

Interface name	ARGO PROFILE FORMAT	
Reference	CO-IF0022	
Version	2.0	
Comment	-	
Contact	Thierry.Carval@ifremer.fr	
History	20/02/01	version 0.80
	11/09/01	version 1.21
	12/11/01	version 1.2d
	22/11/01	argo data meeting comments approved version by argo data management meeting in Brest

The differences between version 2.0 and the previous proposals are typed in orange letters.

Dimension definitions

```
// Date and time dimension (YYYYMMDDHH24MISS), UTC universal time
DATE_TIME = 14 ;
```

```
// Null terminated string dimensions
```

```
STRING256 = 256 ;
```

```
STRING64 = 64 ;
```

```
STRING32 = 32 ;
```

```
STRING16 = 16 ;
```

```
STRING8 = 8 ;
```

```
STRING4 = 4 ;
```

```
STRING2 = 2 ;
```

```
// The following dimensions depend on the data set
```

```
// Number of profiles
```

```
N_PROF = 10 ;
```

```
// Number of parameters
```

```
N_PARAM = 4 ;
```

```
//Number of vertical levels
```

```
N_LEVELS = 270 ;
```

```
// Number of technical parameters
```

```
N_TECH_PARAM = 25;
```

```
// Number of calibrations
```

```
N_CALIB = 10;
```

```
// Quality control history
```

```
N_HISTORY = UNLIMITED;
```

Global attributes

```
// global attributes  
// Empty
```

General information on profile

// General information on profile

```

char DATA_TYPE(STRING16);
    DATA_TYPE:comment = "Data type" ;
    // Argo profile

char FORMAT_VERSION(STRING4);
    FORMAT_VERSION:comment = "File format version " ;
    // Now 2.0

float HANDBOOK_VERSION;
    HANDBOOK_VERSION:comment = "Data handbook version" ;
    // Now 1.0

char REFERENCE_DATE_TIME(DATE_TIME) ;
    REFERENCE_DATE_TIME:comment = "Date of reference (convention : YYYYMMDDHH24MI) for Julian days";
    REFERENCE_DATE_TIME:conventions = "YYYYMMDDHH24MISS" ;
    // 195001010000 recommended as the first day (day one)

char PROJECT_NAME(N_PROF, STRING64);
    PROJECT_NAME:comment = "Name of the project" ;

char PI_NAME (N_PROF, STRING64);
    PI_NAME:comment = "Primary investigator name" ;

char STATION_PARAMETERS(N_PROF, N_PARAM,STRING4) ;
    STATION_PARAMETERS:long_name = "List of available parameters for the station" ;
    STATION_PARAMETERS:conventions = "GF3 code list in (PRES, TEMP, PSAL, CNDC)";

char PLATFORM_NUMBER (N_PROF, STRING8) ;
    PLATFORM_NUMBER:long_name = "Float unique identifier" ;
    PLATFORM_NUMBER:conventions = "WMO float identifier : QA911111" ;

int CYCLE_NUMBER(N_PROF);
    CYCLE_NUMBER:long_name = "Float cycle number" ;
    CYCLE_NUMBER:conventions = "0..N, 0 : launch cycle (if exists), 1 : first complete cycle" ;
    CYCLE_NUMBER:_FillValue = 99999 ;

char DIRECTION(N_PROF);
    DIRECTION:long_name = "Direction of the station profiles" ;
    DIRECTION:conventions = "A: ascending profiles, D: descending profiles " ;

char DATA_CENTRE(N_PROF, STRING2) ;
    DATA_CENTRE:long_name = "Data centre in charge of float data processing" ;
    DATA_CENTRE:conventions = "GTSP table" ;

char DATE_CREATION(DATE_TIME) ;
    DATE_CREATION:comment = "Date of file creation " ;
    DATE_CREATION:conventions = "YYYYMMDDHH24MISS" ;

char DATE_UPDATE(DATE_TIME) ;
    DATE_UPDATE:long_name = "Date of update of this file" ;
    DATE_UPDATE:conventions = "YYYYMMDDHH24MISS" ;

char DC_REFERENCE(N_PROF, STRING16) ;
    DC_REFERENCE:long_name = "Station unique identifier in data centre" ;
    DC_REFERENCE:conventions = "Data centre convention" ;

char DATA_STATE_INDICATOR(N_PROF, STRING4);
    DATA_STATE_INDICATOR:long_name = "Degree of processing the data have passed through";
    DATA_STATE_INDICATOR:conventions = "OOPC table";

```

```

char DATA_MODE(N_PROF) ;
    DATA_MODE:long_name = "Delayed mode or real time data" ;
    DATA_MODE:conventions = "R : real time ; D : delayed mode" ;

char INST_REFERENCE(N_PROF, STRING64) ;
    INST_REFERENCE:long_name = "Instrument type" ;
    INST_REFERENCE:conventions = "Brand/type/serial number" ;

char WMO_INST_TYPE(N_PROF, STRING4) ;
    WMO_INST_TYPE:long_name = "Coded instrument type" ;
    WMO_INST_TYPE:conventions = "WMO code table 1770 – instrument type" ;

double JULD(N_PROF);
    JULD:long_name = "Julian day (UTC) of the station relative to REFERENCE_DATE_TIME " ;
    JULD:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD:conventions = "Relative julian days with decimal part (as parts of day)" ;
    JULD:_FillValue = 999999. ;

char JULD_QC(N_PROF);
    JULD_QC:long_name = "Quality on Date and Time" ;
    JULD_QC:conventions = "Q where Q =[0-9]" ;
    JULD_QC:_FillValue = "0" ;

double JULD_LOCATION(N_PROF);
    JULD:long_name = "Julian day (UTC) of the location relative to REFERENCE_DATE_TIME " ;
    JULD:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD:conventions = "Relative julian days with decimal part (as parts of day)" ;
    JULD:_FillValue = 999999. ;

double LATITUDE(N_PROF);
    LATITUDE:long_name = "Latitude of the station, best estimated value" ;
    LATITUDE:units = "degree_north" ;
    LATITUDE:_FillValue = 99999. ;
    LATITUDE:valid_min = -90. ;
    LATITUDE:valid_max = 90. ;

double LONGITUDE(N_PROF);
    LONGITUDE:long_name = "Longitude of the station, best estimated value" ;
    LONGITUDE:units = "degree_east" ;
    LONGITUDE:_FillValue = 99999. ;
    LONGITUDE:valid_min = -180. ;
    LONGITUDE:valid_max = 180. ;

char POSITION_QC(N_PROF);
    POSITION_QC:long_name = "Quality on position (latitude and longitude)" ;
    POSITION_QC:conventions = "Q where Q =[0-9]" ;
    POSITION_QC:_FillValue = "0" ;

char POSITIONING_SYSTEM(N_PROF, STRING8) ;
    POSITIONING_SYSTEM:long_name = "Positioning system" ;
    POSITIONING_SYSTEM:conventions = "ARGOS or GPS" ;

// Following QC Variables may be discarded if the matching parameter is not measured
char PROFILE_PRES_QC(N_PROF);
    PROFILE_PRES_QC:long_name = "Global quality flag of pressure profile" ;
    PROFILE_PRES_QC:conventions = "Q where Q =[0-9]" ;
    PROFILE_PRES_QC:_FillValue = "0" ;

char PROFILE_TEMP_QC(N_PROF);
    PROFILE_TEMP_QC:long_name = "Global quality flag of temperature profile" ;
    PROFILE_TEMP_QC:conventions = "Q where Q =[0-9]" ;
    PROFILE_TEMP_QC:_FillValue = "0" ;

char Q_PROFILE_PSAI_QC(N_PROF);

