# WCR User Notes

Radar processed data are written in NetCDF files. The catalog of all available files from the experiment and the associated release notes are available in the Catalog_Release_Notes.txt. In addition a NetCDF header text file (a 'cdl' file) augmented with additional comments is also available to help browse and understand all the variables that may be present in every individual file. Due to the versatility of the radar using more than one antenna and capable of acquiring different reflectivity and velocity products there could be differences in what measurements are taken and therefore what variables and products are written in each file.

In this document we provide a brief description of the WCR main processed data products (variables) and how to use them.

## 1. Reflectivity and Doppler velocity variables (not all attributes are listed)

```
float reflectivity(np,profile,range) ; // multiple power products
   reflectivity:long_name = "Equivalent reflectivity factor" ;
   reflectivity:units= "mm^6/m^3" ;
   reflectivity:_FillValue = -32767.f ;
   reflectivity:antenna =" ";  // list(np) of active antenna names
   reflectivity:beamid = 0s;   // list(np) of active antenna beam IDs
   reflectivity:calcoef=0.f;   // list(np) of calibration coef. in dBZ@1km
   reflectivity:npid=0;        // list(np) of reflectivity product IDs
   reflectivity:status="mean noise subtracted, range correction applied,
                        no threshold applied, no atten.correction";

float velocity(nv,profile,range) ; // multiple velocity products
   velocity:long_name = "Doppler radial velocity. Positive is toward the
                         radar";
   velocity:units = "m/s" ;
   velocity:_FillValue = -32767.f ;
   velocity:antenna =" ";      // list(nv) of active antenna names
   velocity:beamid = 0s;       // list(nv) of active beam IDs
   velocity:nvid=0;            // list(nv) of velocity product IDs
   velocity:maxvel = 0.f ;     // list(nv) of Doppler Nyquist velocity
```

Reflectivity and Doppler velocity variables are the 2 main processed radar measurements written in a NetCDF file. They are present in every WCR NetCDf file regardless of the radar acquisition mode used.

**reflectivity** is the radar reflectivity factor (Z) in $mm^6m^{-3}$. No correction for scattering and absorption is applied. The mean noise has been subtracted, but no further detection is performed. In other words about half of the noise pixels are left in the data array with positive values; the negative values are the received power (noise) lower than the mean. This representation of the reflectivity gives the possibilities to apply different schemes to detect cloud (or precipitation) pixels (data points). For more information see the description of **reflectivity_mask** below.

**velocity** is the reflectivity weighted radial Doppler velocity measured by the radar. Positive velocity is toward the radar. It is corrected for the aircraft motion contribution along the beam using the aircraft navigational and GPS data. All target and noise pixels are present and one needs to use a detection scheme or the appropriate reflectivity mask - the **reflectivity_mask** matching the reflectivity with the same id (**npid**) as the id (**nvid**) of the velocity of interest. Velocity maximum unambiguous value (±Nyquist velocity) is given in the attribute **maxvel**. Any target radial Doppler velocity exceeding these limits will fold. **velocity** is not corrected for folding.

*Dimension description:*

The dimension order depends on the computing environment used. When reading the variables in IDL, the dimension order is reversed.

***range***: measurements along the radar beam; determined by the number of the sampled range gates. The distance from the radar to each range gate is recorded in the variable **range**:

```
float range(range) ;
  range:long_name = "Range to (geometric) center of radar range gates" ;
  range:units = "meters" ;
```

The radar resolution in meters along the range is given in the global attribute **WCRrangeresolution**

The radar received power diminishes with range. For meteorological (distributed) targets the loss of power with distance is a function of 1/range$^2$. The variable **range_cor** provides the received power correction for every range gate, where the range is given in kilometers. This correction has been applied to the **reflectivity** variable.

```
float range_cor(range) ;
  range_cor:long_name = "Received power range(in km)correction (1/r^2)" ;
  range_cor:units = "dB" ;
```

***profile***: measurements in time (along flight path); determined by the radar dwell time (ie, integration time, number of pulses averaged for every active beam). The time matching each profile is recorded in the variable **time** in unix seconds:

```
double time(profile);  // time stamps representing the middle of radar
                       // profile dwell(integration) time (WCRdaqint)
  time:long_name = "Profile acquisition time"; // according to WCRirigflag
  time:units = "seconds since 1970-01-01 00:00:00 +0000" ;
  time:strptime_format = "seconds since %F %T %z" ;
```

The radar data sampling in time is not done with a precise constant rate and is not anchored at the top of every second (like, for example, the King Air data). The mean time interval between profiles is given in the global attribute **WCRtimeint.** To translate the time interval into distance, if needed, the user is provided with aircraft true air speed only (**TAS**). If different reference (e.g., Earth relative) is desired the user should use **LAT** and **LON** variables and perform the necessary calculations to get the distance.

