DESCRIPTION OF THE VALIDATION FLAGS

A detailed description of the validation procedure applied to the original data is given in the "Hydrographic Atlas of the Southern Ocean" (Olbers, Gouretski, Seiss, Schröter, 1992)

The main idea of the validation procedure applied to the data was to put them through a number of filters. The data which fail to pass a filter (check) are marked with the respective value of the validation flag.

There are two validation flags for every hydrographic station:

Station_Validation_Flag is given in the station header and is applied for the whole station. It characterizes the quality of the information within the station header or/and the quality of all temperature or/and salinity or/and oxygen data at the station.

Standard_Data_Validation_Flag is given for every standard level. It characterizes the quality of temparature or/and salinity or/and oxygen data at every standard level.

In the data base the validation flags are represented by an INTEGER*2 variable (32 bits). It allows to keep information about up to 32 checks simultaneously. INTEGER data types are represented internally in binary twos complement notation and a bit in a binary pattern (Validation_Flag) has a value of 0 (data passed the check) or 1 (data faild the check).

A discription of the validation flags and examples of VAX FORTRAN programs for manipulations with validation flags are given below.

Table		VALIDATION FLAGS	
N	VAL Falg	Meaning of the value	Abbreviation used by PANGAEA
0	0	No Validation applied	
1	1	Cruise_Number changed	Cruise number changed
2	2	Date_Time corrected	Date/Time corrected
3	4	Date_Time doubtful	Date/Time doubtful
4	8	Longitude corrected	Longitude corrected
5	16	Latitude corrected	Latitude corrected
6	32	Longitude and Latitude corrected	Longitude and Latitude corrected
7	64	Coordinates doubtful	Coordinates doubtful
8	128	Out of temperature range	Out of T range
9	256	Out of salinity range	Out of S range
10	512	Out of oxygen range	Out of O2 range
11	1024	Statistical check for Temperature	Statchk for T
12	2048	Statistical check for Salinity	Statchk for S
13	4096	Statistical check for Oxygen	Statchk for O2
14	8192	Special case of salinity correction through the comparison with high-quality data	Special case of S correction
15	16384	Unrealistic salinity profile	Unrealistic S profile
16	32768	Unrealistic temperature profile	Unrealistic T profile
17	65536	Misleading dummy value for temperature (t=0)	t=0 removed

18	131072	T,S-diagram check	T,S-diag. chk
19	262144	Stability check	Stab. chk
20	524288	Statistical check for density	Statchk for density
21	1048576	Temperature/Oxygen diagram check	T,O2-diag. chk
22	2097152	Oxygen data rejected for the station	O2 rejected for the station
23	4194304	NOT USED	NOT USED
24	8388608	Oxygen data rejected for the whole cruise	O2 rejected for the whole cruise
25	16777216	Salinity at the station rejected through analysis of maps	S at the station rejected through analysis of maps
26	33554432	Oxygen at the station rejected through analysis of maps	O2 at the station rejected through analysis of maps
27	67108864	NOT USED	NOT USED
28	134217728	T and S observed with uncalibrated device	T and S observed with uncalibrated device
29	268435456	Salinity data rejected for the whole cruise	S rejected for the whole cruise
30	536870912	Salinity data rejected at the station	S rejected at the station
31	1073741824	Temperature data rejected at the station	T rejected at the station

Two examples of the usage of the validation flags are given below.

DESCRIPTION OF THE VARIABLES in the FORTRAN program:

```
C  VFSTATION -Value of the Validation_Flag in the Station header
C  Array VFSTNDATA of the Validation_Flags for each standard level
C  Array of temperature TEM(k) for the station under consideration
C  Array of salinity    SAL(k) for the station under consideration
C  Array of standard level depths Z (maximum 42 levels)
C  Integer array NV. Sequential numbers of the elements of array NV
C    are equal to the sequential numbers of Validation_Flags as given in
C    the  Description of Validation Flags
C
C.......................
      integer*2 NV(32)
      integer*4 vfstation,vfstndata(42)
      real*4 tem(42), sal(42), z(42)
C.......................
```

EXAMPLE 1.

```
C SELECTION OF TEMPERATURE AND SALINITY DATA WHICH PASSED QUALITY CHECKS
C BOTH FOR TEMPERATURE AND SALINITY
C
      call bit(vfstation,NV)
      ns=nv(7)+nv(25)+nv(28)+nv(29)+nv(30)+nv(31)  !! reject station
      if(ns.gt.0) go to 1
C
      kk=0
      do 2 k=1,NLEVEL
      call bit(vfstndata, NV)
      ns=nv(8)+nv(9)+nv(11)+nv(12)+nv(15)+nv(16)+nv(17)+nv(18)+nv(19)
     *  +nv(20)+nv(23)
C
```

```
      if(ns.ne.0)go to 2 !! reject standard level
      kk=kk+1
      tem(kk)=tem(k)
      sal(kk)=sal(k)
      z(kk)=z(k)
2     continue
1     continue
C.....
```

EXAMPLE 2.

```
C SELECTION OF OXYGEN DATA WHICH PASSED QUALITY CHECKS FOR OXYGEN
      call bit(vfstation,NV)
      ns=nv(7)+nv(22)+nv(24)+nv(26)
      if(ns.gt.0) go to 1 !! reject station
C
      kk=0
      do 2 k=1,NLEVEL
      call bit(vfstndata, NV)
      ns=nv(10)+nv(13)+nv(21)
C
      if(ns.ne.0)go to 2 ! reject standard level
      kk=kk+1
      OX(KK)=OX(K)
      z(kk)=z(k)
2     continue
1     continue
C.....
```

The following subroutine BIT uses VAX intrinsic function BTEST
(see "Programming in VAX FORTRAN", Digital Equipment GmbH, 1987)

```
      subroutine BIT(M,IA)
C
C   M - value of the Validation_Flag to check
C-------------------------------------------------
C   IA - integer array, where IA(i)=1 when i-th bit of M equal 1 (the value
C   has not passed i-th check)
C   or IA(i)=0 when i-th bit of M equal 0 (the value has passed i-th check)
C-------------------------------------------------
C   Index "I" for the element of array "IA" is the number of the
C   corresponding  validation procedure.
C  (For meaning of Flags see Description of Validation Flags)
C
C   IA(i)=1 means that after the validation procedure the previous value
C   of the Validation_Flag was replaced by the new value according to:
C   New_Valid_Flag=Old_Valid_Flag + 2**I
C-------------------------------------------------
      integer*2 ia(32)
      integer*4 N,M,IA1
      logical*2 T
      IA1=M
      do 2 ib=1,32
```

```
      ia(ib)=0
      ia2=ib-1
      ia1=M
      T=BTEST(ia1,ia2)
      if(T.eq..TRUE.)ia(ib)=1
    2 continue
      return
      end
```

Victor Gouretski, AWI 08 July 1993