```

```
PROFILE_Psal_QC:long_name = "Global quality flag of practical salinity profile" ;
PROFILE_Psal_QC:conventions = "Q where Q =[0-9]" ;
PROFILE_Psal_QC:_FillValue = "0" ;

char PROFILE_CNDC_QC(N_PROF);
PROFILE_CNDC_QC:long_name = "Global quality flag of conductivity profile" ;
PROFILE_CNDC_QC:conventions = "Q where Q =[0-9]" ;
PROFILE_CNDC_QC:_FillValue = "0" ;
```

Profile data

```

// Data for profiles
// Variable naming conventions:
// Measured value : GF3 code (mandatory if measured and mentioned in PARAMETERS),
// Corrected measure : GF3_CORRECTED,
// Corrected measure default value is the measure itself,
// Associated QC flag : GF3_CORRECTED_QC,

float PRES(N_PROF, N_LEVELS) ;
    PRES:long_name = "Pressure";
    PRES:_FillValue = 99999.f ;
    PRES:units = "decibar" ;
    PRES:valid_min = 0.f ;
    PRES:valid_max = 15000.f ;
    PRES:C_format = "%7.1f";
    PRES:FORTRAN_format="F7.1";
    PRES:resolution=0.1f;

float PRES_CORRECTED(N_PROF, N_LEVELS) ;
    PRES_CORRECTED:long_name = "Corrected pressure" ;
    PRES_CORRECTED:_FillValue = 99999.f ;
    PRES_CORRECTED:units = "decibar" ;
    PRES_CORRECTED:valid_min = 0.f ;
    PRES_CORRECTED:valid_max = 15000.f ;
    PRES_CORRECTED:C_format = "%7.1f";
    PRES_CORRECTED:FORTRAN_format="F7.1";
    PRES_CORRECTED:resolution=0.1f;

char PRES_CORRECTED_QC(N_PROF, N_LEVELS) ;
    PRES_CORRECTED_QC:long_name = "Quality on pressure" ;
    PRES_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;
    PRES_CORRECTED_QC:_FillValue = "0" ;

float TEMP(N_PROF, N_LEVELS) ;
    TEMP:long_name = "Temperature in situ T90 scale" ;
    TEMP:_FillValue = 99999.f ;
    TEMP:units = "degree_Celsius" ;
    TEMP:valid_min = -3.f ;
    TEMP:valid_max = 40.f ;
    TEMP:comment = "In situ measurement";
    TEMP:C_format = "%9.3f";
    TEMP:FORTRAN_format="F9.3";
    TEMP:resolution=0.001f;

float TEMP_CORRECTED(N_PROF, N_LEVELS) ;
    TEMP_CORRECTED:long_name = "Corrected temperature" ;
    TEMP_CORRECTED:_FillValue = 99999.f ;
    TEMP_CORRECTED:units = "degree_Celsius" ;
    TEMP_CORRECTED:valid_min = -3. f ;
    TEMP_CORRECTED:valid_max = 40. f ;
    TEMP_CORRECTED:comment = "Corrected value";
    TEMP_CORRECTED:C_format = "%9.3f";
    TEMP_CORRECTED:FORTRAN_format="F9.3";
    TEMP_CORRECTED:resolution=0.001f;

char TEMP_CORRECTED_QC(N_PROF, N_LEVELS) ;
    TEMP_CORRECTED_QC:long_name = "quality on temperature" ;
    TEMP_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;
    TEMP_CORRECTED_QC:_FillValue = "0" ;

float PSAL(N_PROF, N_LEVELS) ;
    PSAL:long_name = "Practical salinity, sal78" ;

```

```

PSAL:_FillValue = 99999.f ;
PSAL:units = "psu" ;
PSAL:valid_min = 0. f ;
PSAL:valid_max = 60. f ;
PSAL:comment = "In situ measurement, mandatory if salinity measured";
PSAL :C_format = "%9.3f";
PSAL :FORTRAN_format="F9.3";
PSAL:resolution=0.001f;

float PSAL_CORRECTED(N_PROF, N_LEVELS) ;
PSAL_CORRECTED:long_name = "Corrected practical salinity";
PSAL_CORRECTED:_FillValue = 99999.f ;
PSAL_CORRECTED:units = "psu" ;
PSAL_CORRECTED:valid_min = 0. f ;
PSAL_CORRECTED:valid_max = 60. f ;
PSAL_CORRECTED:comment = " Corrected value";
PSAL_CORRECTED :C_format = "%9.3f";
PSAL_CORRECTED :FORTRAN_format="F9.3";
PSAL_CORRECTED:resolution=0.001f;

float PSAL_CORRECTED_ERROR(N_PROF, N_LEVELS) ;
PSAL_CORRECTED_ERROR:long_name = "Error on corrected practical salinity";
PSAL_CORRECTED_ERROR:_FillValue = 99999.f ;
PSAL_CORRECTED_ERROR:units = "psu" ;
PSAL_CORRECTED_ERROR:valid_min = 0. f ;
PSAL_CORRECTED_ERROR:valid_max = 60. f ;
PSAL_CORRECTED_ERROR:comment = "Error on corrected value";
PSAL_CORRECTED_ERROR:C_format = "%9.3f";
PSAL_CORRECTED_ERROR:FORTRAN_format="F9.3";
PSAL_CORRECTED_ERROR:resolution=0.001f;

char PSAL_CORRECTED_QC(N_PROF, N_LEVELS) ;
PSAL_CORRECTED_QC:long_name = "quality on practical salinity" ;
PSAL_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;
PSAL_CORRECTED_QC:_FillValue = "0" ;

float CNDC(N_PROF, N_LEVELS) ;
CNDC:long_name = "Electrical conductivity" ;
CNDC:_FillValue = 99999.f ;
CNDC:units = "mmho/cm " ;
CNDC:valid_min = 0.f ;
CNDC:valid_max = 60.f ;
CNDC:comment = "Not mandatory, available only if measured";
CNDC:C_format = "%9.3f";
CNDC:FORTRAN_format="F9.3";
CNDC:resolution=0.001f;

float CNDC_CORRECTED(N_PROF, N_LEVELS) ;
CNDC_CORRECTED:long_name = "Corrected electrical conductivity" ;
CNDC_CORRECTED:_FillValue = 99999.f ;
CNDC_CORRECTED:units = "mmho/cm " ;
CNDC_CORRECTED:valid_min = 0.f ;
CNDC_CORRECTED:valid_max = 60.f ;
CNDC_CORRECTED:comment = "Not mandatory, available only if measured, corrected value";
CNDC_CORRECTED:C_format = "%9.3f";
CNDC_CORRECTED:FORTRAN_format="F9.3";
CNDC_CORRECTED:resolution=0.001f;

char CNDC_CORRECTED_QC(N_PROF, N_LEVELS) ;
CNDC_CORRECTED_QC:long_name = "Quality on electrical conductivity" ;
CNDC_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;
CNDC_CORRECTED_QC:_FillValue = "0" ;
CNDC_CORRECTED_QC:comment = " Not mandatory, available only if measured";

