
Subject: netCDF/CF conventions

Posted by [jutta.holst](#) on Thu, 01 Feb 2007 14:47:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am trying to convert ascii data into netCDF files using the climate&forecast conventions.

Unfortunately, it does not really work as the CF-checker allways gives the warning:

'no standard_name or long_name attributes'.

Does anyone have an idea what I am doing wrong (code is given below)?

Thank you!

Jutta

```
PRO WRITE_NETCDF_TUT3
```

```
;+
```

```
;Name:
```

```
;  WRITE_NetCDF
```

```
;
```

```
;Purpose:
```

```
;  This program creates simple NetCDF files containing
```

```
;  meteorological data
```

```
;
```

```
;Input:
```

```
;  Meteorological data
```

```
;  1.  year
```

```
;  2 DOY
```

```
;  3.  time
```

```
;  4.  air temperature above canopy 1.3h (°C)
```

```
;  5.  air temperature within canopy 1.3 m a.g.l. (°C)
```

```
;  6.  relative humidity above canopy 1.3h (%)
```

```
;  7.  relative humidity within canopy 1.3 m a.g.l. (%)
```

```
;  8.  wind velocity 42 m a.g.l. (°)
```

```
;  9.  wind direction 42 m a.g.l. (°)
```

```
; 10.  precipitation above canopy 30 m a.g.l. (mm)
```

```
; 11.  net radiation 16 m a.g.l. (W/m2)
```

```
; 12.  incoming short wave radiation 1.3h (W/m2)
```

```
; 13.  outgoing short wave radiation 1.3h (W/m2)
```

```
; 14.  incoming long wave radiation 1.3h (W/m2)
```

```
; 15.  outgoing long wave radiation 1.3h (W/m2)
```

```
; 16.  soil temperature 0.01 m depth (°C)
```

```
; 17.  soil temperature 0.03 m depth (°C)
```

```
; 18.  soil temperature 0.05 m depth (°C)
```

```
; 19.  soil temperature 0.10 m depth (°C)
```

```
; 20.  soil temperature 0.20 m depth (°C)
```

```

; 21. soil temperature 0.40 m depth (°C)
; 22. volumetric soil moisture content (%)
;
;Output:
; NetCDF file containing Tuttlingen meteorological data
;
;Keywords:
; none
;
;Author and history:
; Dirk Schindler Oct 2006

;outfile = 'C:\COPS-GOP\Daten\TUT_test_data.ncd'
MM='Nov06'
;infile = DIALOG_PICKFILE(Title='Select a file path')
;infile='f:\Disc_C\Daten\Datenbanken\GOP-Daten\testfile_TUT. txt'
infile='c:\COPS-GOP\Daten\' + MM + '.txt'

OPENR, lun, infile, /GET_LUN
nlines = FILE_LINES(infile)-1
title = STRARR(1)
READF,lun,title
data = FLTARR(22L,48)
for file=0,(nlines/48)-1 do begin
READF,lun,data
ERR=99999
description = 'Meteorological data Tuttlingen'
convention = 'CF-1.0'

;datum=strcompress(string(data(1),format='(i3)'),/remove_all )
+'_'+strcompress(string(data(0),format='(i4)'),/remove_all)
dat1=FLTARR(2)
dat1=AML_DAT_doy2dat(data(1),data(0))
jahr=strcompress(string(data(0),format='(i4)'),/remove_all)
monat=strcompress(string(dat1(0),format='(i2)'),/remove_all)
if(strlen(monat) EQ 1)then monat='0'+monat
tag=strcompress(string(dat1(1),format='(i2)'),/remove_all)
if(strlen(tag) EQ 1)then tag='0'+tag
datum=jahr+monat+tag

outfile='c:\COPS-GOP\Daten\' + datum + '_gop8_sta_tutt_met.nc'
oid = NCDF_CREATE(outfile, /Clobber) ;Create output file

print,outfile

;----data start-----

```

```

year_name = 'year'
year      = data(0,*)
sec_till_startofyear=(year-1970)*365.*86400.+(round((year-1970)/
4.))*86400.

