

CFT REFERENCE CARD

SQ-0021 B

Copyright © 1979, 1980, 1981, Cray Research, Inc.

CFT CONTROL STATEMENT FORMAT

CFT (I=idn, L=ldn, B=bdn, C=cdn, E=on, ON=string, OFF=string, TRUNC=n, AIDS=ids, OPT=option)

Parameters may be in any order or may be omitted to cause the default options to apply.

- idn* Source input dataset name; default is \$IN.
- ldn* Listable output dataset name; default is \$OUT. L=0 suppresses all but fatal error messages, which are written on \$OUT.
- bdn* Binary output dataset name; default is \$BLD. No eof is written. If B=0, no binary load files are written.
- cdn* CAL dataset name; default is no dataset.
- n* Highest message level to be suppressed; default is 3. Fatal errors not suppressed.

Level	Severity
1	COMMENT
2	NOTE
3	CAUTION
4	WARNING
5	ERROR

string String of up to 15 letters that selects a compiler option when specified ON or OFF, respectively. Defaults described in the Compiler Option Chart.

nn Number of bits truncated for floating-point results. Truncated bits zeroed. Range is 0<=nn<=47. Default is 0.

ids Number of vectorization inhibition messages.

<i>aias</i>	Number of messages
LOOPNONE	None
LOOPPART	3 per compiler block; 100 messages per compilation (default)
LOOPALL	All

option Optimization level

<i>option</i>	Assumption
NOZEROINC	Constant increment integers cannot be incremented by zero variables (default)
ZEROINC	Constant increment integers can be incremented by zero variables.

FORTRAN LANGUAGE PARAMETER DEFINITIONS

- a* CALL statement: An actual argument; IMPLICIT statement: Beginning letter of all names in a type specification ((a-b) specifies a range); POINTER statement: Pointee
- a(d)* An array declarator
- b* Last letter of a type range
- bloc* Symbolic name of the beginning location
- cb* Symbolic name of a common block
- clist* A list of constants or symbolic names of constants
- d* A dummy argument
- dent* The symbolic name of a destination variable, array element, or array
- e* An expression
- eloc* Symbolic name of the ending location
- en* Function or subroutine name used as an entry
- f* A format identifier or namelist group name
- flist* A format specification
- fwn* Symbolic name of a function subprogram
- group* Group name
- h* A character string recognized as a comment
- i* An integer variable name
- id* A string of 8 digits
- ie* Integer expression
- ilist* A list of specifiers
- iolist* An input/output list
- m* Mode specifier
- n* Number of characters to be processed
- nlist* DATA statement: A list of names and/or an implied-DO list; COMMON and EQUIVALENCE statements: A list of names
- *n* Data length (Integer *2 is 24-bit integer)
- p* PARAMETER statement: A symbolic name; POINTER statement: A pointer
- pgm* Symbolic name of the main program
- proc* The name of an external procedure
- s* A statement label
- sent* Symbolic name of a source variable, array element, or array
- st* Any executable statement that is not a DO, END, or logical IF
- sub* Symbolic name of a subroutine or subroutine entry
- type* Desired data type
- u* Unit identifier
- (u)* Unit identifier specified as a Hollerith constant, left-justified and zero-filled
- v* Name of a constant, variable, function, dummy procedure name, or array name
- w* Local variable or array name

COMPILER DIRECTIVES

Compiler directive lines begin with characters CDIRS in columns 1 through 5 and any of the directives listed below in columns 7 through 72.

DIRECTIVE	FUNCTION
EJECT	Ejects to top of next page
LIST	Resumes listable output
NOLIST	Suppresses production of listable output
CODE	Produces listing of assembler code in selected blocks
NOCODE	Suppresses listing of CFT-generated assembler code
VECTOR	Enables vectorization of inner DO-LOOPS
NOVECTOR	Suppresses vectorization of inner DO-loops
IVDEP	Ignores vector dependencies in the next DO-loop
INT24 <i>list</i>	Identifies listed variables and arrays as 24-bit integers
FLOW	Enables flowtrace
NOFLOW	Disables flowtrace
SCHED	Enables the scheduler
NOSCH	Disables the scheduler
DYNAMIC <i>list</i>	Declares dynamic common blocks
BOUNDS [({ <i>list</i> })]	Checks arrays for out-of-bounds subscripts
VFUNCTION <i>list</i>	Declares that a vector version of an external function exists