```


Profile calibration information

// Float calibration information

```

char PARAMETER(N_PROF, N_CALIB, N_PARAM,STRING4) ;
    PARAMETER:long_name = "List of parameters with calibration information" ;
    PARAMETER:Conventions = "GF3 code list in (PRES, TEMP, CNDC, CPV$, PMV$)";

char SCIENTIFIC_CALIB_EQUATION(N_PROF, N_CALIB, N_PARAM,STRING256) ;
    SCIENTIFIC_CALIB_EQUATION:long_name = "Calibration equation for this parameter" ;

char SCIENTIFIC_CALIB_COEFFICIENT(N_PROF, N_CALIB, N_PARAM,STRING256) ;
    SCIENTIFIC_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation" ;

char SCIENTIFIC_CALIB_COMMENT(N_PROF, N_CALIB, N_PARAM,STRING256) ;
    SCIENTIFIC_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration" ;

char CALIBRATION_DATE(N_PROF, N_CALIB, N_PARAM, DATE_TIME) ;
    CALIBRATION_DATE:long_name = "Date of calibration of this parameter" ;
    CALIBRATION_DATE:conventions = "YYYYMMDDHH24MISS" ;

```

History data

// History on profiles QC

// Based on GTSPP format

// One history row = same index for (HISTORY_INSTITUTION, HISTORY_SOFTWARE, ...)

// N_HISTORY is unlimited to allow new history to be added after file creation

```

char HISTORY_INSTITUTION(N_PROF, N_HISTORY, STRING4) ;
    HISTORY_INSTITUTION:long_name = "Institution which performed action" ;
    HISTORY_INSTITUTION:conventions = "GTSPP institution code" ;

char HISTORY_SOFTWARE(N_PROF, N_HISTORY, STRING4) ;
    HISTORY_SOFTWARE:long_name = "Software which performed action" ;
    HISTORY_SOFTWARE:conventions = "Institution dependent" ;

char HISTORY_SOFTWARE_RELEASE(N_PROF, N_HISTORY, STRING4) ;
    HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action" ;
    HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent" ;

char HISTORY_DATE(N_PROF, N_HISTORY, DATE_TIME) ;
    HISTORY_DATE:long_name = "Date the history record was created" ;
    HISTORY_DATE:conventions = "YYYYMMDDHH24MISS" ;

char HISTORY_ACTION(N_PROF, N_HISTORY, STRING4) ;
    HISTORY_ACTION:long_name = "Action performed on data" ;
    HISTORY_ACTION:conventions = "GTSPP (MEDS) action code" ;

char HISTORY_PARAMETER(N_PROF, N_HISTORY, STRING4) ;
    HISTORY_PARAMETER:long_name = "Station parameter action is performed on" ;
    HISTORY_PARAMETER:conventions = "GF3 parameter code" ;

float HISTORY_START_PRES(N_PROF, N_HISTORY) ;
    HISTORY_START_PRES:long_name = "Start pressure action applied on" ;
    HISTORY_START_PRES:_FillValue = 99999.f ;
    HISTORY_START_PRES:units = "decibar" ;

```

```
float HISTORY_STOP_PRES(N_PROF, N_HISTORY) ;  
    HISTORY_STOP_PRES:long_name = "Stop pressure action applied on" ;  
    HISTORY_STOP_PRES:_FillValue = 99999.f ;  
    HISTORY_STOP_PRES:units = "decibar" ;  
  
float HISTORY_PREVIOUS_VALUE(N_PROF, N_HISTORY) ;  
    HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action" ;  
    HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f ;  
  
char HISTORY_QCTEST(N_PROF, N_HISTORY, STRING16) ;  
    HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)" ;  
    HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP$ ;  
                                tests failed when ACTION=QCF$";
```

Interface name	ARGO NETCDF TRAJECTORY DATA FORMAT
Reference	CO-IF0032
Version	2.0
Contact	Thierry.Carval@ifremer.fr
History	03/04/01 TC creation 31/05/01 TC version 1.0 03/04/01 TC version 1.1 12/11/01 TC version 1.2d argo data management meeting updates 22/11/01 TC approved version by argo data management meeting in Brest
comment	-

The differences with the previous version are typed in orange letters.

Dimension definitions

```
// Date and time dimension (YYYYMMDDHH24MISS), utc universal time
DATE_TIME = 14 ;
```

```
// Null terminated string dimensions
```

```
STRING2 = 2 ;
```

```
STRING4 = 4 ;
```

```
STRING8 = 8 ;
```

```
STRING16 = 16 ;
```

```
STRING32 = 32 ;
```

```
STRING64 = 64 ;
```

```
// The following dimensions depend on the data set
```

```
// Number of parameters
```

```
N_PARAM = 3;
```

```
// Number of history records for a location
```

```
N_HISTORY = 7;
```

```
// Number of cycles
```

```
N_CYCLE = 100;
```

```
// Number of measurements
```

```
N_MEASUREMENT = unlimited;
```

