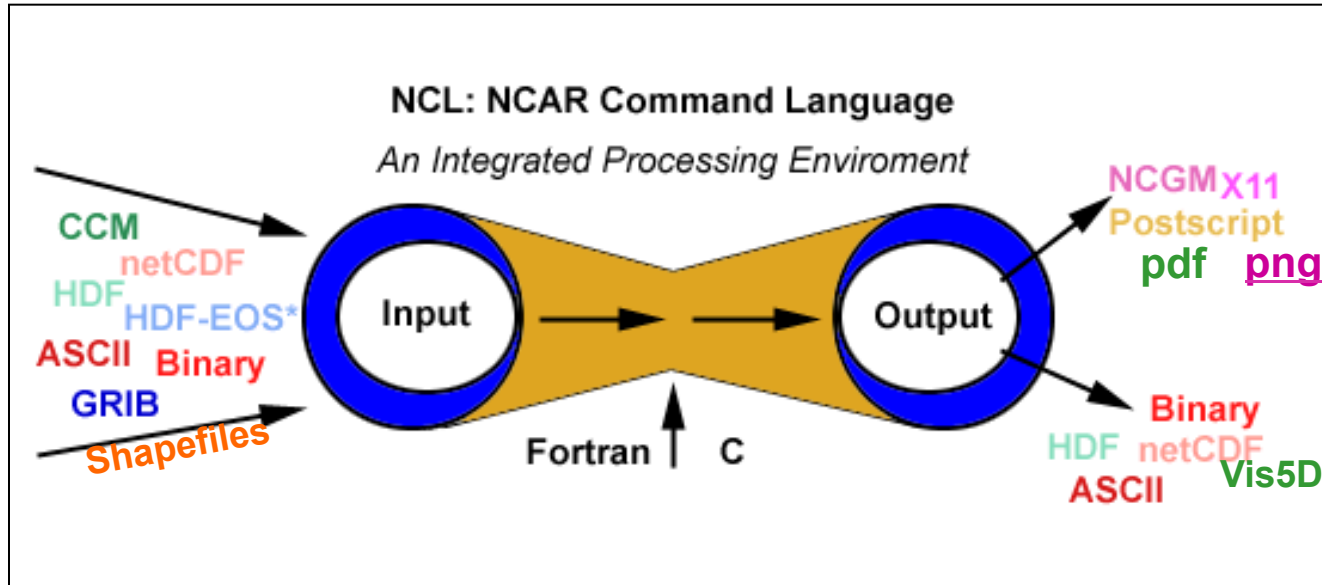


# File Input/Output

## *ASCII and Binary*



**Dennis Shea**

National Center for Atmospheric Research



NCAR is sponsored by the National Science Foundation

# Reading Binary/ASCII data

- **7 functions for reading binary:**

- **fbincread**: reads multiple unformatted sequential records [Fortran; ieee]
- **fbinnumrec**: returns the number of unformatted sequential records [Fortran; ieee]
- **fbindirread**: reads specified record from a Fortran direct access file [ieee]
- **fbinread**: same as **fbincread** but reads only one ieee rec
- **craybincread**: like **fbincread** but for COS blocked data
- **craybinnumrec**: like **fbinnumrec** but for COS blocked data
- **cbinread**: read binary created via C block IO function "write"

- **1 function for reading ASCII data:**

- **asciiread** **[contributed.ncl: readAsciiTable]**
- use NCL str\_\* functions; Fortran/C to read complicated ASCII files

- **all above functions allow data to be shaped**

- x = **fbincread** ("foo\_ieee", rnum, (/10,20,30/), "float")
- a = **asciiread** ("foo\_ascii", (/64,128/), "float")

# setfileoption

[www.ncl.ucar.edu/Document/Functions/Built\\_in/setfileoption.shtml](http://www.ncl.ucar.edu/Document/Functions/Built_in/setfileoption.shtml)

- **allows user to specify file-format-specific options**
  - netCDF, GRIB and Binary options *[currently]*
- **sample usage of selected options**
  - reading/writing Binary
    - **setfileoption**("bin", "ReadByteOrder", "LittleEndian")
    - **setfileoption**("bin", "WriteByteOrder", "BigEndian")