```

```

doy_name = 'doy'
doy      = data(1,*)

```

```

time_name = 'time'
time_units = 's'
;time      = sec_till_startofyear+hh*3600+min*60
time       = sec_till_startofyear+(fix(data(2,*)/100))*3600+
(data(2,*)-100*(fix(data(2,*)/100)))*60

```

```

Ta1_name = 'air_temperature' ;variable name
Ta1_units = 'K' ;variable units
Ta1_height = '1.3hc' ;relative measurement height
ix=where(data(3,*) NE ERR,cnt_Ta1)
Ta1=FLTARR(nlines)
Ta1(*)=!VALUES.F_NAN
if(cnt_Ta1 ne 0)then begin
Ta1(ix) = data(3,ix)+273.15
endif

```

```

Ta2_name = 'air_temperature' ;variable name
Ta2_units = 'K' ;variable units
Ta2_height = '0.1hc' ;relative measurement height
ix=where(data(4,*) NE ERR,cnt_Ta2)
Ta2=FLTARR(nlines)
Ta2(*)=!VALUES.F_NAN
if(cnt_Ta2 ne 0)then begin
Ta2(ix) = data(4,ix)+273.15
endif

```

```

RH1_name = 'relative_humidity'
RH1_units = '1'
RH1_height = '1.3hc'
RH1 = data(5,*)
ix=where(RH1 EQ ERR,cnt_RH1)
if(cnt_RH1 ne 0)then begin
RH1(ix) = !VALUES.F_NAN
endif

```

```

RH2_name = 'relative_humidity'
RH2_units = '1'
RH2_height = '0.1hc'
RH2 = data(6,*)
ix=where(RH2 EQ ERR,cnt_RH2)

```

```
if(cnt_RH2 ne 0)then begin
  RH2(ix)    = !VALUES.F_NAN
endif
```

```
wv_name      = 'wind_speed'
wv_units     = 'm s-1'
wv_height    = '1.3hc'
wv           = data(7,*)
ix=where(wv EQ ERR,cnt_wv)
if(cnt_wv ne 0)then begin
  wv(ix)      = !VALUES.F_NAN
endif
```

```
wd_name      = 'wind_from_direction'
wd_units     = 'degree'
wd_height    = '1.3hc'
wd           = data(8,*)
ix=where(wd EQ ERR,cnt_wd)
if(cnt_wd ne 0)then begin
  wd(ix)      = !VALUES.F_NAN
endif
```

```
prep_name    = 'precipitation_amount'
prep_units   = 'kg m-2'
prep_height  = '1.3hc'
prep        = data(9,*)
ix=where(prepare EQ ERR,cnt_prep)
if(cnt_prep ne 0)then begin
  prep(ix)    = !VALUES.F_NAN
endif
```

```
Rn_name      = 'surface_net_downward_radiative_flux'
Rn_units     = 'W m-2'
Rn_height    = '1.3hc'
Rn           = data(10,*)
ix=where(Rn EQ ERR,cnt_Rn)
if(cnt_Rn ne 0)then begin
  Rn(ix)      = !VALUES.F_NAN
endif
```

```
Kd_name      = 'surface_downwelling_shortwave_flux_in_air'
Kd_units     = 'W m-2'
Kd_height    = '1.3hc'
Kd           = data(11,*)
ix=where(Kd EQ ERR,cnt_Kd)
if(cnt_Kd ne 0)then begin
  Kd(ix)      = !VALUES.F_NAN
endif
```

```

Ku_name   = 'surface_upwelling_shortwave_flux_in_air'
Ku_units  = 'W m-2'
Ku_height = '1.3hc'
Ku        = data(12,*)
ix=where(Ku EQ ERR,cnt_Ku)
if(cnt_Ku ne 0)then begin
    Ku(ix)   = !VALUES.F_NAN
endif

