

Registration No.

23321



GNU C/C++ AND FORTRAN LANGUAGE INTEROPERABILITY WITH FUNCTION USAGE: PART II

Disclaimer: Reference herein to any specific commercial company, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the Department of the Army (DoA). The opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or the DoA, and shall not be used for advertising or product endorsement purposes.

UNCLASSIFIED: Distribution Statement A.

Approved for public release; distribution is unlimited.

September 2012

U.S. Army Tank Automotive Research,
Development, and Engineering Center
Detroit Arsenal
Warren, Michigan 48397-5000

GNU C/C++ AND FORTRAN LANGUAGE INTEROPERABILITY WITH FUNCTION USAGE: PART II

W. Bylsma

Dynamics and Structures
U.S. Army Research, Development and Engineering Command (RDECOM)
U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC)
Detroit Arsenal
ATTN: RDTA-RS/MS157
6501 East 11 Mile Road
Warren, Michigan 48397-5000

*** * * * ***

Disclaimer: Reference herein to any specific commercial company, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the Department of the Army (DoA). The opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or the DoA, and shall not be used for advertising or product endorsement purposes.

*** * * * ***

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 09-17-2012		2. REPORT TYPE TECHNICAL		3. DATES COVERED (From - To) 2012	
GNU C/C++ AND FORTRAN LANGUAGE INTEROPERABILITY WITH FUNCTION USAGE: PART II				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) WESLEY BYLSMA				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DYNAMICS AND STRUCTURES-US ARMY RDECOM/TARDEC ATTN: RDTA-RS/MS157 6501 E 11 MILE RD WARREN, MI 48397-5000				8. PERFORMING ORGANIZATION REPORT NUMBER 23321	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A: Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This report is a continuation of Technical Report (TR) 22646 and documents GNU and Intel Fortran/C++ compiler options for interoperability between C/C++ and FORTRAN. The actions addressed are manipulation of Fortran COMMON BLOCKS with C++ Structures and the usage of multiple definitions.					
15. SUBJECT TERMS GNU GCC, C/C++, FORTRAN, language interoperability, common block, structures, multiple definitions					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Unclassified	18. NUMBER OF PAGES 10	19a. NAME OF RESPONSIBLE PERSON Wesley Bylsma
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code) 586-282-4104

Contents (by Title)

PART II.....	1
4.4 FORTRAN COMMON BLOCK MANIPULATION USING C++ STRUCTURES.....	1
4.4.1 GNU.....	1
4.4.2 INTEL	3
4.5 MULTIPLE DEFINITIONS	3
4.5.1 GNU.....	3
4.5.2 INTEL	7
REFERENCES.....	10

GNU C/C++ AND FORTRAN LANGUAGE INTEROPERABILITY WITH FUNCTION USAGE: PART II

TARDEC Technical Report No. 23321

September 2012

PART II

This report is a continuation of Technical Report (TR) 22646 and documents GNU and Intel Fortran/C++ compiler options for interoperability between C/C++ and FORTRAN. The actions addressed are manipulation of Fortran COMMON BLOCKS with C++ Structures and the usage of multiple definitions.

All examples were run on SUSE Linux Enterprise Server 11 (x86_64) using the GNU utilities as in TR 22646---with the addition of

- xild (GNU ld (GNU Binutils; SUSE Linux Enterprise 11) 2.20.0.20100122-0.7.9)
- ifort 10.1 20080312
- icc 10.1 20080312

References cited are [8] (from TR 22646) and [17] (new for TR 23321).

4.4 FORTRAN COMMON BLOCK MANIPULATION USING C++ STRUCTURES

This example shows how to call C++ from FORTRAN and the usage of the C++ structures to manipulate FORTRAN COMMON blocks (see [8]).

4.4.1 GNU

The GNU Fortran/C++ compiler is used here.

```
echo "---GNU---"
rm main.o
gfortran -m32 -c main.f -lgfortran
rm cpps.o
gcc -m32 -c cpps.cpp -lstdc++
rm pstu.e
gfortran -m32 main.o cpps.o -o pstu.e -lstdc++
pstu.e
```

cpps.cpp

```
#include <iostream>
#include <cstring>
using namespace std;

extern "C" {
    extern struct {
        double x;
        int a,b,c;
        char s10[10];
    } abc_;

    void cpps_() {
        cout << "cpps: begin" << endl;
        abc_.x = 78.9998;
        abc_.a = abc_.b*abc_.c;
        abc_.b = 101;
        abc_.c = -54;
        strncpy(abc_.s10,"CHANGE_ALL",10);
        cout << " x:" << abc_.x << endl;
        cout << " a,b,c:" << abc_.a << "," << abc_.b << "," << abc_.c << endl;
        cout << " s10:" << abc_.s10 << endl;
        cout << "cpps: end" << endl;
    }
}
```

