09 AVR. 1997

Intergovernmental
Oceanographic
Commission

Manuals and Guides No. 17

GF3

A GENERAL FORMATTING SYSTEM FOR GEO-REFERENCED DATA

VOLUME 1

INTRODUCTORY GUIDE TO THE GF3 FORMATTING SYSTEM

1993 UNESCO

CONTENTS

		Page
FOREWORD		i
ACKNOWLEDGEMENTS		ii
1.	**INTRODUCTION**	1
1.1	WHAT IS GF3?	1
1.2	HISTORICAL BACKGROUND	1
1.3	OVERVIEW OF GF3	1
1.4	THE CONCEPT OF STANDARD SUBSETS	2
2.	**HOW DATA ARE ORGANIZED IN GF3**	2
2.1	BASIC TAPE STRUCTURE	2
2.2	TAPE FILES	2
2.3	RECORD STRUCTURES AND THEIR USAGE	4
2.4	THE HIERARCHICAL NATURE OF GF3	5
3.	**EXAMPLES OF THE USE OF GF3**	5
3.1	SAMPLE STRUCTURE OF A TAPE CONTAINING CTD DATA COLLECTED DURING AN OCEANOGRAPHIC CRUISE	5
3.2	SAMPLE STRUCTURE OF A TAPE CONTAINING WAVE SPECTRA COLLECTED AT SEVERAL OBSERVING STATIONS	13
3.3	SAMPLE STRUCTURE OF A TAPE CONTAINING DRIFTING BUOY DATA	16
4.	**AN INTRODUCTION TO GF3-PROC**	19
4.1	THE KEY FEATURES OF GF3-PROC	19
4.2	PROGRAMMING ENVIRONMENT OF GF3-PROC	20
4.3	THE GF3-PROC USER INTERFACE	20
4.4	PROGRAMMING BENEFITS OF GF3-PROC	21
4.5	GF3-PROC PORTABILITY	22
5.	**EXAMPLES OF THE USE OF GF3-PROC**	22
5.1	INTRODUCTION	23
5.2	COPYING A DATA SERIES FROM TAPE TO DISK	23
5.3	CREATING A CURRENT METER DATA SET USING GF3-PROC	27
6.	**AVAILABLE SUPPORT FOR GF3 AND GF3-PROC USERS**	31
6.1	GF3 TAPE INSPECTION UTILITIES	32
6.2	GF3 INTERFACE PROGRAMMES	32
7.	**FUTURE DEVELOPMENT OF GF3**	32

FOREWORD

The General Format 3 (GF3) system was developed by the IOC Committee on International Oceanographic Data and Information Exchange (IODE) as a generalized formatting system for the exchange and archival of data within the international oceanographic community. It was presented to the Ninth Session of the Technical Committee (New York, 15-19 January 1979) which recommended that GF3 be adopted for general use in international oceanographic data exchange and urged Member States to utilize GF3 as the standard international exchange format. This recommendation was subsequently endorsed by the IOC Executive Council at its Eleventh Session (Mexico City, 1-3 March, 1979).

The GF3 format is supported by a comprehensive software package, GF3-Proc, which the IOC is prepared to make freely available on magnetic tape to all organizations or laboratories involved in the international collection, management or exchange of oceanographic and other earth sciences data. Technical support for the distribution, installation and maintenance of GF3-Proc is provided, on behalf of the IOC, by the British Oceanographic Data Centre (BODC). Requests for copies of GF3-Proc should be forwarded to BODC at the address given and should include a clear description of the computer system on which it is to be installed, including the manufacturer, make and model number of the machine, the name and version of the operating system and an identification of the FORTRAN compiler. A small charge may be made to cover the cost of the tape and its documentation.

Requests for technical advice and guidance on the use of GF3 and GF3-Proc should be addressed to:

British Oceanographic Data Centre
Proudman Oceanographic Laboratory
Bidston Observatory
Birkenhead, Merseyside, L43 7RA
UNITED KINGDOM

Support services explaining GF3 and its use are provided by the Service Hydrographique of the International Council for the Exploration of the Sea (ICES), acting as the Responsible National Oceanographic Data Centre for Formats, RNODC (Formats). The ICES Service Hydrographique is assisted in this task by the British Oceanographic Data Centre. Enquiries concerning these services should be addressed to:

RNODC (Formats)
International Council for the Exploration of the Sea
Service Hydrographique
Palaegade 2-4
DK 1261 Copenhagen K
DENMARK

The use and development of the GF3 system is kept under review by the IOC Group of Experts on Technical Aspects of Data Exchange (GETADE). Comments and suggestions for the improvement of GF3 may be forwarded to the Chairman of this Group either through the RNODC (Formats) or through the IOC Secretariat.