```

```

Ld_name   = 'surface_downwelling_longwave_flux_in_air'
Ld_units  = 'W m-2'
Ld_height = '1.3hc'
Ld        = data(13,*)
ix=where(Ld EQ ERR,cnt_Ld)
if(cnt_Ld ne 0)then begin
    Ld(ix)   = !VALUES.F_NAN
endif

```

```

Lu_name   = 'surface_upwelling_longwave_flux_in_air'
Lu_units  = 'W m-2'
Lu_height = '1.3hc'
Lu        = data(14,*)
ix=where(Lu EQ ERR,cnt_Lu)
if(cnt_Lu ne 0)then begin
    Lu(ix)   = !VALUES.F_NAN
endif

```

```

Ts001_name = 'soil_temperature'
Ts001_units = 'K'
Ts001_height = '-0.01m'
ix=where(data(15,*) NE ERR,cnt_TS001)
    TS001=FLTARR(nlines)
    TS001(*)=!VALUES.F_NAN
if(cnt_TS001 ne 0)then begin
    TS001(ix)   = data(15,ix)+273.15
endif

```

```

Ts003_name = 'soil_temperature'
Ts003_units = 'K'
Ts003_height = '-0.03m'
ix=where(data(16,*) NE ERR,cnt_TS003)
    TS003=FLTARR(nlines)
    TS003(*)=!VALUES.F_NAN
if(cnt_TS003 ne 0)then begin
    TS003(ix)   = data(16,ix)+273.15
endif

```

```

Ts005_name = 'soil_temperature'
Ts005_units = 'K'
Ts005_height = '-0.05m'
ix=where(data(17,*) NE ERR,cnt_TS005)
  TS005=FLTARR(nlines)
  TS005(*)=!VALUES.F_NAN
if(cnt_TS005 ne 0)then begin
  TS005(ix) = data(17,ix)+273.15
endif

```

```

Ts010_name = 'soil_temperature'
Ts010_units = 'K'
Ts010_height = '-0.10m'
ix=where(data(18,*) NE ERR,cnt_TS010)
  TS010=FLTARR(nlines)
  TS010(*)=!VALUES.F_NAN
if(cnt_TS010 ne 0)then begin
  TS010(ix) = data(18,ix)+273.15
endif

```

```

Ts020_name = 'soil_temperature'
Ts020_units = 'K'
Ts020_height = '-0.20m'
ix=where(data(19,*) NE ERR,cnt_TS020)
  TS020=FLTARR(nlines)
  TS020(*)=!VALUES.F_NAN
if(cnt_TS020 ne 0)then begin
  TS020(ix) = data(19,ix)+273.15
endif

```

```

Ts040_name = 'soil_temperature'
Ts040_units = 'K'
Ts040_height = '-0.40m'
ix=where(data(20,*) NE ERR,cnt_TS040)
  TS040=FLTARR(nlines)
  TS040(*)=!VALUES.F_NAN
if(cnt_TS040 ne 0)then begin
  TS040(ix) = data(20,ix)+273.15
endif

```

```

SMC_name = 'volumetric_soil_moisture_content'
SMC_units = '%'
SMC_height = '-0.30m'
SMC = data(21,*)
ix=where(SMC EQ ERR,cnt_SMC)
if(cnt_SMC ne 0)then begin
  SMC(ix) = !VALUES.F_NAN
endif