VALUE RANGES REPRESENTED IN THE CRAY-1

BINARY RANGE	APPROXIMATE DECIMAL RANGE
$-2^{63} \leq I < 2^{63}$	64-BIT INTEGER $0 \leq I < 10^{19}$
$-2^{23} \leq I < 2^{23}$	24-BIT INTEGER $-8388608 \leq I < 8388608$
$2^{-8192} \cdot (1/2) \leq R < 2^{8192} \cdot (1/2)$	REAL $10^{-2466} \leq R < 10^{2466}$
$2^{-8192} \cdot (1/2) \leq D < 2^{8192} \cdot (1/2)$	DOUBLE-PRECISION $10^{-2466} \leq D < 10^{2466}$
$2^{-8192} \cdot (1/2) \leq C_r , C_i < 2^{8192} \cdot (1/2)$	COMPLEX $10^{-2466} \leq C_r , C_i < 10^{2466}$

COMPILER OPTIONS

OPTION	DESCRIPTION	DEFAULT
A	Aborts job after compilation if any of the program units contains a fatal error	OFF
B	Lists beginning sequence number of each code generation block (G implies B)	OFF
C	Lists common block names and lengths listed on <i>ldn</i> after each program unit	ON
D	Lists DO-loop table	OFF
E	Enables recognition of compiler directive lines	ON
F	Enables FLOWTRACE option. (Also see FLOW/NOFLOW directives)	OFF
G	Lists generated CAL code for each program unit. (See CODE/NOCODE directives)	OFF
H	Causes listing of only the first statement of each program unit. All other list options are ignored or disabled.	OFF
I	Enters compiler-generated statement labels in symbol table	OFF
J	Causes all DO loops to be executed at least once	OFF
L	Enables recognition of output listing control directives	ON
M	Enables scheduler	ON
N	Enters null symbols in symbol table (defined but not referenced)	OFF
O	Identifies execution-time array references with out-of-bounds subscripts	OFF
P	Double precision; 64-bit values	ON
Q	Aborts compilation when 100 fatal error messages counted	ON
R	Rounds results on multiply operation	ON
S	Lists FORTRAN source code	ON
T	Lists symbol table after each program unit	ON
V	Vectorizes inner DO-loops	ON
W	Compiles all floating-point operations as return jumps to user-supplied external routines	OFF
X	Lists symbol table with cross references after each program unit (X overrides T)	OFF
Y	Aborts job when compiler error is detected	OFF
Z	Writes DEBUG Symbol Table on \$BLD	OFF

FORTRAN CALLABLE SUBPROGRAMS

b=Boolean; i=integer; r=real; d=double precision;
c=complex; l=logical; w=i or r; x=b, i, r, or l;
y=b, i, d, c, or l; z=b, i, r, d, c, or l, h=Mollerith

SUBPROGRAM CALLING SEQ.	CODE GENERATED	VECTORIZATION
TRIGONOMETRIC		
r=SIN(r)	External	Full
d=DSIN(d)	External	Full
c=CSIN(c)	External	Full
r=COS(r)	External	Full
d=DCOS(d)	External	Full
c=CCOS(c)	External	Full
r=TAN(r)	External	Pseudo
r=COT(r)	External	Pseudo
r=ASIN(r)	External	Pseudo
r=ACOS(r)	External	Pseudo
r=ATAN(r)	External	Pseudo
d=DATAN(d)	External	Pseudo
r=ATAN2(r)	External	Pseudo
d=DATAN2(d)	External	Pseudo
r= SINH(r)	External	Pseudo
r= COSH(r)	External	Pseudo
r= TANH(r)	External	Pseudo
LOGARITHMIC		
r=ALOG(r)	External	Full
d=DLOG(d)	External	Pseudo
c=CLOG(c)	External	Pseudo
r=ALOG10(r)	External	Full
d=DLOG10(d)	External	Pseudo
TYPE CONVERSION		
r=FLOAT(i)	Inline	Full
i=IFIX(r)	Inline	Full
d=DBLE(r)	Inline	Full
r=SNGL(d)	Inline	Full
c=CMPLX(r)	Inline	Full
i=INT24(i)	Inline	Full
i=LINT(i)	Inline	Full
EXPONENTIAL		
r=SQRT(r)	External	Full
d=DSQRT(d)	External	Pseudo
c=CSQRT(c)	External	Pseudo
r=EXP(r)	External	Full
d=DEXP(d)	External	Pseudo
c=CEXP(c)	External	Pseudo