main.f

```
PROGRAM main
DOUBLE PRECISION X
INTEGER A,B,C
CHARACTER*10 S10
COMMON /ABC/X,A,B,C,S10
print *, 'main: begin'
X = 55.39
A = 1
B = 2
C = 5
S10 = 'HOW_ARE_U?'
print *, 'X:',X
print *, 'A,B,C:',A,B,C
print *, 'S10:',[' ',S10,']'
CALL cpps()
print *, 'X:',X
print *, 'A,B,C:',A,B,C
print *, 'S10:',[' ',S10,']'
print *, 'main: end'
END
```

OUTPUT:

```
---GNU---
main: begin
X: 55.389999389648438
A,B,C: 1 2 5
S10:[HOW_ARE_U?]
cpps: begin
x:78.9998
a,b,c:10,101,-54
s10:CHANGE_ALL
cpps: end
X: 78.999799999999993
A,B,C: 10 101 -54
S10:[CHANGE_ALL]
main: end
```

4.4.2 INTEL

The Intel Fortran/C++ compiler is used here. The `cpps.cpp` and `main.f` files are the same as in Section 4.4.1.

```
#!/usr/bin/tcsh
# Intel Compiler
source /apps/intel/fc/10.1.015/bin/ifortvars.csh
source /apps/intel/cc/10.1.015/bin/iccvars.csh

echo "---INTEL---"
rm main.o
ifort -m32 -c main.f
rm cpps.o
icc -m32 -c cpps.cpp -L/usr/lib -lstdc++
rm pstu.e
ifort -m32 main.o cpps.o -o pstu.e -L/usr/lib -lstdc++
pstu.e
```

OUTPUT:

```
---INTEL---
main: begin
X: 55.3899993896484
A,B,C: 1 2 5
S10:[HOW_ARE_U?]
cpps: begin
x:78.9998
a,b,c:10,101,-54
s10:CHANGE_ALL
cpps: end
X: 78.99980000000000
A,B,C: 10 101 -54
S10:[CHANGE_ALL]
main: end
```

The differences between the GNU and Intel floating point numbers should be noted.

4.5 MULTIPLE DEFINITIONS

This example shows how to use multiple definitions to modify already compiled object files. The `cpps.cpp` and `main.f` files are the same as in Section 4.4.1.

4.5.1 GNU

The GNU Fortran/C++ compiler is used here. Referencing [17], there are 3 file descriptors, `stdin`, `stdout` and `stderr` (`std=standard`). The number 1 'represents' `stdout` and 2 `stderr`. An example would be

```
cat file 2 >& 1
```

Proper output order when redirecting is maintained by setting

```
setenv GFORTRAN_UNBUFFERED_ALL Y
```

Using “`gfortran -help`” comma-separated options are passed to the linker as “`-Wl,<options>`”. Using “`ld -help`” the linker option “`-z muldefs`” allows multiple definitions.

```

#!/usr/bin/tcsh
setenv GFORTRAN_UNBUFFERED_ALL Y
echo "----GNU----"

# make original
rm main.o
gfortran -m32 -c ../../struct/gnu/main.f -lgfortran
rm cpps.o
gcc -m32 -c ../../struct/gnu/cpps.cpp -lstdc++
rm pstu.e
ld -V -m elf_i386 -r main.o cpps.o -o pstu.o
gfortran -m32 pstu.o -o pstu.e -lstdc++
echo "----main with cpps"
pstu.e

# make update
rm main1.o
gfortran -m32 -c main1.f -lgfortran
rm cpps1.o
gcc -m32 -c cpps1.cpp -lstdc++

# replace main with main1
echo "----replace main with main1"
gfortran -m32 main1.o pstu.o -o pstu1.e -lstdc++
echo "----replace main with main1"
rm pstu1.e
gfortran -m32 main1.o pstu.o -o pstu1.e -lstdc++ -Wl,-z,muldefs
pstu1.e

# replace cpps with cpps1
echo "----replace cpps with cpps1"
gfortran -m32 cpps1.o pstu.o -o pstu2.e -lstdc++
echo "----replace cpps with cpps1"
rm pstu2.e
gfortran -m32 cpps1.o pstu.o -o pstu2.e -lstdc++ -Wl,-z,muldefs
pstu2.e

# replace main and cpps
echo "----replace main/cpps with main1/cpps1"
gfortran -m32 main1.o cpps1.o pstu.o -o pstu3.e -lstdc++
echo "----replace main/cpps with main1/cpps1"
rm pstu3.e
gfortran -m32 main1.o cpps1.o pstu.o -o pstu3.e -lstdc++ -Wl,-z,muldefs
pstu3.e