Global attributes

```
// Global attributes
```

```
// Empty
```


General information on trajectory**// General information on trajectory**

```

char DATA_TYPE(STRING16);
    DATA_TYPE:comment = "Data type" ;
    // "Argo float";

char FORMAT_VERSION(STRING4);
    FORMAT_VERSION:comment = "File format version " ;
    // Now 2.0

float HANDBOOK_VERSION;
    HANDBOOK_VERSION:comment = "Argo data handbook version " ;
    // Now 1.0

char REFERENCE_DATE_TIME(DATE_TIME) ;
    REFERENCE_DATE_TIME:comment = "Date of reference (convention : YYYYMMDDHH24MI) for Julian days " ;
    REFERENCE_DATE_TIME:conventions = "YYYYMMDDHH24MISS" ;
    // 195001010000 recommended as the first day (day one)

char PROJECT_NAME(STRING64);
    PROJECT_NAME:comment = "Name of the project" ;

char PI_NAME (STRING64);
    PI_NAME:comment = "Principal investigator's identifier" ;

char MEASURED_PARAMETERS(N_PARAM,STRING4) ;
    MEASURED_PARAMETERS:long_name = "List of available parameters for the trajectory" ;
    MEASURED_PARAMETERS:conventions = "GF3 code list in (PRES, TEMP, PSAL, CNDC)";

char PLATFORM_NUMBER (STRING8) ;
    PLATFORM_NUMBER:long_name = "Float unique identifier" ;
    PLATFORM_NUMBER:conventions = "Extended WMO float identifier : QA9IIIII" ;

char DATA_CENTRE(STRING2) ;
    DATA_CENTRE:long_name = "Data centre in charge of float data processing" ;
    DATA_CENTRE:conventions = "GTSP table" ;

char DATE_CREATION(DATE_TIME) ;
    CREATION_DATE:comment = "Date of file creation " ;
    CREATION_DATE:conventions = "YYYYMMDDHH24MISS" ;

char DATE_UPDATE(DATE_TIME) ;
    DATE_UPDATE:long_name = "Date of update of this file" ;
    DATE_UPDATE:conventions = "YYYYMMDDHH24MISS" ;

char DC_REFERENCE(STRING16) ;
    DC_REFERENCE:long_name = "Trajectory unique identifier in data centre" ;
    DC_REFERENCE:conventions = "Data centre convention" ;

char DATA_STATE_INDICATOR(STRING4);
    DATA_STATE_INDICATOR:long_name = "Degree of processing the data have passed through";
    DATA_STATE_INDICATOR:convention = "OOPC table";

char INST_REFERENCE(STRING64) ;
    INST_REFERENCE:long_name = "Instrument type" ;
    INST_REFERENCE:conventions = "Brand/type/serial number" ;

char WMO_INST_TYPE(STRING4) ;
    WMO_INST_TYPE:long_name = "Coded instrument type" ;
    WMO_INST_TYPE:conventions = "WMO probe code table 1770" ;

```

```

char POSITIONING_SYSTEM(STRING8) ;
    POSITIONING_SYSTEM:long_name = "Positioning system" ;
    POSITIONING_SYSTEM:conventions = "ARGOS or GPS" ;

```

Trajectory data

```

// Trajectory data
// Each row uses the same N_MEASUREMENT index for each parameter (date, latitude, longitude, temp...)
//
char DATA_MODE(N_MEASUREMENT);
    DATA_MODE:long_name = "Delayed mode or real time data" ;
    DATA_MODE = "R : real time ; D : delayed mode" ;

double JULD(N_MEASUREMENT) ;
    JULD:long_name = "Julian day (UTC) of each measurement relative to REFERENCE_DATE_TIME " ;
    JULD:units = "days since 1950-01-01 00:00:00 UTC" ;
    JULD:conventions = "Relative julian days with decimal part (as parts of the day)" ;
    JULD:_FillValue = 999999.;

char JULD_QC(N_MEASUREMENT);
    JULD_QC:long_name = "Quality on date and time" ;
    JULD_QC:conventions = "Q where Q =[0-9]" ;
    JULD_QC:_FillValue = "0" ;

double LATITUDE(N_MEASUREMENT);
    LATITUDE:long_name = "Latitude of each location" ;
    LATITUDE:units = "degree_north" ;
    LATITUDE:_FillValue = -99999. ;
    LATITUDE:valid_min = -90. ;
    LATITUDE:valid_max = 90. ;

double LONGITUDE(N_MEASUREMENT);
    LONGITUDE:long_name = "Longitude of each location" ;
    LONGITUDE:units = "degree_east" ;
    LONGITUDE:_FillValue = -99999. ;
    LONGITUDE:valid_min = -180. ;
    LONGITUDE:valid_max = 180. ;

char POSITION_ACCURACY(N_MEASUREMENT);
    POSITION_ACCURACY:long_name = "Estimated accuracy in latitude and longitude" ;
    POSITION_ACCURACY:conventions = "Argos locations classes" ;

char POSITION_QC(N_MEASUREMENT);
    POSITION_QC:long_name = "Quality on position" ;
    POSITION_QC:conventions = "Q where Q =[0-9]" ;
    POSITION_QC:_FillValue = "0" ;

int CYCLE_NUMBER(N_MEASUREMENT);;
    CYCLE_NUMBER:long_name = "Float cycle number of the measurement" ;
    CYCLE_NUMBER:conventions = "0..N, 0 : launch cycle, 1 : first complete cycle" ;
    CYCLE_NUMBER:_FillValue = 99999 ;

float PRES(N_MEASUREMENT) ;
    PRES:long_name = "Pressure";
    PRES:_FillValue = 99999.f ;
    PRES:units = "decibar" ;
    PRES:valid_min = 0. ;
    PRES:valid_max = 15000.f ;
    PRES:C_format = "%7.1f";
    PRES:FORTTRAN_format="F7.1";
    PRES:resolution=0.1f;

```

```

float PRES_CORRECTED(N_MEASUREMENT) ;
    PRES_CORRECTED:long_name = "Corrected pressure" ;
    PRES_CORRECTED:_FillValue = 99999.f ;
    PRES_CORRECTED:units = "decibar" ;
    PRES_CORRECTED:valid_min = 0.f ;
    PRES_CORRECTED:valid_max = 15000.f ;
    PRES_CORRECTED:C_format = "%7.1f" ;
    PRES_CORRECTED:FORTRAN_format="F7.1" ;
    PRES_CORRECTED:resolution=0.1f ;

char PRES_CORRECTED_QC(N_MEASUREMENT) ;
    PRES_CORRECTED_QC:long_name = "Quality on pressure" ;
    PRES_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;
    PRES_CORRECTED_QC:_FillValue = "0" ;

float TEMP(N_MEASUREMENT);
    TEMP:long_name = "Temperature in situ T90 scale" ;
    TEMP:_FillValue = 99999.f ;
    TEMP:units = "degree celsius" ;
    TEMP:valid_min = -3. f ;
    TEMP:valid_max = 40. f ;
    TEMP:comment = "In situ measurement";
    TEMP:C_format = "%9.3f" ;
    TEMP:FORTRAN_format="F9.3" ;
    TEMP:resolution=0.001f ;

float TEMP_CORRECTED(N_MEASUREMENT);
    TEMP_CORRECTED:long_name = "Corrected temperature" ;
    TEMP_CORRECTED:_FillValue = 99999.f ;
    TEMP_CORRECTED:units = "degree celsius" ;
    TEMP_CORRECTED:valid_min = -3. f ;
    TEMP_CORRECTED:valid_max = 40.f ;
    TEMP_CORRECTED:comment = "Corrected value";
    TEMP_CORRECTED:C_format = "%9.3f" ;
    TEMP_CORRECTED:FORTRAN_format="F9.3" ;
    TEMP_CORRECTED:resolution=0.001f ;

char TEMP_CORRECTED_QC(N_MEASUREMENT);
    TEMP_CORRECTED_QC:long_name = "Quality on temperature" ;
    TEMP_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;
    TEMP_CORRECTED_QC:_FillValue = "0" ;

float PSAL(N_MEASUREMENT);
    PSAL:long_name = "Practical salinity, sal78" ;
    PSAL:_FillValue = 99999.f ;
    PSAL:units = "psu" ;
    PSAL:valid_min = 0. f ;
    PSAL:valid_max = 60. f ;
    PSAL:comment = "In situ measurement, mandatory if salinity measured";
    PSAL :C_format = "%9.3f" ;
    PSAL :FORTRAN_format="F9.3" ;
    PSAL:resolution=0.001f ;

float PSAL_CORRECTED(N_MEASUREMENT);
    PSAL_CORRECTED:long_name = "Corrected practical salinity" ;
    PSAL_CORRECTED:_FillValue = 99999.f ;
    PSAL_CORRECTED:units = "psu" ;
    PSAL_CORRECTED:valid_min = 0. f ;
    PSAL_CORRECTED:valid_max = 60. f ;
    PSAL_CORRECTED:comment = " Corrected value";
    PSAL_CORRECTED :C_format = "%9.3f" ;
    PSAL_CORRECTED :FORTRAN_format="F9.3" ;
    PSAL_CORRECTED:resolution=0.001f ;

