

```

C          IR50FN = ICHCK value
INTEGER FN(4)          ! radix 50 file name
BYTE    QUEST(1)       ! prompt
C. Description:        Prompts for file name in 3 parts, converts to ASCII

C/ Name:                IFNQ
C Purpose:              Prompt for file name or any text
C Interface:            Fortran function
C Interface:
C FUNCTION IFNQ(PMPT,STG,LP,LS)
C          IFNQ = ICHCK value
BYTE    PMPT(LP)       ! prompt
BYTE    STG(LS)        ! ASCII file name
INTEGER LP             ! length of prompt
INTEGER LS             ! length of file name string
C. Description:        Type prompt, old text.  Input new text

C/ Name:                IFSTG
C Purpose:              Convert string to integer
C Interface:            Fortran function
C FUNCTION IFSTG(VAR,STG,RADIX)
C          IFSTG = ICHCK value, =0 bad string
INTEGER VAR            ! result
BYTE    STG(1)         ! source string
INTEGER RADIX          ! conversion radix
C. Description:        Converts string, checking for bad radix
C.                      characters.

C/ Name:                IGLOSS
C Purpose:              Implement HELP & GLOSSARY facility
C Interface:
C FUNCTION IGLOSS(STG1,STG2,BUF)
C          returns ICHCK value, = 4 or 5
BYTE    STG1(1)        ! 1st help string
BYTE    STG2(1)        ! 2nd help string
BYTE    BUF(512)       ! sector buffer workspace
C. Description:        Prompts for name and types file of that name.

C/ Name:                IITSTG
C Purpose:              Convert double precision integer to string
C Interface:            Fortran function
C FUNCTION IITSTG(VAR,STG,RADIX)
C          IITSTG=length of resulting string
INTEGER*4    VAR        ! value
BYTE         STG(1)     ! string
INTEGER*4    RADIX      ! conversion radix
C. Description:        Converts integer to string, shifts result over
C.                      for minimum length.

C/ Name:                IQ
C Purpose:              Prompt for integer value
C Interface:            Fortran function
C FUNCTION IQ(VAR,UB,LB,QUEST,RADIX)
C          IQ= ICHCK value
INTEGER VAR        ! result
INTEGER UB        ! upper bound
INTEGER LB        ! lower bound
BYTE    QUEST(1)  ! prompt string

```