The radar resolution in the plane normal to beam depends on the range and antenna beamwidth given in the global attribute **WCR_BeamWidth** and is approximately the same in H- and E-plane.

***np* (*nv*)**: number of measured reflectivity(velocity) products. Some of the files may have different number of reflectivity and velocity products, ie *np* may not equal to *nv* and their order may not match. The reason is that some of the files may contain more than one velocity and or reflectivity product per antenna(beam) due to the use of the antenna dual-polarization capability (currently the only linear dual-polarization antenna for the WCR installation on the UW King Air is the up/side pointing antenna). The indices are defined by the order of the radar beams used and polarization measurements taken during the acquisition. For example, for single-polarization antennas, if 'up', 'down', and 'down-fore' beams, in that order, were used during the acquisition then *np=nv=3* and the 1$^{st}$ index would be 'up' beam, etc. Typically, one power and one velocity product are recorded per active non-polarimetric antenna and therefore for these antennas *np=nv.* To find match between reflectivity and velocity products use their IDs (***nvid*** and ***npid***).

*Attributes:*

The attributes of the variables define which antennas/beams are used, what polarization is transmitted and received, the calibration coefficient applied, and what the data represent. For simplicity, in the NetCDF variable descriptions included in this document, only the more important attributes are shown.

Several of the attributes have been already described in conjunction with the description of the reflectivity and velocity variables.

There are two attributes (**antenna** and **beamid)** that identify the antenna/beam used for every product written in the file. **antenna** contains the names of the active antennas and **beamid** is the active beam identifier. Note that for UW King Air there could be two beams (and therefore 2 values for **beamid**) associated with one and the same antenna – this is the antenna for the 'up' (**beamid**=1) and 'side' (**beamid**=0) beams using an external reflector plate to direct the beam to up-pointing position. The **global attributes** shown below also identify this UW King Air specific installation:

```
:WCR_AntID    =  1s,       1s,      2s,   3s,        4s ; // antenna IDs
:WCR_AntName  = "side/up, side/up, down, side-fore, down-fore";

:WCR_BeamID   =  0s,      1s,     2s,   3s,        4s ;  // beam IDs
:WCR_BeamName = "side,    up,    down, side-fore, down-fore";

:WCRmirror=1.f    // wcrheader: N2UW mirror position,0=side beam,1=up beam
```

## 2. Reflectivity mask

```
short reflectivity_mask(np,profile,range);// target mask
  reflectivity_mask:long_name = "Two byte target mask" ;
  reflectivity_mask:units = "" ;
  reflectivity_mask:beamid = 0s;
  reflectivity_mask:npid=0;
  reflectivity_mask:activebits = 0,1,2,3,8,9,10,11;// currently active bits
  reflectivity_mask:setbitname =
"1_noise_StDev,2_noise_StDev,3_noise_StDev,saturation,surface_cluter,surface,
sub_surface,leak";
  reflectivity_mask:status="none set,LSB bits set,LSB&MSB bits set";
  reflectivity_mask:statusid=0s;  // 0,1,2
```

```
Description of reflectivity_mask:
LSB set bit # (bit decimal value):
bit 0 (1) - signal above 1 StDev of mean noise (LSB value=1)
bit 1 (2) - signal above 2 StDev of mean noise (LSB value=3)
bit 2 (4) - signal above 3 StDev of mean noise (LSB value=7)
bit 3 (8) - signal in the receiver saturation range (LSB value=15)
bit 4-7   - reserved
Note: bits 0,1,2, and 3 are inter-dependent
MSB set bit # (bit decimal value):
bit  8(256) - surface clutter affected gate (mask value>=256)
             (clutter may not be detectable; max range 150 m)
bit  9(512) - estimated surface return gate (mask value=512)
             (surface return may not be detectable)
bit 10(1024)- sub-surface gate (mask value=1024)
             (sub-surface gates may contain detectable levels)
bit 11(2048)- gate affected by surface return cross-talk
             (leak may not be detectable; max range 75 m; mask value>=2048)
bit 12-15   - reserved
```