```

```
char PSAL_CORRECTED_QC(N_MEASUREMENT);  
    PSAL_CORRECTED_QC:long_name = "Quality on practical salinity" ;  
    PSAL_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;  
    PSAL_CORRECTED_QC:_FillValue = "0" ;  
  
float CNDC(N_MEASUREMENT);  
    CNDC:long_name = "Electrical conductivity" ;  
    CNDC:_FillValue = 99999.f ;  
    CNDC:units = "mmho/cm " ;  
    CNDC:valid_min = 0.f ;  
    CNDC:valid_max = 60. f ;  
    CNDC:comment = "Not mandatory, available only if measured";  
    CNDC:C_format = "%9.3f";  
    CNDC:FORTRAN_format="F9.3";  
    CNDC:resolution=0.001f;  
  
float CNDC_CORRECTED(N_MEASUREMENT);  
    CNDC_CORRECTED:long_name = "Corrected electrical conductivity" ;  
    CNDC_CORRECTED:_FillValue = 99999.f ;  
    CNDC_CORRECTED:units = "mmho/cm " ;  
    CNDC_CORRECTED:valid_min = 0. ;  
    CNDC_CORRECTED:valid_max = 60. ;  
    CNDC_CORRECTED:comment = "Not mandatory, available only if measured, corrected value";  
    CNDC_CORRECTED:C_format = "%9.3f";  
    CNDC_CORRECTED:FORTRAN_format="F9.3";  
    CNDC_CORRECTED:resolution=0.001f;  
  
char CNDC_CORRECTED_QC(N_MEASUREMENT) ;  
    CNDC_CORRECTED_QC:long_name = "Quality on electrical conductivity" ;  
    CNDC_CORRECTED_QC:conventions = "Q where Q =[0-9]" ;  
    CNDC_CORRECTED_QC:_FillValue = "0" ;  
    CNDC_CORRECTED_QC:comment = " Not mandatory, available only if measured,";
```

Cycle data

// Informations related to profiler cycles.

```

double JULD_ASCENT_START(N_CYCLE, DATE_TIME) ;
    JULD_ASCENT_START:long_name = " Julian day (UTC) of the ascending profile" ;
    JULD_ASCENT_START:conventions = " Relative julian days with decimal part (as parts of day)" ;
    JULD_ASCENT_START:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD_ASCENT_START:_FillValue = 999999. ;

string JULD_ASCENT_START_STATUS(N_CYCLE) ;
    JULD_ASCENT_START_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 :Transmitted" ;

double JULD_ASCENT_END(N_CYCLE, DATE_TIME) ;
    JULD_ASCENT_END:long_name = " Julian day (UTC) of the ascending profile" ;
    JULD_ASCENT_END:conventions = " Relative julian days with decimal part (as parts of day)" ;
    JULD_ASCENT_END:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD_ASCENT_END:_FillValue = 999999. ;

string JULD_ASCENT_END_STATUS(N_CYCLE) ;
    JULD_ASCENT_END_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 :Transmitted" ;

double JULD_DESCENT_START(N_CYCLE, DATE_TIME) ;
    JULD_DESCENT_START:long_name = "Descent start Julian day (UTC) of the cycle" ;
    JULD_DESCENT_START:conventions = " Relative julian days with decimal part (as parts of day)" ;
    JULD_DESCENT_START:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD_DESCENT_START:_FillValue = 999999. ;

string JULD_DESCENT_START_STATUS(N_CYCLE) ;
    JULD_DESCENT_START_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 :Transmitted" ;

double JULD_DESCENT_END(N_CYCLE, DATE_TIME) ;
    JULD_DESCENT_END:long_name = "Descent end Julian day (UTC) of the cycle" ;
    JULD_DESCENT_END:conventions = " Relative julian days with decimal part (as parts of day)" ;
    JULD_DESCENT_END:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD_DESCENT_END:_FillValue = 999999. ;

string JULD_DESCENT_END_STATUS(N_CYCLE) ;
    JULD_DESCENT_END_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 :Transmitted" ;

double JULD_START_TRANSMISSION(N_CYCLE, DATE_TIME) ;
    JULD_DESCENT_END:long_name = "Start Julian day (UTC) of transmsion" ;
    JULD_DESCENT_END:conventions = " Relative julian days with decimal part (as parts of day)" ;
    JULD_DESCENT_END:units = " days since 1950-01-01 00:00:00 UTC" ;
    JULD_DESCENT_END:_FillValue = 999999. ;

string JULD_START_TRANSMISSION_STATUS(N_CYCLE) ;
    JULD_START_TRANSMISSION_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 :Transmitted" ;

char GROUND(N_CYCLE) ;
    GROUND:long_name = "Did the profiler touch the ground for that cycle";
    GROUND:conventions = "Y,N" ;

