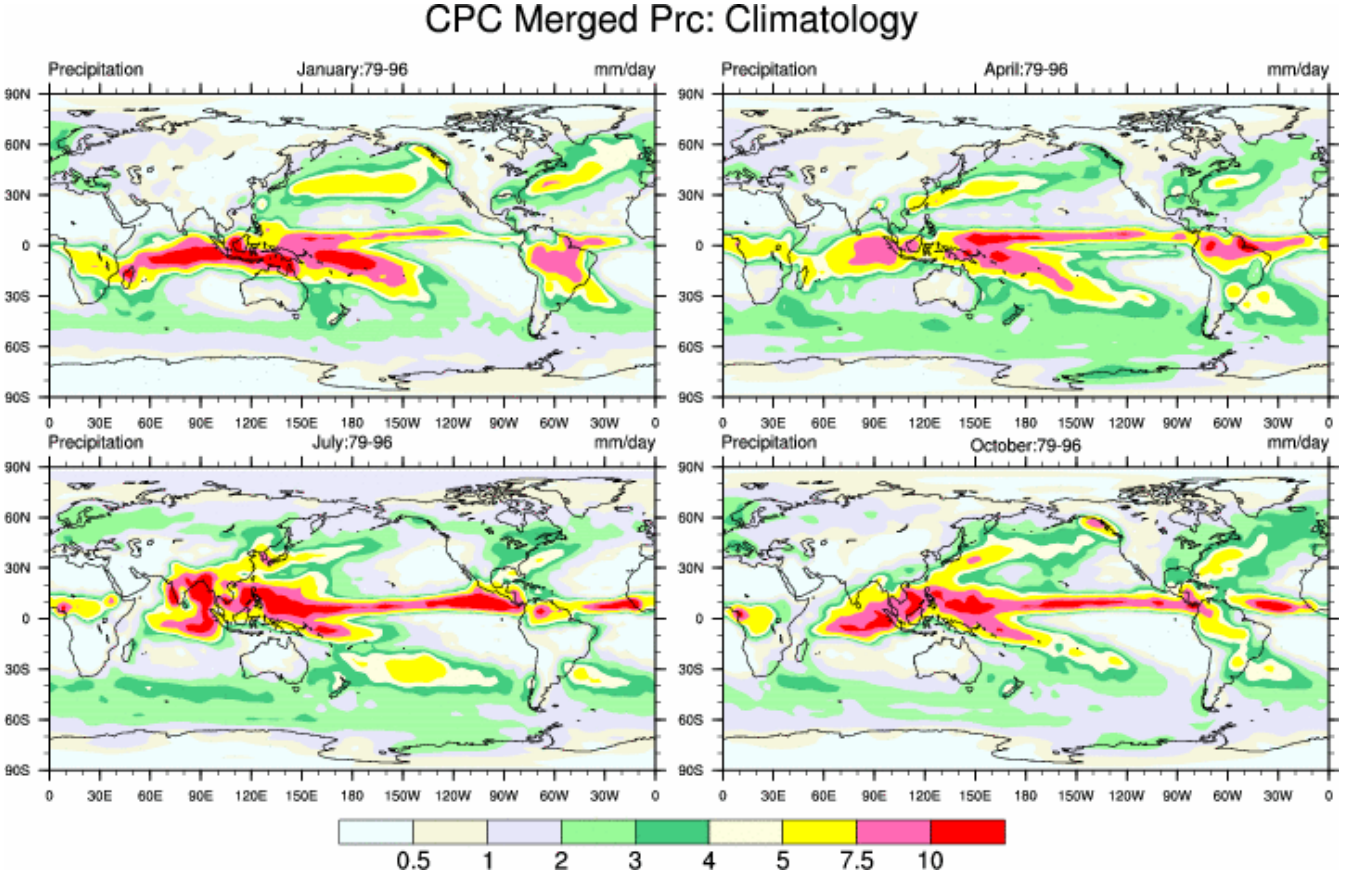
NCL Data Processing



Dennis Shea

National Center for Atmospheric Research

NCAR is sponsored by the National Science Foundation

Data Processing: Meta Data

- **Know Your Data:** most important rule in data processing

- **Meta data:** information **about** the data
  - facilitates processing and data sharing
  - associated with a file and/or a variable