# Writing Binary/ASCII data

- **4 procedures for writing (ieee) binary data**

- **fbinrecwrite**: write unformatted fortran sequential recs
- **fbindirwrite**: write specified record; fortran direct access
- **fbinwrite**: write a binary file containing a single record
- **cbinwrite**: write binary file ; mimics C block IO "write"

- **setfileoption**: can be used to alter default behavior

- **3 procedures to write ascii data to a file**

- **asciwrite**: write a file containing ASCII characters
  - writes a single flat ASCII file. One value per line.
  - No user control of format
- **write\_matrix**: write a multi-dim array to std out or to a file
  - user has format control ... pretty-print
- **write\_table**: write multiple variables with format control

# netCDF, GRIB, HDF ==> binary

```
fin      = addfile ("in.grb", "r")      ; .nc .hdf hdfEOS
u        = fin->U
v        = fin->V
speed    = sqrt(u^2 + v^2)
fout     = "out.bin"
system ("/bin/rm -f "+fout)
```

```
-----
;
; output binary: -1 means append to previous record
; default is to write same endian as current environment
;
;-----
```

```
setfileoption("bin", "WriteByteOrder", "BigEndian")
```

```
fbinrecwrite (fout, -1, fin->time)      ; fortran sequential
fbinrecwrite (fout, -1, fin->lev)       ; different lengths
fbinrecwrite (fout, -1, fin->lat)
fbinrecwrite (fout, -1, fin->lon)
fbinrecwrite (fout, -1, u)             ; (fout, -1, fin->U)
fbinrecwrite (fout, -1, v)             ; (fout, -1, fin->V)
fbinrecwrite (fout, -1, speed)
```

# Reading Simple ASCII (TEXT) Files

1881	-999.9	0.2	-999.9	-999.9	1.5	-999.9	-999.9	-0.2
1882	-1.7	-0.5	0.6	0.1	0.9	-1.9	-3.5	-4.6
1995	-1.0	-0.8	0.4	-1.8	-1.2	-0.4	0.6	-0.1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

```
; read in data
ncols = 9
nrows = 3
ksoi = asciiread ("ascii.in", (/nrows,ncols/), "float")

; partition total array into individual vector arrays
yrs = ksoi(:, 0)
mon1 = ksoi(:, 1)
data = ksoi(:, 1:) ; all but leftmost column

; if you were going to plot/compute, must assign meta data
data@_FillValue = -999.9 ; manually assign
```

# Read ASCII Table with Header

Jan-to-Aug Southern Oscillation Index 1881-1995								
Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
1881	-999.9	-999.9	-999.9	-999.9	-999.9	-999.9	999.9	-999.9
1882	-1.7	-0.5	0.6	0.1	0.9	-1.9	-3.5	-4.6
1995	-1.0	-0.8	0.4	-1.8	-1.2	-0.4	0.6	-0.1

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
```

```
ncols = 9
```

```
nhead = 2 ; number of lines to skip
```

```
ksoi = readAsciiTable ("ascii.in", ncols, "float", nhead)
```

```
yrs = ksoi(:, 0)
```

```
col1 = ksoi(:, 1)
```

```
data = ksoi(:, 1:) ; all but leftmost column
```

```
data@_FillValue = -999.9
```

Last argument could be string:

```
ksoi = readAsciiTable ("ascii.in", ncols, "float", "Year")
```

# write\_matrix(x[\*][\*], fmt, opt)

- **pretty-print 2D array to standard out**
  - integer, float, double
  - user format control (fmt)
  - if not 2D use T=**onedtond**( **ndtooned**(TT) , (/N,M/))
  - T(7,5): **write\_matrix** (T, "5f7.2", False)

```
4.35      4.39      0.27      -3.35      -6.90
4.36      4.66      3.77      -1.66      4.06
9.73      -5.84      0.89      8.46      10.39
4.91      4.59      -3.09      7.55      4.56
  17      3.68      5.08      0.14      -5.63
-0.63     -4.12     -2.51      1.76     -1.43
-4.29      0.07      5.85      0.87      8.65
```

- **create an ASCII file**

```
opt      = True
opt@fout = "foo.ascii"      ; file name
write_matrix (T, "5f7.2", opt)
```