```

History data

```

// History on profiles QC
// Based on GTSP format
// One history row = same index for (HISTORY_INSTITUTION, HISTORY_SOFTWARE, ...)
//

char HISTORY_INSTITUTION(N_MEASUREMENT, N_HISTORY, STRING4) ;
    HISTORY_INSTITUTION:long_name = "Institution which performed action";
    HISTORY_INSTITUTION:conventions = "GTSP institution code" ;

char HISTORY_SOFTWARE(N_MEASUREMENT, N_HISTORY, STRING4) ;
    HISTORY_SOFTWARE:long_name = "Software which performed action" ;
    HISTORY_SOFTWARE:conventions = "Institution dependent" ;

char HISTORY_SOFTWARE_RELEASE(N_MEASUREMENT, N_HISTORY, STRING4) ;
    HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action" ;
    HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent" ;

char HISTORY_DATE(N_MEASUREMENT, N_HISTORY, DATE_TIME) ;
    HISTORY_DATE:long_name = "Date the history record was created" ;
    HISTORY_DATE:conventions = "YYYYMMDDHH24MISS" ;

char HISTORY_ACTION(N_MEASUREMENT, N_HISTORY, STRING4) ;
    HISTORY_ACTION:long_name = "Action performed on data" ;
    HISTORY_ACTION:conventions = "GTSP (MEDS) action code" ;

char HISTORY_PARAMETER(N_MEASUREMENT, N_HISTORY, STRING4) ;
    HISTORY_PARAMETER:long_name = "Parameter action is performed on" ;
    HISTORY_PARAMETER:conventions = "GF3 parameter code" ;

float HISTORY_PREVIOUS_VALUE(N_MEASUREMENT, N_HISTORY) ;
    HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action" ;
    HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f ;

char HISTORY_QCTEST(N_HISTORY, N_MEASUREMENT, N_HISTORY, STRING16) ;
    HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)" ;
    HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP$ ;
    tests failed when ACTION=QCF$";

```

Interface name	ARGO TECHNICAL DATA FORMAT	
Reference	-	
Version	2.0	
Comment	-	
Contact	Thierry.Carval@ifremer.fr	
History	12/11/01	TC creation upon arga data management requirement

This format contains technical informations for all the cycles of a profiler.

Dimension definitions

```
// Date and time dimension (YYYYMMDDHH24MISS), UTC universal time
DATE_TIME = 14 ;
```

```
// Null terminated string dimensions
```

```
STRING256 = 256 ;
STRING64  = 64 ;
STRING32  = 32 ;
STRING16  = 16 ;
STRING8   = 8 ;
STRING4   = 4 ;
STRING2   = 2 ;
```

```
// Number of technical parameters, this dimension depends on profiles
N_TECH_PARAM = 25;
```

```
// Number of cycles performed by the profiler
N_CYCLE = UNLIMITED;
```

Global attributes

```
// global attributes
// Empty
```

General information on profiler**// General information on profiler**

```
char DATA_TYPE(STRING32);
    DATA_TYPE:comment = "Argo technical data" ;
    // Argo profiler technical data

char FORMAT_VERSION(STRING4);
    FORMAT_VERSION:comment = "File format version " ;
    // Now 2.0

float ARGO_HANDBOOK_VERSION;
    ARGO_VERSION:comment = "Argo data handbook version" ;
    // Now 1.0

char PLATFORM_NUMBER(STRING8) ;
    PLATFORM_NUMBER:long_name = "Float unique identifier" ;
    PLATFORM_NUMBER:conventions = "WMO float identifier : QA911111" ;

char DATA_CENTRE(STRING2) ;
    DATA_CENTRE:long_name = "Data centre in charge of float data processing" ;
    DATA_CENTRE:conventions = "GTSP table" ;

char DATE_CREATION(DATE_TIME) ;
    DATE_CREATION:comment = "Date of file creation " ;
    DATE_CREATION:conventions = "YYYYMMDDHH24MISS" ;

char DATE_UPDATE(DATE_TIME) ;
    DATE_UPDATE:long_name = "Date of update of this file" ;
    DATE_UPDATE:conventions = "YYYYMMDDHH24MISS" ;
```

Technical parameters

```
// Technical data from the float
// This information is not mandatory
// The name of the parameter is the name given by the manufacturer
```

```
char TECHNICAL_PARAMETER_NAME(N_CYCLE, N_TECH_PARAM, STRING32)
    TECHNICAL_PARAMETER_NAME:long_name="Name of technical parameters for this cycle";

char TECHNICAL_PARAMETER_VALUE(N_CYCLE, N_TECH_PARAM, STRING32)
    TECHNICAL_PARAMETER_VALUE:long_name="Value of technical parameters for this cycle";
```

Interface name	ARGO META -DATA FORMAT	
Reference		
Version	2.0	
Comment	Proposal by Bob Keeley	
Contact		
History	30/08/01	B. Keeley initial proposal
	17/09/01	T. Carval updates
	20/09/01	B. Keeley updates
	10/10/01	Y. Takatsuki comments
	12/10/01	B. Keeley comments
	16/10/01	R. Sabina comments
	17/10/01	B. Keeley comments
	19/10/01	S. Loch comments
	22/10/01	Y. Takatsuki comments
	23/10/01	B. Keeley comments
	23/10/01	S. Pouliquen comments
	12/11/01	Argo dm meeting comments
	22/11/01	Approved version by argo data management meeting in Brest

Dimension definition

```
// Dimension definition

// Date and time dimension (YYYYMMDDHH24MISS)
DATE_TIME = 14 ;

// Null terminated string dimensions
STRING256 = 256 ;
STRING64 = 64 ;
STRING32 = 32 ;
STRING16 = 16 ;
STRING8 = 8 ;
STRING4 = 4 ;
STRING2 = 2 ;

// These dimensions depends on the float
N_CYCLES = 1 ;
N_PARAM=3 ;
N_TECH_PARAM = 25;
```

General information

```
// General information
```

```
char DATA_TYPE(STRING16);  
    DATA_TYPE:comment = "Data type" ;  
    // Argo meta data  
  
char FORMAT_VERSION(STRING4);  
    FORMAT_VERSION:comment = "File format version " ;  
    // Now 2.0  
  
float HANDBOOK_VERSION;  
    HANDBOOK_VERSION:comment = "Argo data handbook version" ;  
    // Now 1.0  
  
char DATE_CREATION(DATE_TIME) ;  
    DATE_CREATION:comment = "Date of file creation " ;  
    DATE_CREATION:conventions = "YYYYMMDDHH24MISS" ;  
  
char DATE_UPDATE(DATE_TIME) ;  
    DATE_UPDATE:long_name = "Date of update of this file" ;  
    DATE_UPDATE:conventions = "YYYYMMDDHH24MISS" ;  
  
char PROJECT_NAME(STRING32) ;  
    PROJECT_NAME:long_name = "The program under which the float was deployed" ;  
  
char DATA_CENTRE(STRING2) ;  
    DATA_CENTRE:long_name = "Data centre in charge of float real-time processing" ;  
    DATA_CENTRE:Conventions = "GTSP table" ;  
  
char PI_NAME (STRING64);  
    PI_NAME:comment = "Name of the primary investigator" ;  
  
char ANOMALY(STRING256) ;  
    ANOMALY:long_name = "Describe any anomalies or problems the float may have had" ;
```