- **Why meta data is useful within NCL**
  - facilitates writing netCDF/HDF file; **automatically written**
  - **gsn_csm*** graphics are **meta data aware**
    - query input variable(s) about attributes and coordinates
  - make variables '**self-describing**'; facilitates debugging
    - **printVarSummary**(…)
  - facilitates building **robust** functions/procedures
    - eg: check units
  - data extraction (coordinate variables): **{**latS**:**latN**}**, **&**lon

NCL (netCDF) Variable Model

X

Scalar
or
Array

attributes

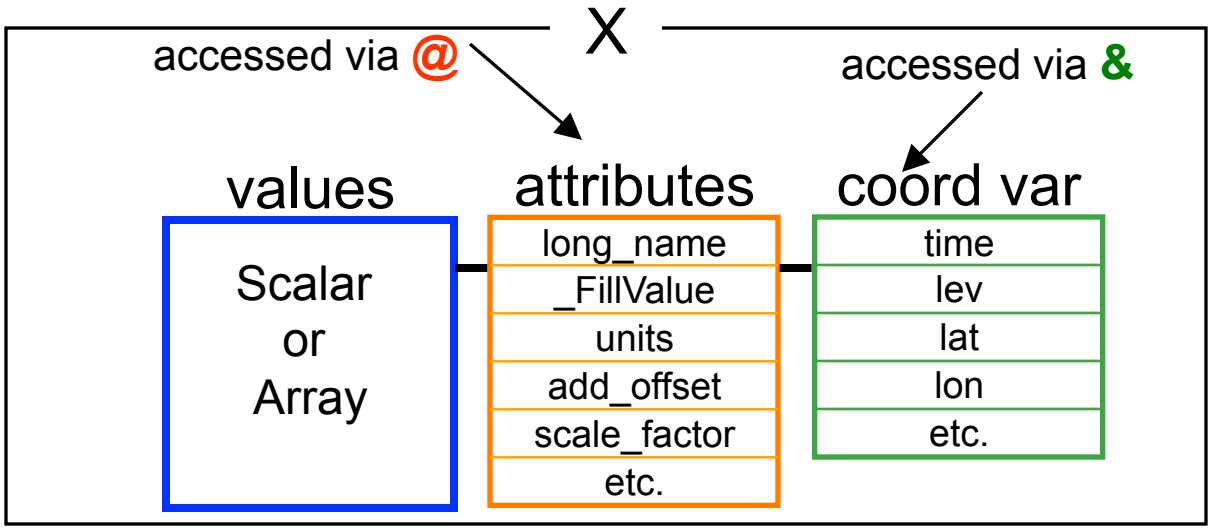
long_name
_FillValue
units
add_offset
scale_factor
etc.

coordinates

time
lev
lat
lon
etc.

```
f = addfile("foo.nc","r")    ; grb/hdf
print(f)                     ; f@title
x = f->X
printVarSummary(x)
```

NCL reads scalar/array variable, attributes, and coordinate variables as one object (structure)



Data Processing: File Meta Data

- **File** meta data: information about file's contents
  - contained within the **global** (file) attributes
    - **C**onventions = "CF-1.4" ; netCDF
    - project_id = "CMIP5"
    - case = "b.e11.B1850C5CN.ne30.004"
    - creation_date = "Mon Jan 5 13:11:07 MST 2015"

    - DYN_OPT = 2 ; WRF netCDF
    - MP_PHYSICS = 2
    - MAP_PROJ = 1

    - HDFEOSVersion : HDFEOS_V2.9 ; HDF, HDF-EOS
    - StructMetadata_0 : GROUP=SwathStructure

  - can be elaborate: span many lines
    - story = "Data were derived via …. "
    - references = "Haley, M. (2015): E=mc^2 is Wrong!"

