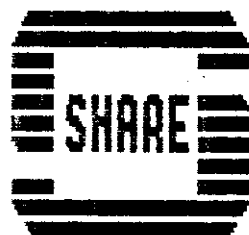


SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

122010

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257

SHARE PROGRAM LIBRARY SUBMITTAL FORMSHARE PROGRAM LIBRARY AGENCY

Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, North Carolina
27709 USA

Attention: Mr. Joe Ragland

SPLACONTROL NUMBER:

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the "SHARE Program Library Standards Manual".

- (1) Program Number (to be filled in by SPLA) 360D-12.2.010
(2) System Type (machine) 360
(3) Search Key CDC to IBM FORTRAN Conversion

- (4) Programming Language SNOBOL4
(5) Author's Name and Address Mr. George Gorsline, Jr.
UTLAS
130 St. George Street
Toronto, Ontario
Canada M5S 1A5

- (6) Direct Inquiries to Name and Address
(if different than Author)

- (7) Title of Program CONVERT - Source conversion of CDC
FORTRAN to IBM "G level"

- (8) Submitter's Installation Membership Code..... OBC
(9) Submitter's Own Program Identification and Suffix(Optional)... CONVERT
(10) Primary Subject Code..... 12.2 12.0
(11) Operating or Monitor System Required Any SNOBOL4 System
(12) New or Revision Code (if revision, show prior Program Number in Item 1).. New
(13) Year Completed..... 1971
(14) Date of Submittal..... March 13, 1973
(15) Documentation (number of original pages submitted)..... 4
(16) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

SHARE PROGRAM LIBRARY SUBMITTAL FORM

DISCLAIMER

Subject Guide:

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements

Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

ABSTRACT

CONVERT will change most CDC FORTRAN* statements to compatible IBM FORTRAN IV, G level. It was designed to eliminate as much of the hand-work as possible, but not to become a full-fledged compiler. The program lists all statements changed, both the original and the modifications made and produces file of converted source images ready for compilation. Compiling of CONVERT will produce comments listing options available while suppressing the source listing. CONVERT requires no special system features and should run on any SNOBOL4 (at least Ver 3.0) system.

* 3000 series, Chippewa and 6000 series through FORTRAN 68.

(Please attach additional pages if necessary).....Total pages attached _____

Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program."

(17) Signature of Submitter and Date _____

3/13/73

(18) Signature of Installation Addressee _____

CONVERT

Source Conversion of CDC FORTRAN To IBM "G Level"

INTRODUCTION

CONVERT was written in the summer of 1971 to meet the needs of the Ohio University Physics and Chemistry Departments in converting large scale CDC FORTRAN programs to run under an IBM G-level compiler. The goal was a program which could produce a file from a CDC source deck which would require little or no additional work to become operational on an IBM compiler. Currently the author does not have access to a SNOBOL4 compiler to make several needed modifications, but it is hoped that the program's modularity will ease the task of further changes.

CAPABILITIES AND LIMITATIONS:

1. All statements changed by the program are printed out labeled as either the original or the changed form. Statements changed have an asterisk in column 73.
2. A summary table of all variable names created and symbolic data set names found is printed out at the end of the program listing. Variables are listed by subprogram names.
3. Restrictions on input and output mode are dependent on the implementation of SNOBOL4 in use. Appropriate changes can be made easily, most sections of the program are clearly marked by comments.
4. All real variables may be made DOUBLE PRECISION and COMPLEX made into COMPLEX*16.
5. Variable names with over six characters are shortened.
6. Input and punched output are in card-image (80-byte record) form.
7. Most CDC functions not available in IBM FORTRAN will be labeled "UNCONVERTIBLE" and not otherwise changed.