Float characteristics

// Float characteristics

```

char PLATFORM_NUMBER(STRING8) ;
    PLATFORM_NUMBER:long_name = "Float unique identifier" ;
    PLATFORM_NUMBER:Conventions = "Extended WMO float identifier : QA911111" ;

char PTT (STRING256) ;
    PTT:long_name = "Transmission identifier (Argos, Orbcomm, etc.)" ;
    // Comma separated list for multi-beacons transmission

char TRANS_SYSTEM(STRING16) ;
    TRANS_SYSTEM:long_name = "The telecommunications system used" ;
    TRANS_SYSTEM:Conventions = "ARGOS or ORBCOMM" ;

char TRANS_SYSTEM_ID(STRING32) ;
    TRANS_SYSTEM_ID:long_name = "The program identifier used by the transmission system" ;

char TRANS_FREQUENCY(STRING16) ;
    TRANS_FREQUENCY:long_name = "The frequency of transmission from the float" ;
    TRANS_FREQUENCY:units = "hertz" ;

float TRANS_REPETITION ;
    TRANS_REPETITION:long_name = "The repetition rate of transmission from the float" ;
    TRANS_REPETITION:units = "second" ;

char POSITIONING_SYSTEM(STRING8) ;
    POSITIONING_SYSTEM:long_name = "Positioning system" ;
    POSITIONING_SYSTEM:Conventions = "ARGOS or GPS" ;

float CLOCK_DRIFT ;
    CLOCK_DRIFT:long_name = "The rate of drift of the float clock" ;
    CLOCK_DRIFT:units = "decisecond/day" ;
    CLOCK_DRIFT:_FillValue = "99999.f" ;

char PLATFORM_MODEL (STRING16) ;
    PLATFORM_MODEL:long_name = "Model of the float " ;

char PLATFORM_MAKER (STRING256) ;
    PLATFORM_MAKER:long_name = "The name of the manufacturer " ;

char PLATFORM_SERIAL_NO(STRING16) ;
    PLATFORM_SERIAL_NO:long_name = "The serial number of the sensor " ;

char DIRECTION;
    DIRECTION:long_name = "Direction of the station profiles" ;
    DIRECTION:Conventions = "A: ascending profiles, B: descending and ascending profiles " ;

```

Float deployment information**// Float deployment information**

```

char LAUNCH_DATE(DATE_TIME) ;
    LAUNCH_DATE:long_name = "Date (UTC) of the deployment" ;
    LAUNCH_DATE:Conventions = "YYYYMMDDHH24MISS" ;

double LAUNCH_LATITUDE;
    LAUNCH_LATITUDE:long_name = "Latitude of the float when deployed" ;
    LAUNCH_LATITUDE:units = "degrees_north" ;
    LAUNCH_LATITUDE:_FillValue = 99999. ;
    LAUNCH_LATITUDE:valid_min = -90. ;
    LAUNCH_LATITUDE:valid_max = 90. ;

double LAUNCH_LONGITUDE;
    LAUNCH_LONGITUDE:long_name = "Longitude of the float when deployed" ;
    LAUNCH_LONGITUDE:units = "degrees_east" ;
    LAUNCH_LONGITUDE:_FillValue = 99999. ;
    LAUNCH_LONGITUDE:valid_min = -180. ;
    LAUNCH_LONGITUDE:valid_max = 180. ;

char LAUNCH_QC;
    LAUNCH_DATE_QC:long_name = "Quality on launch date, time and location" ;
    LAUNCH_DATE_QC:conventions = "Q where Q =[0-9]" ;
    LAUNCH_DATE_QC:_FillValue = "0" ;

char START_DATE(DATE_TIME) ;
    START_DATE:long_name = "Date (UTC) of the first descent of the float." ;
    START_DATE:Conventions = "YYYYMMDDHH24MISS" ;

char START_DATE_QC;
    START_DATE_QC:long_name = "Quality on launch date, time and location" ;
    START_DATE_QC:conventions = "Q where Q =[0-9]" ;
    START_DATE_QC:_FillValue = "0" ;

char DEPLOY_PLATFORM(STRING32) ;
    DEPLOY_PLATFORM:long_name = "Identifier of the deployment platform" ;

char DEPLOY_MISSION(STRING32) ;
    DEPLOY_MISSION:long_name = "Identifier of the mission used to deploy the float" ;

char DEPLOY_AVAILABLE_PROFILE_ID(STRING256) ;
    DEPLOY_AVAILABLE_PROFILE_ID:long_name = "Identifier of a stations used to verify the first profile" ;

```

Profiler sensor information**// Profiler sensor information**

```
char SENSOR(N_PARAM,STRING4) ;  
    SENSOR:long_name = "List of sensors on the float " ;  
    SENSOR:Conventions = "GF3 code list in (PRES, TEMP, CNDC)";  
  
char SENSOR_MAKER(N_PARAM,STRING256) ;  
    SENSOR_MAKER:long_name = "The name of the manufacturer " ;  
  
char SENSOR_MODEL (N_PARAM,STRING256) ;  
    SENSOR_MODEL:long_name = "Type of sensor " ;  
  
char SENSOR_SERIAL_NO(N_PARAM,STRING16) ;  
    SENSOR_SERIAL_NO:long_name = "The serial number of the sensor " ;  
  
char SENSOR_UNITS(N_PARAM, STRING16) ;  
    SENSOR_UNITS:long_name = "The units of accuracy and resolution of the sensor" ;  
  
float SENSOR_ACCURACY(N_PARAM) ;  
    SENSOR_ ACCURACY:long_name = "The accuracy of the sensor" ;  
  
float SENSOR_RESOLUTION(N_PARAM) ;  
    SENSOR_RESOLUTION:long_name = "The resolution of the sensor" ;
```

Float calibration information**// Float calibration information**

```
char PARAMETER(N_PARAM,STRING4) ;  
    PARAMETER:long_name = "List of parameters with calibration information" ;  
    PARAMETER:Conventions = "GF3 code list in (PRES, TEMP, CNDC, CPV$, PMV$)";  
  
char PREDEPLOYEMENT_CALIB_EQUATION(N_PARAM,STRING256) ;  
    PREDEPLOYEMENT_CALIB_EQUATION:long_name = "Calibration equation for this parameter" ;  
  
char PREDEPLOYEMENT_CALIB_COEFFICIENT (N_PARAM,STRING256) ;  
    PREDEPLOYEMENT_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation" ;  
  
char PREDEPLOYEMENT_CALIB_COMMENT (N_PARAM,STRING256) ;  
    PREDEPLOYEMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration" ;
```