```

```

;--data end----

;----header start-----
NCDF_ATTPUT, oid, 'Description', description, /GLOBAL
NCDF_ATTPUT, oid, 'Conventions', convention, /GLOBAL ;write CF
conventions
NCDF_ATTPUT, oid, "System", 'Forest_station_Tuttlingen', /CHAR, /
global

NCDF_ATTPUT, oid, "Location", 'Tuttlingen, Germany', /CHAR, /global
NCDF_ATTPUT, oid, "Longitude", '8.75°E', /CHAR, /global
NCDF_ATTPUT, oid, "Latitude", '47.98°N', /CHAR, /global
NCDF_ATTPUT, oid, "altitude", '645', /CHAR, /global
NCDF_ATTPUT, oid, "mean_slope_exposition", '67 deg', /CHAR, /global
NCDF_ATTPUT, oid, "mean_slope_inclination", '23 deg', /CHAR, /global
NCDF_ATTPUT, oid, "forest_stand", 'Beech, fagus sylvatica', /CHAR, /
global
NCDF_ATTPUT, oid, "canopy_height", '27.7 m', /CHAR, /global
NCDF_ATTPUT, oid, "stand_age", '70 y', /CHAR, /global
NCDF_ATTPUT, oid, "stand_density", '526 trees/ha', /CHAR, /global
NCDF_ATTPUT, oid, "plant_area_index", '5.3 m2/m2', /CHAR, /global
NCDF_ATTPUT, oid, "mean_breast_height_diameter", '27.6 cm', /CHAR, /
global
NCDF_ATTPUT, oid, "Comments", 'time: sec since 01/01/1970 at the end of
the interval', /CHAR, /global
;----header end-----

```

```

year = LINDGEN(nlines)

```

```

yid = NCDF_DIMDEF(oid, year_name, /UNLIMITED) ;define year
dimension

```

```

;vid = NCDF_VARDEF(oid,year_name, [yid],/LONG) ;define
variable
;vid = NCDF_VARDEF(oid,doy_name, [yid],/LONG)
;vid = NCDF_VARDEF(oid,time_name, [yid],/LONG)
vid = NCDF_VARDEF(oid,Ta1_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,Ta2_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,RH1_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,RH1_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,wv_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,wd_name, [yid],/LONG)
;vid = NCDF_VARDEF(oid,prep_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,Rn_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,Kd_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,Ku_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,Ld_name, [yid],/FLOAT)

```

```

;vid = NCDF_VARDEF(oid,Lu_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,TS001_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,TS003_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,TS005_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,TS010_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,TS020_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,TS040_name, [yid],/FLOAT)
;vid = NCDF_VARDEF(oid,SMC_name, [yid],/FLOAT)

;NCDF_ATTPUT, oid, 'year', 'longname', year_name ;write
variables' long names
;
;NCDF_ATTPUT, oid, 'doy', 'longname', doy_name ;write
variables' long names
;
;NCDF_ATTPUT, oid, 'time', 'longname', time_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'time', 'units', time_units ;write
variables' units
;
;NCDF_ATTPUT, oid, vid, 'longname', Ta1_name ;write
variables' long names
;NCDF_ATTPUT, oid, Ta1_name, 'units', Ta1_units ;write
variables' units
;NCDF_ATTPUT, oid, Ta1_name, 'meas_height', Ta1_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'Ta2', 'longname', Ta2_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'Ta2', 'units', Ta2_units ;write
variables' units
;NCDF_ATTPUT, oid, 'Ta2', 'meas_height', Ta2_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'RH1', 'longname', RH1_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'RH1', 'units', RH1_units ;write
variables' units
;NCDF_ATTPUT, oid, 'RH1', 'meas_height', RH1_height ;write measuring
height
;
;NCDF_ATTPUT, oid, 'RH2', 'longname', RH2_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'RH2', 'units', RH2_units ;write
variables' units
;NCDF_ATTPUT, oid, 'RH2', 'meas_height', RH2_height ;write measuring
height

```

```

;
;NCDF_ATTPUT, oid, 'wv', 'longname', wv_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'wv', 'units', wv_units ;write
variables' units
;NCDF_ATTPUT, oid, 'wv', 'meas_height', wv_height ;write measuring
height
;
;NCDF_ATTPUT, oid, 'wd', 'longname', wd_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'wd', 'units', wd_units ;write
variables' units
;NCDF_ATTPUT, oid, 'wd', 'meas_height', wd_height ;write measuring
height
;
;NCDF_ATTPUT, oid, 'prep', 'longname', prep_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'prep', 'units', prep_units ;write
variables' units
;NCDF_ATTPUT, oid, 'prep', 'meas_height', prep_height ;write measuring
height
;
;NCDF_ATTPUT, oid, 'Rn', 'longname', Rn_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'Rn', 'units', Rn_units ;write
variables' units
;NCDF_ATTPUT, oid, 'Rn', 'meas_height', Rn_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'Kd', 'longname', Kd_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'Kd', 'units', Kd_units ;write
variables' units
;NCDF_ATTPUT, oid, 'Kd', 'meas_height', Kd_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'Ku', 'longname', Ku_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'Ku', 'units', Ku_units ;write
variables' units
;NCDF_ATTPUT, oid, 'Ku', 'meas_height', Ku_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'Ld', 'longname', Ld_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'Ld', 'units', Ld_units ;write
variables' units
;NCDF_ATTPUT, oid, 'Ld', 'meas_height', Ld_height ;write