SUBPROGRAM CALLING SEQ.	CODE GENERATED	VECTORIZATION
ARITHMETIC		
i=IABS(i)	Inline	Full
r=ABS(r)	Inline	Full
d=DABS(d)	Inline	Full
r=CABS(c)	External	Full
i=IDIM(i)	Inline	Full
r=DIM(r)	Inline	Full
i=INT(r)	Inline	Full
r=AINT(r)	Inline	Full
i=IDINT(d)	Inline	Full
i=MOD(i)	External	None
r=AMOD(r)	Inline	Full
d=DMOD(d)	External	Pseudo
i=ISIGN(i)	Inline	Full
r=SIGN(r)	Inline	Full
d=DSIGN(d)	Inline	Full
r=RANF()	External	Full
r=RANGET()	External	None
RANSET	External	None
c=CONJG(c)	Inline	Full
r=REAL(c)	Inline	Full
r=AIMAG(c)	Inline	Full
MAXIMUM/MINIMUM		
i=MAX0(i)	Inline	Full
r=AMAX0(i)	Inline	Full
i=MAX1(r)	Inline	Full
r=AMAX1(r)	Inline	Full
d=DMAX1(d)	Inline	Full
i=MIN0(i)	Inline	Full
r=AMIN0(i)	Inline	Full
i=MIN1(r)	Inline	Full
r=AMIN1(r)	Inline	Full
d=DMIN1(d)	Inline	Full
INPUT/OUTPUT		
EODW(i)	External	None
i=IEOF(i)	External	None
r=EOF(i)	External	None
r=UNIT(i)	External	None
i=LENGTH(i)	External	None
b=GETPOS(i)	External	None
SETPOS(i,i)	External	None
DEBUG AIDS		
ENDRPV	External	None
SETRPV(h,h,h)	External	None
SYMDEBUG(h)	External	None

SUBPROGRAM CALLING SEQ.	CODE GENERATED	VECTORIZATION
BOOLEAN		
b=AND(x,x)	Inline	Full
b=OR(x,x)	Inline	Full
b=XOR(x,x)	Inline	Full
b=EQV(x,x)	Inline	Full
b=COMPL(x)	Inline	Full
b=MASK(i)	Inline	None
b=SHIFT(x,i)	Inline	None
b=SHIFTL(x,i)	Inline	None
b=SHIFTR(x,i)	Inline	None
i=LEADZ(x)	Inline	None
i=POPCNT(x)	Inline	None
b=CSMG(x,x,x)	Inline	None
TIME/DATE		
r=RTC()	Inline	None
i=IRTC()	Inline	None
r=TIMEF()	External	None
r=SECOND(r)	External	None
h=CLOCK(w)	External	None
h=DATE(w)	External	None
h=JDATE(w)	External	None
SYSTEM		
EXIT	External	None
i=LOC(z)	External	None
SSWITCH(i,i)	External	None
ABORT(z)	External	None
ERREXIT	External	None
TRBK(i,y)	External	None
REMARK(z)	External	None
REMARK2(z)	External	None
REMARKF(i,x)	External	None
SENSEFI(i)	External	None
CLEARFI	External	None
SETFI	External	None
CLEARFIS	External	None
SETFIS	External	None
i=NUMARG()	Inline	None
VECTORIZATION		
b=CVMGP(x,x,x)	Inline	Full
b=CVMGM(x,x,x)	Inline	Full
b=CVMGZ(x,x,x)	Inline	Full
b=CVMGN(x,x,x)	Inline	Full
b=CVMGT(x,x,1)	Inline	Full

DO-LOOP FEATURES THAT INHIBIT VECTORIZATION

- Parentheses in subscripts
- A dependent or ambiguous subscript
- A call to a subroutine
- A call to a non-library or non-vector function
- An IF or GO TO statement in a loop
- Any I/O statement

SYMBOLIC DEBUG PACKAGE

The symbolic debug package provides a symbolic memory dump to aid in determining the cause of a job abort. It can be invoked by the DEBUG control statement or the library routine SYMDEBUG. The ON=Z option in the CFT control statement must be specified.

REPRIEVE PROCESSING

Repriev processing suspends normal system error processing and allows the user to attempt to recover from what normally would be an abort condition. The CRAY-OS Version 1 Reference Manual lists reparable abort conditions and selection codes.

MANUALS FOR FORTRAN USERS

- CRAY-1 FORTRAN (CFT) Reference Manual - SR-0009
Describes the CRAY-1 FORTRAN Language, the CRAY-1 FORTRAN Compiler, and related CRAY-1 Operating System characteristics.
- CRAY-1 Library Reference Manual - SR-0014
Summarizes the subprograms available to users of the CRAY-1 Computer System.