TECHNICAL DESCRIPTION:

Source Language:	SNOBOL4, Version 3.6
Applicable Systems:	Any system with SNOBOL4
Records:	Estimate 3 records of output for each card of input.
Data sets used:	Nominally standard system devices
Source images:	332 cards

REFERENCES:

CDC FORTRAN Reference Manual for 3400/3600 #60132900
CDC FORTRAN Extended Reference Manual 6400/6500/6600 #60176600
CDC Chippewa FORTRAN Reference Manual #60132700
FORTRAN IV Reference Manual IBM SRL #GC28-6515

Multiple statements separated by dollar signs

```
ORIGINAL STATEMENT:10      I=2*J $ K=3*L
STATEMENT CHANGED:10      I=2*J
STATEMENT CHANGED:        K=3*L
```

Note that the statement number always refers to the first statement.

Hollerith assignment statements

```
ORIGINAL STATEMENT:11      A=4HTEST
STATEMENT CHANGED:        DATA A$/4HTEST/
STATEMENT CHANGED:11      A=A$
```

A dummy variable name is created, denoted by a trailing \$ in a DATA statement. Multiple definitions and oversize literals may result.

Multiple equals signs in a statement

```
ORIGINAL STATEMENT:100     A=B=C=D
STATEMENT CHANGED:100     C=D
STATEMENT CHANGED:        B=C
STATEMENT CHANGED:        A=B
```

Note that the statement number is assigned to the first pair. For branching, this is correct, for DO loops, this will have to be altered manually. Also note that these statements are executed right to left.

TYPE statement

```
ORIGINAL STATEMENT:        TYPE COMPLEX FREQ
STATEMENT CHANGED:        COMPLEX FREQ
```

```
ORIGINAL STATEMENT:        TYPE DOUBLE VARI
STATEMENT CHANGED:        REAL*8 VARI
```

```
ORIGINAL STATEMENT:        TYPE INTEGER GTYPE
STATEMENT CHANGED:        INTEGER GTYPE
```

```
ORIGINAL STATEMENT:        TYPE DOUBLE PRECISION KLOD
STATEMENT CHANGED:        DOUBLE PRECISION KLOD
The word TYPE is dropped.
```

DOUBLE statement

```
ORIGINAL STATEMENT:        DOUBLE T,R
STATEMENT CHANGED:        REAL*8 T,R
```

FORMAT Statement differences

```
ORIGINAL STATEMENT:1000    FORMAT(* THE ANSWER IS*,R7)
STATEMENT CHANGED:1000    FORMAT(' THE ANSWER IS',A7)
```

Asterisks and "R" type alphanumeric constants changed; in some cases this change in constant type may not be appropriate, as in masking.

ORIGINAL STATEMENT: READ TAPE 8,A,B
STATEMENT CHANGED: READ(IDVC8)A,B

ORIGINAL STATEMENT: WRITE TAPE 8,A,B
STATEMENT CHANGED: WRITE(IDVC8)A,B

These are changed to standard unformatted forms. Device numbers are changed to variables starting with IDVC if not variables already.

READ INPUT TAPE & WRITE OUTPUT TAPE

ORIGINAL STATEMENT: READ INPUT TAPE 8,555,A,B
STATEMENT CHANGED: READ(IDVC8,555)A,B

ORIGINAL STATEMENT: WRITE OUTPUT TAPE 8,555,A,B
STATEMENT CHANGED: WRITE(IDVC8,555)A,B

These are changed to standard formatted forms. Device numbers are changed to variables starting with IDVC if not already variables.

READ & WRITE All dataset reference numbers are changed to symbolic
ENDFILE variables. If they refer to the system reader, printer,
REWIND or punch, the optional forms READ, PRINT, and PUNCH
BACKSPACE may also be used.