```

```

measuring height
;
;NCDF_ATTPUT, oid, 'Lu', 'longname', Lu_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'Lu', 'units', Lu_units ;write
variables' units
;NCDF_ATTPUT, oid, 'Lu', 'meas_height', Lu_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'TS001', 'longname', TS001_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'TS001', 'units', TS001_units ;write
variables' units
;NCDF_ATTPUT, oid, 'TS001', 'meas_height', TS001_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'TS003', 'longname', TS003_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'TS003', 'units', TS003_units ;write
variables' units
;NCDF_ATTPUT, oid, 'TS003', 'meas_height', TS003_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'TS005', 'longname', TS005_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'TS005', 'units', TS005_units ;write
variables' units
;NCDF_ATTPUT, oid, 'TS005', 'meas_height', TS005_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'TS010', 'longname', TS010_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'TS010', 'units', TS010_units ;write
variables' units
;NCDF_ATTPUT, oid, 'TS010', 'meas_height', TS010_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'TS020', 'longname', TS020_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'TS020', 'units', TS020_units ;write
variables' units
;NCDF_ATTPUT, oid, 'TS020', 'meas_height', TS020_height ;write
measuring height
;
;NCDF_ATTPUT, oid, 'TS040', 'longname', TS040_name ;write
variables' long names
;NCDF_ATTPUT, oid, 'TS040', 'units', TS040_units ;write
variables' units

```

```

;NCDF_ATTPUT, oid, 'TS040', 'meas_height', TS040_height    ;write
measuring height
;
;NCDF_ATTPUT, oid, 'SMC', 'longname',  SMC_name    ;write
variables' long names
;NCDF_ATTPUT, oid, 'SMC', 'units',    SMC_units    ;write
variables' units
;NCDF_ATTPUT, oid, 'SMC', 'meas_height', SMC_height    ;write
measuring height
;
NCDF_CONTROL, oid, /ENDEF                ;exit define mode

;NCDF_VARPUT, oid, 'year', year            ;write the year
;
;NCDF_VARPUT, oid, 'doy', REFORM(doy)
;
;NCDF_VARPUT, oid, 'time', REFORM(time)
;
;NCDF_VARPUT, oid, Ta1_name, REFORM(Ta1)
;
;NCDF_VARPUT, oid, 'Ta2', REFORM(Ta2)
;
;NCDF_VARPUT, oid, 'RH1', REFORM(RH1)
;
;NCDF_VARPUT, oid, 'RH2', REFORM(RH2)
;
;NCDF_VARPUT, oid, 'wv', REFORM(wv)
;
;NCDF_VARPUT, oid, 'wd', REFORM(wd)
;
;NCDF_VARPUT, oid, 'prep', REFORM(prep)
;
;NCDF_VARPUT, oid, 'Rn', REFORM(Rn)
;
;NCDF_VARPUT, oid, 'Kd', REFORM(Kd)
;
;NCDF_VARPUT, oid, 'Ku', REFORM(Ku)
;
;NCDF_VARPUT, oid, 'Ld', REFORM(Ld)
;
;NCDF_VARPUT, oid, 'Lu', REFORM(Lu)
;
;NCDF_VARPUT, oid, 'TS001', REFORM(TS001)
;
;NCDF_VARPUT, oid, 'TS003', REFORM(TS003)
;
;NCDF_VARPUT, oid, 'TS005', REFORM(TS005)
;