Profile cycle information

// Information about float profiling characteristics

```

int REPETITION_RATE(N_CYCLES);
    REPETITION_RATE:long_name = "The number of times this cycle repeats" ;
    REPETITION_RATE:units = "number" ;
    REPETITION_RATE:_FillValue = 99999.f ;

float CYCLE_TIME(N_CYCLES);
    CYCLE_TIME:long_name = "The total time of a cycle : descent + parking + ascent + surface" ;
    CYCLE_TIME:units = "decimal days" ;
    CYCLE_TIME:_FillValue = 99999.f ;

float PARKING_TIME(N_CYCLES);
    PARKING_TIME:long_name = "The time spent at the parking pressure" ;
    PARKING_TIME:units = "decimal days" ;
    PARKING_TIME:_FillValue = 99999.f ;

float DESCENDING_PROFILING_TIME(N_CYCLES);
    DESCENDING_PROFILING_TIME:long_name = "The time spent sampling the descending profile" ;
    DESCENDING_PROFILING_TIME:units = "decimal hours" ;
    DESCENDING_PROFILING_TIME:_FillValue = 99999.f ;

float ASCENDING_PROFILING_TIME(N_CYCLES);
    ASCENDING_PROFILING_TIME:long_name = "The time spent sampling the ascending profile" ;
    ASCENDING_PROFILING_TIME:units = "decimal hours" ;
    ASCENDING_PROFILING_TIME:_FillValue = 99999.f ;

float SURFACE_TIME(N_CYCLES);
    SURFACE_TIME:long_name = "The time spent at the surface." ;
    SURFACE_TIME:units = "decimal hours" ;
    SURFACE_TIME:_FillValue = 99999.f ;

float PARKING_PRESSURE(N_CYCLES);
    PARKING_PRESSURE:long_name = "The pressure of immersion drifts" ;
    PARKING_PRESSURE:units = "decibar" ;
    PARKING_PRESSURE:_FillValue = 99999.f ;

float DEEPEST_PRESSURE(N_CYCLES);
    DEEPEST_PRESSURE:long_name = "The deepest pressure sampled in the ascending profile" ;
    DEEPEST_PRESSURE:units = "decibar" ;
    DEEPEST_PRESSURE:_FillValue = 99999.f ;

float DEEPEST_PRESSURE_DESCENDING(N_CYCLES);
    DEEPEST_PRESSURE_DESCENDING:long_name = "The deepest pressure sampled in the descending profile" ;
    DEEPEST_PRESSURE_DESCENDING:units = "decibar" ;
    DEEPEST_PRESSURE_DESCENDING:_FillValue = 99999.f ;

```

Technical parameters

```
// Additional technical data about the float  
// This information is not mandatory
```

```
CHAR TECHNICAL_PARAMETER_NAME(N_TECH_PARAM, STRING32)  
    TECHNICAL_PARAMETER_NAME:long_name="Name of technical parameters for this float";
```

```
CHAR TECHNICAL_PARAMETER_VALUE(N_TECH_PARAM, STRING32)  
    TECHNICAL_PARAMETER_VALUE:long_name="Value of technical parameters for this float";
```

Interface name	ARGO REFERENCE TABLES
Version	2.0
Comment	-
Contact	Thierry.Carval@ifremer.fr
History	12/11/01 TC creation

This document contains Arga data reference tables.

Quality flags	
Value	Description
0	Unqualified
1	Correct value (all checks passed)
2	Probably good but value inconsistent with statistics (differ from climatology)
3	Probably bad (spike, gradient, ... if other tests passed)
4	Bad value, Impossible value (out of scale, vertical instability, constant profile, ...)
5	Value modified during quality control
6-7	Not used (available)
8	Interpolated value
9	Missing value

Data centres and institutions	
ME	MEDS (Canada)
AO	AOML (USA)
SI	SIO (USA, Scripps)
CS	CSIRO (Australia)
GE	Germany
JA	Japan
JM	Jamstec (Japan)
IF	Ifremer (France)
RU	Russia
SP	Spain

Argos location classes	
Value	Estimated accuracy in latitude and longitude
3	< 150 m
2	150 m <= accuracy < 350 m
1	350 m <= accuracy < 1000 m
0	> 1000 m
A	No estimate of location accuracy
B	No estimate of location accuracy
Z	Invalid location

OOPC data state indicators

OOPC Level	Descriptor
0	Data are the raw output from instruments, without calibration, and not necessarily converted to engineering units. These data are rarely exchanged
1	Data have been converted to values independent of detailed instrument knowledge. Automated calibrations may have been done. Data may not have full geospatial and temporal referencing, but have sufficient information to uniquely reference the data to the point of measurement.
2	Data have complete geospatial and temporal references. Information may have been compressed (e.g. subsampled, averaged, etc.) but no assumptions of scales of variability or thermodynamic relationships have been used in the processing.
3	The data have been processed with assumptions about the scales of variability or thermodynamic relationships. The data are normally reduced to regular space, time intervals with enhanced signal to noise.

OOPC Class	SubclassDescriptor
A	<p>No scrutiny, value judgements or intercomparisons are performed on the data. The records are derived directly from the input with no filtering, or subsampling.</p> <ul style="list-style-type: none"> - Some reductions or subsampling has been performed, but the original record is available. <p>+ Geospatial and temporal properties are checked. Geophysical values are validated. If not validated, this is clearly indicated.</p>
B	<p>Data have been scrutinized and evaluated against a defined and documented set of measures. The process is often automated (i.e. has no human intervention) and the measures are published and widely available.</p> <ul style="list-style-type: none"> - Measures are completely automated, or documentation is not widely available. <p>+ The measures have been tested on independent data sets for completeness and robustness and are widely accepted.</p>
C	<p>Data have been scrutinized fully including intra-record and intra-dataset comparison and consistency checks. Scientists have been involved in the evaluation and brought latest knowledge to bear. The procedures are published, widely available and widely accepted.</p> <ul style="list-style-type: none"> - Procedures are not published or widely available. Procedures have not undergone full scrutiny and testing. <p>+ Data are fully quality controlled, peer reviewed and are widely accepted as valid. Documentation is complete and widely available.</p>

Information on HISTORY_QCTEST handling

Every time a record passes through a QC test, a record is created in HISTORY.
 The nature of QC test is stored in HISTORY_QCTEST.
 The HISTORY_ACTION is set to QCP\$: QC performed.
 QC performed only means that the test was performed, whether it failed or not.

If a test failed a second record is created in HISTORY.
 The nature of the failed test is stored in HISTORY_QCTEST.
 The HISTORY_ACTION is set to QCF\$: QC failed.

For example, assume the case of a profile with a test failure.
 The history structure would be as follows.

Note that I have used a parameter code of RCRD to indicate that the result applies to the entire record, not just to any particular parameter.

HISTORY_INSTITUTION	MEDS
HISTORY_SOFTWARE	QCA1
HISTORY_SOFTWARE_RELEASE	1.0
HISTORY_ACTION	QCP\$
HISTORY_PARAMETER	RCRD
HISTORY_START_PRES	99999.
HISTORY_STOP_PRES	99999.
HISTORY_PREVIOUS_VALUE	99999.
HISTORY_QCTEST	6A34CFE

HISTORY_INSTITUTION	MEDS
HISTORY_SOFTWARE	QCA1
HISTORY_SOFTWARE_RELEASE	1.0
HISTORY_ACTION	QCF\$
HISTORY_PARAMETER	RCRD
HISTORY_START_PRES	99999.
HISTORY_STOP_PRES	99999.
HISTORY_PREVIOUS_VALUE	99999.
HISTORY_QCTEST	0A