```


cpps1.cpp

```
#include <iostream>
#include <cstring>
using namespace std;

extern "C" {
    extern struct {
        double x;
        int a,b,c;
        char s10[10];
    } abc_;

    void cpps_() {
        cout << "cpps1: begin" << endl;
        abc_.x = 90023.876;
        abc_.a = 23.97;
        abc_.b = 1.0004;
        abc_.c = -999.9;
        strncpy(abc_.s10,"ALLCHANGED",10);
        cout << " x:" << abc_.x << endl;
        cout << " a,b,c:" << abc_.a << "," << abc_.b << "," << abc_.c << endl;
        cout << " s10:" << abc_.s10 << endl;
        cout << "cpps1: end" << endl;
    }
}
```

main1.f

```
PROGRAM main
DOUBLE PRECISION X
INTEGER A,B,C
CHARACTER*10 S10
COMMON /ABC/X,A,B,C,S10
print *, 'main1: begin'
X = 888.2954
A = -4.307
B = 2.3571
C = 67.9
S10 = 'U_ARE_FINE'
print *, 'X:',X
print *, 'A,B,C:',A,B,C
print *, 'S10:', '[' ,S10, ']'
CALL cpps()
print *, 'X:',X
print *, 'A,B,C:',A,B,C
print *, 'S10:', '[' ,S10, ']'
print *, 'main1: end'
END
```

OUTPUT:

```
---GNU---
GNU ld (GNU Binutils; SUSE Linux Enterprise 11) 2.20.0.20100122-0.7.9
Supported emulations:
  elf_x86_64
  elf_i386
  i386linux
  elf_llom
---main with cpps
main: begin
X: 55.389999389648438
A,B,C: 1 2 5
S10:[HOW_ARE_U?]
cpps: begin
x:78.9998
a,b,c:10,101,-54
s10:CHANGE_ALL
cpps: end
X: 78.999799999999993
A,B,C: 10 101 -54
S10:[CHANGE_ALL]
main: end
---replace main with main1
pstu.o: In function `MAIN__':
(.text+0x0): multiple definition of `MAIN__'
main1.o:main1.f:(.text+0x0): first defined here
collect2: ld returned 1 exit status
---replace main with main1
rm: cannot remove `pstu.e': No such file or directory
main1: begin
X: 888.29541015625000
A,B,C: -4 2 67
S10:[U_ARE_FINE]
cpps: begin
x:78.9998
a,b,c:134,101,-54
s10:CHANGE_ALL
cpps: end
X: 78.999799999999993
A,B,C: 134 101 -54
S10:[CHANGE_ALL]
main1: end
---replace cpps with cpps1
pstu.o: In function `cpps_':
(.text+0x57c): multiple definition of `cpps_'
cpps1.o:cpps1.cpp:(.text+0x5c): first defined here
collect2: ld returned 1 exit status
---replace cpps with cpps1
rm: cannot remove `pstu2.e': No such file or directory
main: begin
X: 55.389999389648438
A,B,C: 1 2 5
S10:[HOW_ARE_U?]
cpps1: begin
x:90023.9
a,b,c:23,1,-999
s10:ALLCHANGED
cpps1: end
X: 90023.876000000004
A,B,C: 23 1 -999
S10:[ALLCHANGED]
main: end
```

```

---replace main/cpps with main1/cpps1
pstu.o: In function `cpps_':
(.text+0x57c): multiple definition of `cpps_'
cpps1.o:cpps1.cpp:(.text+0x5c): first defined here
pstu.o: In function `MAIN__':
(.text+0x0): multiple definition of `MAIN__'
main1.o:main1.f:(.text+0x0): first defined here
collect2: ld returned 1 exit status
---replace main/cpps with main1/cpps1
rm: cannot remove `pstu3.e': No such file or directory
main1: begin
X: 888.29541015625000
A,B,C: -4 2 67
S10:[U_ARE_FINE]
cpps1: begin
x:90023.9
a,b,c:23,1,-999
s10:ALLCHANGED
cpps1: end
X: 90023.8760000000004
A,B,C: 23 1 -999
S10:[ALLCHANGED]
main1: end
service0 328%