```

```

;NCDF_VARPUT, oid, 'TS010', REFORM(TS010)
;
;NCDF_VARPUT, oid, 'TS020', REFORM(TS020)
;
;NCDF_VARPUT, oid, 'TS040', REFORM(TS040)
;
;NCDF_VARPUT, oid, 'SMC', REFORM(SMC)
;
NCDF_CLOSE, oid
stop
endfor
STOP
END

;+
; NAME:
;   AML_DAT_doy2dat.pro
;
; PURPOSE:
;   rechnet aus dem day of year (doy) das datum in form von monat und
tag aus.
;   nicht array-fähig!
;
; CALLING SEQUENCE:
;   result=AML_DAT_doy2dat(doy, jjjj, format=3)
;
; INPUTS:
;   doy   : tag des jahres (1...366)
;   jjjj  : jahreszahl mit hunderter und tausender vom typ "1999"
"2001"
;   format : folgende formate sind möglich:
;           0 = intarr(2) mit 0=tag und 1=monat (default)
;           1 und höher : string nach AML_DAT_jul2string (ohne zeit)
;
; OUTPUT:
;   array mit [monat,zeit] oder string
;
; REVISION HISTORY:
;   13.12.01 mit AML_DAT_jul2string verknüpft
;-

function AML_DAT_doy2dat, doy, jjjj, format=format

    if keyword_set(format) eq 0 then format=0
    ;                               j f m a m j
j a s o n d
    if AML_DAT_schaltjahr(jjjj) eq 0 then monatdoy =
[0,31,59,90,120,151,181,212,243,273,304,334]

```

```

    if AML_DAT_schaltjahr(jjjj) eq 1 then monatdoy =
[0,31,60,91,121,152,182,213,244,274,305,335] ;schaltjahr
    monate=where(monatdoy lt doy)
    monat=fix(n_elements(monate))
    tag=doy-monatdoy[monat-1]

    if format eq 0 then return, [monat,tag] else begin
        julian=JULDAY(monat,tag,jjjj)
        return, AML_DAT_jul2string(julian, format=format)
    endelse

end

```

```

;+
; NAME:
;   AML_DAT_schaltjahr.pro
;
; PURPOSE:
;   prüft ob eine (vierstellige) jahreszahl ein schlatjahr ist.
;
; CALLING SEQUENCE:
;   result=schaltjahr(jahreszahl)
;
; INPUTS:
;   jahreszahl : vierstellige jahreszahl, z.B. 1999 oder 2000
;
; OUTPUT:
;   1 wenn Schaltjahr
;   0 wenn kein Schaltjahr
;
; REFERENCE
;   DWD, "Allgemeine Meteorologie", S. 158
;
; REVISION HISTORY:
;   22.12.99 AC
;-

```

```

function AML_DAT_schaltjahr, jahreszahl
    on_error, 2
    s=0
    if jahreszahl mod 4 eq 0 then s=1
    if jahreszahl mod 100 eq 0 AND ((jahreszahl mod 1000)/100) mod 4
ne 0 then s=0 ;hunderterjahre
    return, s
end

```

```

;+
; NAME:

