

INCLUDE FORTRAN Compiler Directive Translation

CLAS-NOTE 90-012

Donald Joyce, CEBAF

Oct 1, 1990

Abstract

A set of utility programs have been written to translate the FORTRAN compiler directive "INCLUDE" statements. This translation is available for both the machine specific directive and the following file path. The programs are easily modified for other translations.

Introduction

Until recently, almost all code being developed for the CLAS collaboration was being written on DEC VMS based machines in FORTRAN. Most of this code takes good advantage of the "INCLUDE" directive for the FORTRAN compiler. *CLAS NOTE 90-008* specified that such include directives make use of VMS DCL logical definitions for include file paths.

At present, members of the CLAS collaboration are either starting to use, or considering the use of computers other than VAX VMS machines. Although, such machines come with FORTRAN compilers which support the "INCLUDE" directive, this directive can take a form different from that on VMS machines. For example, the VMS FORTRAN compiler supports the statement:

INCLUDE path : filename

whereas, the ULTRIX FORTRAN compiler supports the statement:

\$INCLUDE path/filename

Although, the above illustrated difference appears trivial, machine dependent changes to code should be automated to avoid introduction of unintended error. To port CLAS code to various machines the programs *UPDAT.TBL* and *INC.TRAN* were written. *UPDAT.TBL* maintains machine specific translation tables, whereas *INC.TRAN* modifies "INCLUDE" statements in FORTRAN code to be compatible with target compilers. Both programs are intended to be used with a "MAKE" utility.

Both utility programs are written in *ANSI C* hopefully making them easy to port to all intended computer platforms.

UPDAT.TBL

The *C* program *UPDAT.TBL* is utilized to maintain a machine specific translation table. *UPDAT.TBL* can be used with either a standard CLAS-standard translation table or a combination of standard and personal translation tables. If a personal translation table is used, personal entries supercede duplicate CLAS-standard translations. Use of personal translation tables is

intended for code development work, whereas CLAS-standard tables are utilized primarily for code porting. *UPDAT_TBL* can also be utilized to generate ascending order translation tables from raw translation tables.

UPDAT_TBL requires at least two arguments or parameters, and a maximum of three. The first argument is the path and name of the CLAS-standard translation table, and the second is the path and name for the output of *UPDAT_TBL*. The third, and optional argument is the path and name of the personal translation table. For example on a VMS system one may use:

```
UPDAT_TBL CLAS_TBL:CLAS_TRAN.TXT MY_ROOT:CLAS_TRAN.TXT MY_ROOT:TEMP_TRAN.TXT
```

In this example, the standard translation is read from *CLAS_TBL:CLAS_TRAN.TXT* and is added to and or superceded by the table contained in *MY_ROOT:TEMP_TRAN.TXT* . The result of using *UPDAT_TBL* is then written to *MY_ROOT:CLAS_TRAN.TXT* . The file *MY_ROOT:CLAS_TRAN.TXT* would then be used by the utility *INC_TRAN* to insert machine specific "INCLUDE" STATEMENTS in the FORTRAN code being ported.

Table Format

Translation tables require a specific form for the programs *UPDAT_TBL* and *INC_TRAN* to work properly. Translation tables are in the form of ASCII text files wherein each line contains both a search item and a substitution item. For example, the line:

```
'celeg$cmn:common.cmn' 'groups:[clas.new.src.celeg.cmn]common.cmn'
```

In this example *celeg\$cmn:common.cmn* is the search item, while the latter term, *groups:[clas.new.src.celeg.cmn]common.cmn* is the substitution item. Each standard file must also contain a search item for the *INCLUDE* directive, and a compiler specific substitution. For example, if one were to port code to HP UNIX machines, one would include the line:

```
INCLUDE #INCLUDE
```

Given this line, each occurrence of a translatable include statement results in the substitution of *#INCLUDE* at the appropriate location. A short sample of a translation table is included in appendix A.

INC_TRAN

The C program *INC_TRAN* is used to modify FORTRAN source code such that the code contains *INCLUDE* compiler directives appropriate for compiler and target operating system. In operation, *INC_TRAN* reads a code file, line by line and searches for lines containing the word *include* . Upon finding this, *INC_TRAN* searches the remainder of the line for a match with a search entry in the translation table. If a match is found, the existing line is commented out, and the appropriate line is substituted, with the proper compiler directive. If a match is not found the existing line remains unchanged and a comment is inserted in the code, stating that a match was not found.

INC_TRAN requires three arguments for parameters. The first argument is the path and name of the translation table. The second argument is the path and name of the FORTRAN file which is to be modified, while the third argument is the path and name of the result. For example:

```
INC_TRAN MY_ROOT:TEMP_TRAN.TXT CELEG$EVENT_GEN:INIT.FOR INIT.F
```

Here *MY_ROOT:TEMP_TRAN.TXT* is the translation file, *CELEG\$EVENT_GEN:INIT.FOR* is the file needing modification, and *INIT.FOR* is the output file.