Data Processing: Variable Meta Data

- **Variable** meta data: information **about** a variable
  - associated with a variable via **attributes** and **coordinates**
    - **not required** but is *highly* recommended
    - self describing

Variable: T
Type: float
Total Size: 65536 bytes
16384 values
Number of Dimensions: 3
Dimensions and Sizes: [time|2] x [lat | 64] x [lon | 128]
Coordinates: **; anything listed here is a CV**
time: [1 .. 7] **; => coordinate variable**
lat: [-87.8638 .. 87.8638] **; =>** can use **{…}** , **&**
lon: [0 .. 357.185]
Number of **Attributes**: 3
_FillValue : 1e36
units : degK
long_name : Temperature

Computations and Meta Data

- **computations can cause loss of meta data**
  - y = x ; variable to variable transfer; **all meta copied**
  - T = T+273.15 ; T retains all meta data; T@units = ”C"
    - T@units = "K" ; user responsibility to update meta
  - z = 5*x ; z will have no meta data

- **built-in functions cause loss of meta data**
  - Tavg = **dim_avg_n**(T, 0)
  - s = **sqrt**(u^2 + v^2)

- **vinth2p is the exception**
  - retains coordinate variables
  - http://www.cgd.ucar.edu/csm/support/Data_P/vert_interp.shtml
    - hybrid to pressure (sigma to pressure) + other examples

Ways to Retain Meta Data (1 of 3)

- **use _Wrap functions** (eg:)
  - dim_avg_n_**Wrap**
  - dim_variance_n_**Wrap**
  - dim_stddev_n_Wrap
  - dim_sum_n_Wrap
  - dim_rmsd_n_Wrap
  - smth9_Wrap
  - g2gsh_Wrap
  - g2fsh_Wrap
  - f2gsh_Wrap
  - f2fsh_Wrap
  - natgrid_Wrap

- f2fosh_Wrap
- g2gshv_Wrap
- g2fshv_Wrap
- f2gshv_Wrap
- f2fshv_Wrap
- f2foshv_Wrap
- linint1_Wrap
- linint2_Wrap
- linint2_points_Wrap
- eof_cov_Wrap
- eof_cov_ts_Wrap
- zonal_mpsi_Wrap
- etc

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

```
                                        ; load not needed 6.2.0
  f = addfile("dummy.nc", "r")
  x = f->X                              ; (time,lev,lat,lon),  (0,1,2,3)
  xZon = dim_avg_n_Wrap(x, 3)           ; xZon will have meta data
                                        ; xZon(time,lev,lat)
```

Ways to Retain Meta Data(2 of 3)

- use copy functions in **contributed.ncl**
  - **copy_VarMeta**(a,b) ; coords **and** atts
  - **copy_VarCoords**(a,b) ; coordinates **only**
  - **copy_VarAtts**(a,b) ; attributes **only**

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
                                        ;load not needed 6.2.0
f = addfile("dummy.nc", "r")

  x = f->X                              ; (time,lat,lon),  (0,1,2)
 ; --------------- calculations--------------------------
   xZon = dim_avg _n(x, 2)              ; xZon(ntim,nlat)
; ---------------copy meta data-----------------------
  copy_VarMeta(x, xZon)                 ; xZon(time,lat)
```

Ways to Retain Meta Data (3 of 3)

- use **variable to variable transfer** + **dimension reduction** to prefine array before calculation
  - requires that user know a priori the array structure

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
                                        ; load not needed from 6.2.0 onward
  f = addfile("dummy.nc", "r")
  x = f->X                               ; x(time,lev,lat,lon), (0,1,2,3)
; -------- var-to-var transfer + dim reduction--------
  xZon  = x(:,:,:,0)                     ; xZon(time,lev,lat)
  xTim  = x(0,:,:,:)                     ; xTim(lev,lat,lon
; --------------------calculations-----------------------
   xZon = dim_avg_n (x, 3)
   xZon@op = "Zonal Avg: "+x@long_name    ; add extra info

   xTim = dim_avg_n (x, 0)
   xTim@op = ”Time Avg: "+x@long_name
```

Meta Data Facilitates Writing Robust Functions

```
undef ("density")
function density( T:numeric, P:numeric, opt:logical)
local t, p, R
begin
  t = T                ; variable to variable transfer of meta data
  p = P
  if (isatt(T,"units") .and. (T@units.eq."C" .or. T@units.eq."degC") ) then
      t = t+273.15
      t@units = "K"
  end if
 if (isatt(P,"units") .and. (P@units.eq."hPa" .or. P@units.eq."mb") ) then
      p = P*100
      p@units = "Pa"
 end if
 R = 287.058          ; J/(kg·K)
 density = R*t/p
 copy_VarCoords(t,density)         ; make return variable self describing
 density@long_name = "density"     ; with meta data
 density@units = "kg/m^3"
 return(density)
end
```