```

```

; AML_DAT_jul2string.pro
;
; PURPOSE:
; erstellt aus einem julianischen datumszahl einen string. der
; string kann verschiedene
; formate aufweisen.
;
; CALLING SEQUENCE:
; result=AML_DAT_jul2string(julian,format=format, time=time)
;
; INPUTS:
; julian : zahl. long oder double.
; format : folgende formate sind möglich:
;         1 = string der form "1.7.2000" oder "1.7.2000
9:20" (default)
;         2 = string der form "01.07.2000" oder "01.07.2000 09:20"
;         3 = string der form "1.7." oder "1.7. 9:20"
;         4 = string der form "01.07." oder "01.07. 09:20"
;         5 = string der form "1. Juli 2000" oder "1. Juli 2000
10:10"
;         6 = string der form "July 1 2000" oder "July 1 2000
10:10"
;         7 = string der form "July 2000" oder "July 2000 09:20"
;         8 = string der form '20000701' oder '20000701' (dBase)
;         9 = string der form '2000' oder '2000'
;         liste fortsetzbar
; time   : keyword 0 / nicht gesetzt = nur datum wird ausgegeben
;         1 = zeit und datum werden ausgegeben
;         2 = nur zeit wird ausgegeben
; quattro: input "julian" ist quattro zahl anstatt julday.
;
; OUTPUT:
; string in obigem format
;
; REVISION HISTORY:
; 05.01.00 AC
; 09.10.00 AC format-keyword hinzugefügt und somit ist die funktion
"doy_stringdat" ab sofort überflüssig.
; 01.03.01 AC format=8 hinzugefügt.
; 13.12.01 AC umbenannt zu "AML_DAT_jul2string" allgemein für julday
umgebaut.
; 10.01.02 AC quattro hinzugefügt
;-

function AML_DAT_jul2string, julian, format=format, time=time,
quattro=quattro

    if keyword_set(quattro) then julian=double(julian)

```

```

+2415018.5 ;umwandlung quattro -> julian
  if not keyword_set(time) then time=0
  if n_params() lt 1 then julian=systime(/julian)

monatstr_de=['Januar','Februar','März','April','Mai','Juni'
,'Juli','August','September','Oktober','November','Dezember' ]
monatstr_en=['January','February','March','April','May'
,'June','July','August','September','October','November','December']

caldat, julian, monat, tag, jahr, stunde, minute, sekunde
if keyword_set(format) eq 0 then format=1
del='.'

;datum
case format of
  1 : dstr=strcompress(string(fix(tag), format='(i2)')
+ '.'+string(fix(monat), format='(i2)')+ '.'+string(fix(jahr),
format='(i4.4)'),/remove_all)
  2 : dstr=strcompress(string(fix(tag), format='(i2.2)')
+ '.'+string(fix(monat), format='(i2.2)')+ '.'+string(fix(jahr),
format='(i4.4)'),/remove_all)
  3 : dstr=strcompress(string(fix(tag), format='(i2)')
+ '.'+string(fix(monat), format='(i2)')+ '.'./remove_all)
  4 : dstr=strcompress(string(fix(tag), format='(i2.2)')
+ '.'+string(fix(monat), format='(i2.2)')+ '.'./remove_all)
  5 : dstr=strcompress(string(fix(tag), format='(i2)'), /
remove_all)+ '.'+monatstr_de[monat-1]+ ' '+string(fix(jahr),
format='(i4.4)')
  6 : dstr=monatstr_en[monat-1]+ ' '+strcompress(string(fix(tag),
format='(i2)'), /remove_all)+ ' '+string(fix(jahr), format='(i4.4)')
  7 : dstr=monatstr_en[monat-1]+ ' '+string(fix(jahr),
format='(i4.4)')
  8 : dstr=strcompress(string(fix(jahr), format='(i4.4)')
+string(fix(monat), format='(i2.2)')+string(fix(tag),format='(i2.2)'),/
remove_all)
  9 : dstr=strcompress(string(fix(jahr), format='(i4.4)'),/
remove_all)
  else: begin & message, 'Unknown date format. format set to default
(0)', /informational & return, " & end
endcase

;zeit
case format of
  1 : tstr=strcompress(string(fix(stunde), format='(i2)')
+ '.'+string(fix(minute), format='(i2.2)'),/remove_all)
  3 : tstr=strcompress(string(fix(stunde), format='(i2)')
+ '.'+string(fix(minute), format='(i2.2)'),/remove_all)
  else : tstr=strcompress(string(fix(stunde), format='(i2.2)')

```

```
+':'+string(fix(minute), format='(i2.2)'),/remove_all)
endcase

;output
case time of
1      : return, dstr+' '+tstr
2      : return, tstr
else   : return, dstr
endcase

end
```