```

4.5.2 INTEL

The Intel Fortran/C++ compiler is used here. The files are the same as in Section 4.5.1. Using “ifort -help” comma-separated options are passed to the linker as “-Wl,<o1>[,<o2>,...]”. Using “xild -help” the linker option “-z muldefs” allows multiple definitions.

```

#!/usr/bin/tcsh

# Intel Compiler
source /apps/intel/fc/10.1.015/bin/ifortvars.csh
source /apps/intel/cc/10.1.015/bin/iccvars.csh

echo "----INTEL----"

# make original
rm main.o
ifort -m32 -c ../../struct/intel/main.f
rm cpps.o
icc -m32 -c ../../struct/intel/cpps.cpp
rm pstu.e
xild -V -m elf_i386 -r main.o cpps.o -o pstu.o
ifort -m32 pstu.o -o pstu.e -L/usr/lib -lstdc++
echo "----main with cpps"
pstu.e

# make update
rm main1.o
ifort -m32 -c main1.f
rm cpps1.o
icc -m32 -c cpps1.cpp

# replace main with main1
echo "----replace main with main1"
ifort -m32 main1.o pstu.o -o pstu1.e -L/usr/lib -lstdc++
echo "----replace main with main1"
rm pstu1.e
ifort -m32 main1.o pstu.o -o pstu1.e -L/usr/lib -lstdc++ -Wl,-z,muldefs
pstu1.e

# replace cpps with cpps1

```

```

echo "----replace cpps with cpps1"
ifort -m32 cpps1.o pstu.o -o pstu2.e -L/usr/lib -lstdc++
echo "----replace cpps with cpps1"
rm pstu2.e
ifort -m32 cpps1.o pstu.o -o pstu2.e -L/usr/lib -lstdc++ -Wl,-z,muldefs
pstu2.e

# replace main and cpps
echo "----replace main/cpps with main1/cpps1"
ifort -m32 main1.o cpps1.o pstu.o -o pstu3.e -L/usr/lib -lstdc++
echo "----replace main/cpps with main1/cpps1"
rm pstu3.e
ifort -m32 main1.o cpps1.o pstu.o -o pstu3.e -L/usr/lib -lstdc++ -Wl,-z,muldefs
pstu3.e

```

OUTPUT:

```

---INTEL---
xild: warning empty_IPO_object_list: empty multi-file optimizations object list
xild: executing 'ld'
GNU ld (GNU Binutils; SUSE Linux Enterprise 11) 2.20.0.20100122-0.7.9
  Supported emulations:
    elf_x86_64
    elf_i386
    i386linux
    elf_llom
---main with cpps
main: begin
X: 55.3899993896484
A,B,C: 1 2 5
S10:[HOW_ARE_U?]
cpps: begin
x:78.9998
a,b,c:10,101,-54
s10:CHANGE_ALL
cpps: end
X: 78.9998000000000
A,B,C: 10 101 -54
S10:[CHANGE_ALL]
main: end
---replace main with main1
pstu.o: In function `MAIN__':
(.text+0x0): multiple definition of `MAIN__'
main1.o:main1.f:(.text+0x0): first defined here
---replace main with main1
rm: cannot remove `pstu1.e': No such file or directory
main1: begin
X: 888.295410156250
A,B,C: -4 2 67
S10:[U_ARE_FINE]
cpps: begin
x:78.9998
a,b,c:134,101,-54
s10:CHANGE_ALL
cpps: end
X: 78.9998000000000
A,B,C: 134 101 -54
S10:[CHANGE_ALL]
main1: end

```

```

---replace cpps with cpps1
pstu.o: In function `cpps_':
(.text+0x45a): multiple definition of `cpps_'
cpps1.o:cpps1.cpp:(.text+0x22): first defined here
---replace cpps with cpps1
rm: cannot remove `pstu2.e': No such file or directory
main: begin
X: 55.3899993896484
A,B,C: 1 2 5
S10:[HOW_ARE_U?]
cpps1: begin
x:90023.9
a,b,c:23,1,-999
s10:ALLCHANGED
cpps1: end
X: 90023.8760000000
A,B,C: 23 1 -999
S10:[ALLCHANGED]
main: end
---replace main/cpps with main1/cpps1
pstu.o: In function `cpps_':
(.text+0x45a): multiple definition of `cpps_'
cpps1.o:cpps1.cpp:(.text+0x22): first defined here
pstu.o: In function `MAIN__':
(.text+0x0): multiple definition of `MAIN__'
main1.o:main1.f:(.text+0x0): first defined here
---replace main/cpps with main1/cpps1
rm: cannot remove `pstu3.e': No such file or directory
main1: begin
X: 888.295410156250
A,B,C: -4 2 67
S10:[U_ARE_FINE]
cpps1: begin
x:90023.9
a,b,c:23,1,-999
s10:ALLCHANGED
cpps1: end
X: 90023.8760000000
A,B,C: 23 1 -999
S10:[ALLCHANGED]
main1: end

```

REFERENCES

- [8] <http://www.yolinux.com/TUTORIALS/LinuxTutorialMixingFortranAndC.html>
- [17] <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO-3.html>