

MQNC (short for 'MQuest NetCdf') netcdf is based on the MEDS-ascii format. MQNC databases consist of a folder (and subfolders) containing the netcdf files and a master 'keys' netcdf file. The 'keys' file contains metadata about each profile in the database. Below is an example of a keys file header:

```
netcdf IQUODtestdataNOQC_keys {
dimensions:
    N_Casts = UNLIMITED ; // (114665 currently)
    String_1 = 1 ;
    String_2 = 2 ;
    String_4 = 4 ;
    String_10 = 10 ;
variables:
    float obslat(N_Casts) ;
%latitude
    float obslng(N_Casts) ;
%longitude
    float c360long(N_Casts) ;
%longitude in 360degrees
    int autoqc(N_Casts) ;
%QC information for CSIRO automated tests
    autoqc:conventions = "0=OK,1=fail aut1, 2=fail aut2, 3=fail
aut1&2" ;
    autoqc:_FillValue = 9 ;
    char stn_num(N_Casts, String_10) ;
%profile unique ID number
    char callsign(N_Casts, String_10) ;
%ship or platform callsign
    char obs_y(N_Casts, String_4) ;
% year
    char obs_t(N_Casts, String_4) ;
% time
    char obs_m(N_Casts, String_2) ;
% month
    char obs_d(N_Casts, String_2) ;
% day
    char data_t(N_Casts, String_2) ;
% data type (e.g., XB, BO, CT) for a full list, see below
    char d_flag(N_Casts, String_1) ;
% duplicate flag
    d_flag:conventions = "D=yes, N=no" ;
    d_flag:_FillValue = "N" ;
    char data_source(N_Casts, String_10) ;
% Original Data source
    int priority(N_Casts) ;
%CSIRO priority code. The lower the number, the more trustworthy the
data/metadata
```

Data_type codes ('data_t' field in keys file)

- > BA - bathy
- > BO - bottle
- > BT - micro or digital bathythermograph (early xbt)?

```

> CT - CTD
> CU ,Ãì CTD upcast
> DB - drifting buoy
> DT ,Ãì digital thermograph
> MB ,Ãì MBT (mechanical bathythermograph)
> MR - moored buoy (moorings)
> ST ,Ãì surface temperature
> TE - tesac
> UN - Unknown
> UO - Undulating oceanic recorder
> XB - XBT
> XC ,Ãì XCTD

```

THE DATABASE:

Every profile is listed in the keys file and the 'stn_num' variable give information on the location of the profile netcdf file in the database. For example, in the database 'fr02', stn_num (station or profile number) 88938009 is located at fr02/88/93/80/09ed.nc. There is also a fr02/88/93/80/09raw.nc file. 'ed' means edited (includes automatic and manual QC flags added by CSIRO) and 'raw' contains the originator's version of the data (including the originator's flags if present). For the best copy of the dataset, the 'ed' version should be used.

Each profile's netcdf file is structured as given below. Comments on the right give information for the most relevant fields. If more information on the fields is required, refer to the table at:
<http://www.nodc.noaa.gov/GTSPP/document/datafmt/medsfmt.html>

```

netcdf 3ed {
dimensions:
    N_Prof = UNLIMITED ; // (1 currently)
    Nparms = 30 ;
    Nsurfc = 30 ;
    Num_Hists = 100 ;
    time = 1 ;
    latitude = 1 ;
    longitude = 1 ;
    depth = 900 ;
    String_1 = 1 ;
    Single = 1 ;
    String_2 = 2 ;
    String_4 = 4 ;
    String_5 = 5 ;
    String_8 = 8 ;
    String_10 = 10 ;
    String_12 = 12 ;
    String_6 = 6 ;
    String_16 = 16 ;
variables:
    int woce_date(Single) ; %Date
in the format yyyyymmdd UTC
    woce_date:long_name = "date" ;
    woce_date:units = "yyyyymmdd UTC" ;
    woce_date:data_min = 19990307 ;
    woce_date:data_max = 19990307 ;