Use of MAKE Utility

Both programs are intended for use with the *UNIX MAKE* utility. Versions of this utility are also available for VMS, VM, MVS, MSDOS, etc. Use of *MAKE* assures that translation tables are current, and that new modifications are accounted for with all code being ported. The following illustrates a typical "makefile" incorporating the programs *UPDAT.TBL* and *INC_TRAN*:

```
my_root:clas_tran.txt:          clas_tbl:clas_tran.txt\  
                                my_root:temp_tran.txt  
    $ updat_tbl clas_tbl:clas_tran.txt my_root:clas_tran.txt my_root:temp_tran.txt  
  
test_exam.f                   test_exam.for\  
                                my_root:clas_tran.txt  
    $ inc_tran my_root:clas_tran.txt test_exam.for test_exam.f  
  
test_exam.obj:                test_exam.f  
    $ for/object=test_exam.obj test_exam.f  
  
test_exam.exe:                test_exam.obj  
    $ link/exe=test_exam.exe test_exam.obj  
    $ delete test_exam.f  
    $ delete test_exam.obj
```

This example is based on the VMS version of *MAKE*. In examining this file, the executable *test_exam.exe* has the dependency *test_exam.obj*. This object file in turn has the dependency *test_exam.f*. This FORTRAN file is the result of translating the file *test_exam.for*. As illustrated in the *makefile* listing above, *test_exam.f* has two dependencies, namely *test_exam.for* and *my_root:clas_tran.txt*. Therefore, prior to translating *test_exam.for* the status of the translation file is checked by the *MAKE* utility. This translation file also has two dependencies: *clas_tbl:clas_tran.txt* and *my_root:temp_tran.txt*. These represent the standard translation file and a personal translation file. To utilize the above illustrated *makefile* and create the executable, one would enter on the system prompt

```
MAKE TEST_EXAM.EXE
```

In general use, each FORTRAN subroutine or function is compiled using *MAKE* with the proper translation table dependencies specified. Very often, files included into FORTRAN codes, themselves have *include* statements. A common example is to *include* parameter statements into common block files. Therefore, one should remember to also code these dependencies into the *makefile*.

Source Code Location

Source code for *INC_TRAN* and *UPDAT_TBL* as well as the VMS makefile, are located in the CLAS directories on the CEBAF VAXes. These directories are accessible for those persons logging into these machines by entering the DCL command:

```
SETUP CLAS.NEW or SETUP CLAS.PRO
```

and then looking in the directory *INC_TRAN\$SRC*. Those persons accessing the CEBAF VAX'es via FTP or DECNET will find the files located in *GROUPS: [CLAS.UTILITY.INC_TRAN]*.

Appendix A

This appendix is attached to illustrate the effect of the programs *UPDAT_TBL* and *INC_TRAN*.

One begins with the sample FORTRAN program *test_exam.for*. This program is rather simple:

```
PROGRAM test_exam
C-----
C-
C- Purpose and Methods : illustration only
C-
C- Libraries required  : none
C- Inputs              : none
C- Outputs             : none
C- Controls           : none
C-
C- Created 01-OCT-1990 Donald Joyce, CEBAF
C-
C-----
IMPLICIT NONE
C-----
INCLUDE 'LOST$DISK:UNKNOWN.PAR'
INCLUDE 'CELEG$CMM:COMMON.CMM'
INCLUDE 'MY_ROOT:TEST_EXAM.CMM'
CALL exit
END
```

This program contains three *INCLUDE* statements, which the program *INC_TRAN* recognizes, and will translate, if a translation exists.

Now one needs to examine the example translation tables. a portion of the CLAS-standard translation table for VMS to ULTRIX is illustrated below:

```
include $include
'celeg$cmm:common.cmm' 'usr/groups/clas/new/src/celeg/cmm/common.cmm'
```

A personal translation file might take the form:

```
include #include
'my_root:test_exam.cmn' 'usr/whatshisname/test/test_exam.cmn'
```

Here the personal translation file has an entry which supercedes one in the standard file, namely the translation for the *INCLUDE* compiler directive. For Ultrix, both forms happen to be acceptable.

Running the program *UPDAT_TBL* results in the following translation file:

```
'celeg$cmn:common.cmn' 'usr/groups/clas/new/src/celeg/cmn/common.cmn'
'my_root:test_exam.cmn' 'usr/whatshisname/test/test_exam.cmn'
include #include
```

Finally, running the program *INC_TRAN* with the use of this translation file results in the following modified FORTRAN file:

```
PROGRAM test_exam
C-----
C-
C- Purpose and Methods : illustration only
C-
C- Libraries required : none
C- Inputs : none
C- Outputs : none
C- Controls : none
C-
C- Created 01-OCT-1990 Donald Joyce, CEBAF
C-
C-----
IMPLICIT NONE
C-----
C*****
C following include statement without translation
C*****
INCLUDE 'LOST$DISK:UNKNOWN.PAR'
C*****
C*** line substitution by INC_TRAN
C--> INCLUDE 'CELEG$CMN:COMMON.CMN'
C*****
#include 'usr/groups/clas/new/src/celeg/cmn/common.cmn'
C*****
C*** line substitution by INC_TRAN
C--> INCLUDE 'MY_ROOT:TEST_EXAM.CMN'
C*****
#include 'usr/whatshisname/test/test_exam.cmn'
CALL exit
END
```

Here the include statements for 'CELEG\$CMN:COMMON.CMN' and 'MY_ROOT:TEST_EXAM.CMN' are properly translated. However, no translation exists for 'LOST\$DISK:UNKNOWN.PAR'. Therefore this compiler directive is flagged as being untranslated.