ORIGINAL STATEMENT: READ(5,100)A,B
STATEMENT CHANGED: READ 100,A,B

ORIGINAL STATEMENT: WRITE(6,101)A,B
STATEMENT CHANGED: PRINT 101,A,B

ORIGINAL STATEMENT: WRITE(7,102)A,B
STATEMENT CHANGED: PUNCH 102,A,B

ORIGINAL STATEMENT: READ(10,111)A,B
STATEMENT CHANGED: READ(IDVC10,111)A,B

ORIGINAL STATEMENT: WRITE(9,110)A,B
STATEMENT CHANGED: WRITE(IDVC9,110)A,B

ORIGINAL STATEMENT: ENDFILE 4
STATEMENT CHANGED: ENDFILE IDVC4

ORIGINAL STATEMENT: BACKSPACE 4
STATEMENT CHANGED: BACKSPACE IDVC4

ORIGINAL STATEMENT: REWIND 4
STATEMENT CHANGED: REWIND IDVC4

```
ORIGINAL STATEMENT:      IF (A-B) 101,100
STATEMENT CHANGED:      IF (A-B) 101,100,101
```

Logical

```
ORIGINAL STATEMENT:      IF(A.GT.B)101,100
STATEMENT CHANGED:      IF(A.GT.B)GO TO 101
STATEMENT CHANGED:      GO TO 100
```

Note that in the case of the logical IF, the GO TO statement may be superfluous.

DATA statement

ORIGINAL STATEMENT: DATA (A=B) , (B=7)
STATEMENT CHANGED: DATA A/3/ , B/7/

DATA statements initializing labeled COMMON are legal in CDC FORTRAN but IBM FORTRAN syntax requires the use of a BLOCK DATA subprogram.

Numbered COMMON

```
ORIGINAL STATEMENT:      COMMON/10/DUMMY,X,Y
STATEMENT CHANGED:      COMMON/$10/DUMMY,X,Y
```

PROGRAM

```
ORIGINAL STATEMENT:      PROGRAM CONTRL
STATEMENT CHANGED:C      PROGRAM CONTRL
```

Indirect RETURNS from SUBROUTINES
ENTRY points in SUBROUTINES

```
ORIGINAL STATEMENT:      CALL X(A,B),RETURNS(100,200)
STATEMENT CHANGED:      CALL X(A,B,&100,&200)
```

ORIGINAL STATEMENT:	SUBROUTINE X(A,B),RETURNS(100,200)
STATEMENT CHANGED:	SUBROUTINE X(A,B,*,*)

ORIGINAL STATEMENT: ENTRY H
STATEMENT CHANGED: ENTRY H(A,B,*,*)

SUMMARY OF PROGRAM CHANGES: Lists all variable names created or those symbolic data set names encountered.

[illegible]

```
PROGRAM CONTRL      A$  IRDR  IDVC10  IOUT  IDVC9  IPCH  IDVC8  IDVC4
SUBROUTINE X(A,B,*,*)      $10
```



SHARE PROGRAM LIBRARY AGENCY

USER REPORT

PROGRAM CATALOG NUMBER 360D122010

COMPUTER MODEL 370/158

OPERATING SYSTEM VM OS/VS1 CMS

1. Were you able to implement this program using the instructions supplied? no see section 4
2. Did the program run as documented? yes, after one modification
3. Was the documentation adequate? basic and sufficient, more complete description of input deck description would help
4. Did you find any errors in the program? possibly, changed statement 61 from
GT(EØFSW,1) :S(EØØ) → =T(EØFSW,C) :S(EØØ) to
stop problem of program reading file FT01F002
5. Do you have any suggestions to improve the distributed material? no
6. Comments: could not see why program read FT01F001 then
FT01F002 so made fix (Section 4) and placed end
blank card at end of CBC source deck

NAME M. Milner DATE July 27/77 INSTALLATION CODE _____
(if any)

ADDRESS MR. MARTIN MILNER
ACADEMIC SYSTEMS
UNIV. OF VICTORIA
P.O. Box 1700, VICTORIA, B.C.
V8W 2Y2

PLEASE RETURN TO:

SHARE Program Library Agency
Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, NC 27709