Meta Data: Coordinate Extraction

Coordinate meta data may be used to extract information associated with a variable: **&**, **{…}**

```
T      = f->TMP       ; T(time,lev,lat,lon)

Tnew = linint2_Wrap(T&lon, T&lat, T, True,  LON, LAT, 0 )

        ; Tnew(time,lev,LAT,LON)
```

```
plo    = 850
phi    = 400

Tp     = linint2_Wrap(T&lon, T&lat, T(:, {phi:plo} ,:,:  ) \
                                ,True,  LON, LAT, 0 )

       ;  Tp(time,lev,LAT,LON),       850 <= lev <=400
```

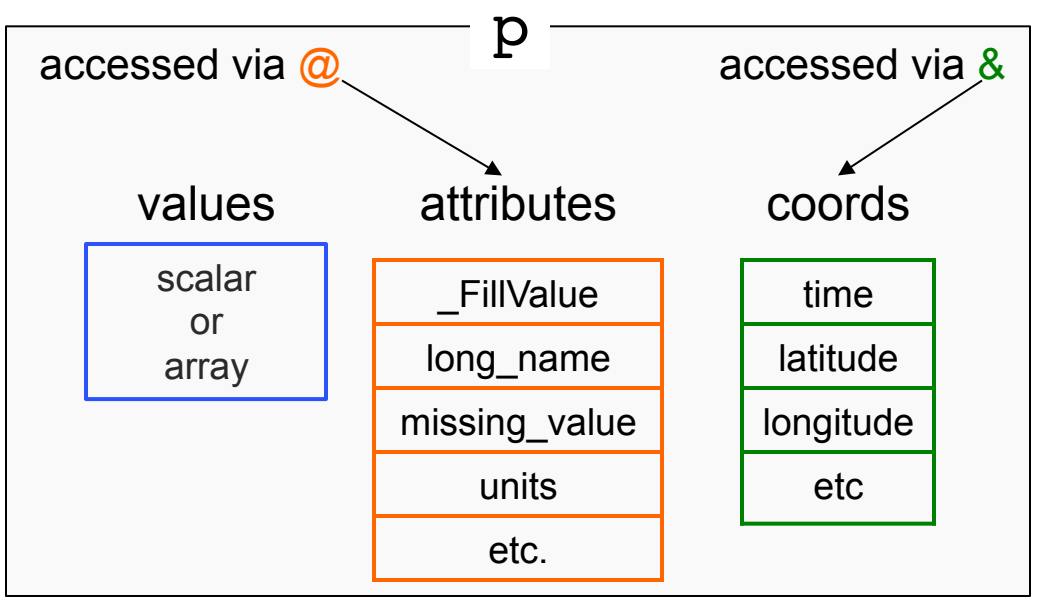
NetCDF [NCL] Variable model

```
p = f->SLP
```

NCL reads

- *data values*
- *attributes*
- *coordinate arrays*

as a **single** data object.



"**printVarSummary**(p)" output

```
Variable: p
Type: float
Total Size: 29272320 bytes
            7318080 values
Number of Dimensions: 3
Dimensions and sizes:     [time | 252] x [latitude | 121] x [longitude | 240]
Coordinates:
            time: [780168..963504]          <=  coordinate variable
            latitude: [90..-90]             <=      "          "      &latitude
            longitude: [ 0..358.5]          <=      "          "      &longitude
Number Of Attributes: 4
  _FillValue :    1e+20
  units :         hPa
  long_name :     Mean sea level pressure
  missing_value :1e+20
```

Meta Data Examined by gsn_csm

```
f = addfile("erai_1989-2009.mon.msl_psl.nc","r")    ; open file [hdf,grib]
p = f->SLP                                          ; (time,lat,lon)
                                                    ; ( 252,121,240)
printVarSummary(p)                                  ; examine variable
wks  = gsn_open_wks("ps","sample")                  ; open a PS file (sample.ps)
plot = gsn_csm_contour_map(wks,p(0,:,:),False)      ; B&W plot of 1st time step
```