The Reflectivity mask is a convenient way to remove the noise pixels in the reflectivity and velocity images. It is a 2-byte word, with each bit described above. The mask array matches the reflectivity array and therefore there is separate mask for every reflectivity product. The simplest way to use the mask is to identify all pixels that have bit 2 set. This will provide reflectivity that exceeds the measurements mean noise by 3 sigma and should eliminate the vast majority of the noise pixels. However this may not be a robust detection due to many reasons. In addition, keep in mind that this is not a weather (cloud and or precipitation) mask. Any return signal exceeding the pre-determined power is considered a valid signal and therefore surface, insects, birds, etc., and possibly some artifacts could be present in the reflectivity after applying the reflectivity mask and marking all pixels for which the mask is 0 as no-signal or missing value. In the radar NetCDF files -32767 is used as missing value. Typically there should not be any missing values in the reflectivity and velocity variables, but in the reflectivity there will be values equal or smaller than 0, which are not valid reflectivities (as described above).

## 3. Other variables

There are many other variables in the radar NetCDF files. Most of them are included to allow advanced radar data analysis. Some of the variables are written in a file only if a specific acquisition mode is used. For example, if a Doppler spectrum mode is used (also called FFT mode) then variables starting with prefix **psd_** are written and variables with a prefix **ppm** are not. It is the other way around if a pulse-pair acquisition mode (also called SPP or CPP) is used.

Here we will mention only a few of the variables that have a straightforward use and can be easily utilized if needed:

```
float wcraspect(profile) ;  // range/profile ratio based on ACID air speed
  wcraspect:long_name = "WCRrangesampling/(WCRtimeint*tas)" ;
  wcraspect:units = "" ;
  wcraspect:_FillValue = -32767.f;        //for any tas<50 or tas>200 m/s
  wcraspect:dependencies = "ACID tas" ;  // ACID=N2UW or N130AR
```

**wcraspect** is a convenient variable that provides the aspect ratio between the range gate sampling and the sampling along the flight based on the true air speed. It gives the scaling factor for the image. It varies from profile to profile (due to very slight non-regular sampling in time and the varying true aircraft speed).

```
float wcrbeamvector(beam,profile,vector3) ;  // multi-antenna/beam
  wcrbeamvector:long_name = "(East,North,Up) radar beam unit vectors" ;
  wcrbeamvector:units = "" ;
  wcrbeamvector:_FillValue = -32767.f;
  wcrbeamvector:beamid = 0s ;
  wcrbeamvector:dependencies = "ACID IRS variables" ;// ACID=N2UW or N130AR
```

**wcrbeamvector** is the unit vector of the antenna beam pointing angles in Earth coordinate reference frame. Note that it has a dimension **beam** (not **np** or **nv**), representing all active antennas/beams for this acquisition mode. Use the variable attribute **beamid** to determine the beam vector (index) you should use with a particular **reflectivity** or **velocity**. **wcrbeamvector** can be used to position the reflectivity and velocity in Earth reference space instead of their native aircraft reference frame. In addition to **wcrbeamvector**, the geo-referenced position of the radar for every profile is given by the variables **LAT**, **LON**, and **ALT**:

```
double LON(profile) ;
  LON:long_name = "Radar platform Longitude" ;
  LON:units = "degree_east" ;
  LON:_FillValue = -32767.d;
  LON:source = "ACID" ;
  LON:dependencies = "" ;    // from AC global attribute 'coordinates'

double LAT(profile) ;
  LAT:long_name = "Radar platform Latitude" ;
  LAT:units = "degree_north" ;
  LAT:_FillValue = -32767.d;
  LAT:source = "ACID" ;
  LAT:dependencies = "" ;    // from AC global attribute 'coordinates'

float ALT(profile) ;
  ALT:long_name = "Radar platform Altitude from MSL" ;
  ALT:units = "meters" ;
  ALT:_FillValue = -32767.f;
  ALT:source = "ACID" ;
  ALT:dependencies = "" ;   // from AC global attribute 'coordinates'
```