) Fortran Console Application の empty project を作る。
- (2) ダミーの .f90 ファイルを「Source Files」に加え、ビルドする。
- (3) 本命の .cpp ファイルを「Source Files」に加える。
- (4) [Link]-[Object/Library Modules]に「IMSL.LIB IMSLS_ERR.LIB」を追加しビルドする。

「<http://www.compaq.co.jp/products/dvf/visual/faq.html#Linking>」から抜粋
リンクした時、「dformt.lib (matherr.obj) : error LNK2005: __matherr already defined in LIBCMT.lib (matherr.obj)」(2 番目のライブラリ名は変化する場合があります) というエラーでリンクが終了します。19-Jun-2001

Fortran 実行時ライブラリには、**_matherr** の再定義が含まれています。これにより、**math** ライブラリの組み込みルーチンを実行する時に発生する例外を処理することができるようになっていきます。C で

は、`_matherr`が再定義されていないと、C ライブラリは基本的には何もしない基本 `_matherr` の定義を使用します。アプリケーションをリンクする時、シンボル `_matherr` に関して「multiply defined symbol」というエラーをリンカが生成することがあります。これは、リンカがアプリケーションのシンボルを解決しようとする順番と`_matherr` シンボルを解決しようとした時の最後に検索したライブラリに依存します。シンボル `xxx` は、`libc`で解決され、リストの次のシンボルが `_matherr` であったとすると、リンカは `_matherr` を解決するために`libc` を最初に参照し、Fortran の再定義が得られず、リンカのエラーが発生します。Fortran ライブラリのモジュール `matherr.obj` には参照する他のシンボルも含まれているため、エラーが発生します。それが参照された時、リンカは `_matherr` の他の定義を参照し、エラーが発生します。

この問題を回避するには、Fortran ライブラリが C ライブラリよりも前に指定されていることを確認します。Developer Studio では、「Project」メニューから「Settings...」を選択し、「Link」タブをクリックします。表示される画面で、「Object/Library Modules」フィールドに示されているリストの先頭に「`dfor.lib`」（またはマルチスレッド・ライブラリを選択した場合は「`dformat.lib`」）を入力します。

（注意）上記は、参考になる記述であるが、この方法では、エラーを回避できない。

4. Fortran → C++ 呼び出し

以下の要領に従う。リンカーの使う名前は、`_Cmain` である。なお、「C++ → IMSL 呼び出し」で記述した手続きが必要かもしれない。

```
// for_to_C.cpp
#include<stdio.h>
extern "C" void Cmain()
{
    class A{public: char *str;} a;
    a.str = "this is from C++¥n";
    printf(a.str);
}
! for_to_C.f90
program fot_to_c
!subroutine a
    INTERFACE TO SUBROUTINE Cmain [C,ALIAS:'_Cmain']
    END
    write(*,*) "this is from F90"
    call Cmain
end
```