```

```

        int woce_time(time) ;                                %Time
in hhmss
        woce_time:long_name = "time of day" ;
        woce_time:units = "hhmss" ;
        woce_time:data_min = 115600 ;
        woce_time:data_max = 115600 ;
        double time(time) ;                                %date since
        time:long_name = "time" ;
        time:units = "days since 1900-01-01 00:00:00" ;
        time:data_min = 36224.4972222224 ;
        time:data_max = 36224.4972222224 ;
        float latitude(latitude) ;                          % latitude
        latitude:long_name = "latitude" ;
        latitude:units = "degrees_N" ;
        latitude:valid_min = -90.f ;
        latitude:valid_max = 90.f ;
        latitude:C_format = "%8.4f" ;
        latitude:FORTRAN_format = "F8.4" ;
        latitude:data_min = -17.7667f ;
        latitude:data_max = -17.7667f ;
        float longitude(longitude) ;
%longitude
        longitude:long_name = "longitude" ;
        longitude:units = "360degrees_E" ;
        longitude:valid_min = 0.f ;
        longitude:valid_max = 360.f ;
        longitude:C_format = "%9.4f" ;
        longitude:FORTRAN_format = "F9.4" ;
        longitude:data_min = 108.7667f ;
        longitude:data_max = 108.7667f ;
        int Num_Hists(Single) ;                                %
number of history codes in the Act_Code, Act_Parm, Aux_ID etc fields.
        int No_Prof(Single) ;                                %
number of profiles in this file (e.g. for CTD you might have TEMP and PSAL, so
No_Prof = 2)
        int Nparms(Single) ;                                % number in
Pcode, Parm, Q_Parm fields
        int Nsurfc(Single) ;                                % number of
surface codes (in SRFC_Code, SRFC_Parm, SRFC_Q_Parm variables)
        char Mky(String_8) ;
        char One_Deg_Sq(String_8) ;
        char Cruise_ID(String_10) ;                                %
cruise identifier if available
        char Data_Type(String_2) ;                            % data type
(see list above)
        char Iumsgno(String_12) ;
        char Stream_Source(String_1) ;
        char Uflag(String_1) ;
        char MEDS_Sta(String_8) ;
        char Q_Pos(String_1) ;
        char Q_Date_Time(String_1) ;
        char Q_Record(String_1) ;
        char Up_date(String_8) ;                                % date the
file was last updated
        char Bul_Time(String_12) ;
        char Bul_Header(String_6) ;
        char Source_ID(String_4) ;

```

```

char Stream_Ident(String_4) ;
char QC_Version(String_4) ;
char Data_Avail(String_1) ;
char Prof_Type(N_Prof, String_16) ; %
type of data in each profile (No_Prof). Eg, 'TEMP', 'PSAL', 'COND'
char Dup_Flag(N_Prof, String_1) ; %
Duplicate flag ('D' = duplicate, 'N' = not duplicate)
char Digit_Code(N_Prof, String_1) ;
char Standard(N_Prof, String_1) ;
float Deep_Depth(N_Prof) ; % deepest
depth for each profile (No_Prof)
char Pcode(Nparms, String_4) ;
char Parm(Nparms, String_10) ;
char Q_Parm(Nparms, String_1) ;
char SRFC_Code(Nsurfc, String_4) ; %
Surface code (see
http://www.nodc.noaa.gov/GTSPP/document/codetbls/gtsppcode.html for full
listing)
char SRFC_Parm(Nsurfc, String_10) ; %
Surface parameter data for each SRFC_Code
char SRFC_Q_Parm(Nsurfc, String_1) ; %
Surface quality flag
char Ident_Code(Num_Hists, String_2) ;
% Organisation that created the record
char PRC_Code(Num_Hists, String_4) ; %
Software that altered the record
char Version(Num_Hists, String_4) ; %
Software version number
char PRC_Date(Num_Hists, String_8) ; %
Date of alteration
char Act_Code(Num_Hists, String_2) ; %
Action performed on parameter
char Act_Parm(Num_Hists, String_4) ; %
Parameter that action was performed on
float Aux_ID(Num_Hists) ; % Depth of
changed parameter
int Flag_severity(Num_Hists) ; %
Severity of the flag (see
http://www.nodc.noaa.gov/GTSPP/document/codetbls/gtsppcode.html#QUAL)
char Previous_Val(Num_Hists, String_10) ; %
Value of the parameter before the change
char D_P_Code(N_Prof, String_1) ; % D =
Depth, P = pressure
int No_Depths(N_Prof) ; %
number of depths/pressures in each profile
float Depthpress(N_Prof, depth) ; %
Depth/pressure values
Depthpress:_FillValue = -99.99f ;
float Profparm(N_Prof, time, depth, latitude, longitude) ;
% Parameter values (TEMP, PSAL, COND found here)
Profparm:_FillValue = -99.99f ;
char DepresQ(N_Prof, depth, String_1) ;
% Individual quality flags for each depth/pressure point (see
http://www.nodc.noaa.gov/GTSPP/document/codetbls/gtsppcode.html#QUAL)
char ProfQP(N_Prof, time, depth, latitude, longitude, String_1) ;
% Individual quality flags for each TEMP(/PSAL/COND) point (see
http://www.nodc.noaa.gov/GTSPP/document/codetbls/gtsppcode.html#QUAL)

```

```
// global attributes:  
      :title = "Meds-ASCII UOT data" ;
```