The documentation for the GF3 system is published in IOC Manuals and Guides N° 17 in 6 separate volumes under the title GF3 - A General Formatting System for Geo-Referenced Data.

Volume 1: *Introductory Guide to the GF3 Formatting System* is intended to familiarize the new user with the purpose and scope of the GF3 system without overburdening him with technical detail. An introduction is provided, illustrated by examples, both to the GF3 format and to its supporting software package GF3-Proc.

Volume 2: *Technical Description of the GF3 Format and Code Tables* contains a detailed technical specification of the GF3 format and its associated code tables.

Volume 3: *Standard Subsets of the GF3 Format* contains a description of standard subsets of the GF3 format tailored to a range of different types of data. It also serves as a set of examples illustrating how the GF3 format may be used.

Volume 4: *User Guide to the GF3-Proc Software* provides an overview of GF3-Proc explaining what it does, how it works and how it is used. It also provides an introduction to the subroutine calls in the user interface to the package.

Volume 5: *Reference Manual for the GF3-Proc Software* contains a detailed specification of each GF3-Proc subroutine callable from the users programme and provides detailed instruction on how and when these routines may be used.

Volume 6: *Quick Reference Sheets for GF3 and GF3-Proc* contains quick and easy reference sheets to the GF3 format and the GF3-Proc software.

ACKNOWLEDGEMENTS

The design and technical specification of the GF-3 format were prepared by Mr. Meirion T. Jones of the UK British Oceanographic Data Center of the Institute of Oceanographic Sciences, working in close collaboration with the IODE Group of Experts on Format Development (later GE on Technical Apects of Data Exchange) and with Mr. P. Winiarski of Germany. This document was finalized by an *ad hoc* Group of Experts working at the Marine Environmental Data Service (MEDS), Canada.

1. INTRODUCTION

1.1 WHAT IS GF3?

GF3 is a general purpose format scheme which has been developed for use in the exchange of data within the international oceanographic community. It will be equally of value for meteorological, geophysical and other environmental data.

It is a highly flexible, self-documenting magnetic tape system designed primarily for numerical data. It is not, however, restricted to numerical data, as the variety of structures available permit the inclusion of textual information in several ways.

The scheme was developed to facilitate the exchange and dissemination on magnetic tape of many types of oceanographic data, ranging from the most simple cases to complex multi-disciplinary datasets. For certain types of data such as project datasets, however, GF3 could be the most logical archival format.

GF3 is not recommended as a real time telecommunications format. It was not designed to be efficient for such a carrier.

1.2 HISTORICAL BACKGROUND

GF3 is based on the experience of the format used for the exchange of oceanographic and meteorological data from the GARP Atlantic Tropical Experiment (GATE). This experiment was part of the Global Atmospheric Research Programme (GARP) of the World Meteorological Organization (WMO) and the International Council of Scientific Unions (ICSU).

During GATE, a format was developed which introduced the idea of self-documenting, multi-disciplinary, magnetic tape files amenable to automatic processing. The value of a format system of this type to the international exchange system was immediately recognized. Such a system avoids the unending development of detailed character-by-character formats for every new type of data and measuring instrument. The problem of incorporating new data variables or old data variables from new instrumentation into the international exchange system could be reduced to specifying standards and content.

Having recognized the value of such a formatting system, the Working Committee on IODE proceeded with the development, which resulted in GF3 being adopted by its Ninth Session, held in New York in January 1979. The numeral 3 in GF3 does not mean there are 3 general format schemes which have been approved for use by the IOC Working Committee of IODE. GF1 & GF2 were phases in the development of the GF3 format.

Persons wishing to exchange data can format the magnetic tapes in any of several convenient formats conforming to the rules of the GF3 scheme. A user receiving such a tape finds all the necessary information to interpret and use the data included on the magnetic tape. If the user has GF3-Proc software to take advantage of the automatic processing features of the system, it is not necessary to know the detailed format of the tape or develop new software to extract and process the data. An example of this is a user who is routinely extracting sea temperature from GF3 bathythermograph tapes and is producing section plots of temperature versus depth. If the user is taking full advantage of the automatic processing capabilities of the system and receives a GF3 tape of CTD data, the processing software need not be modified in spite of the vastly different format and content of the tape.

1.3 OVERVIEW OF GF3

The GF3 format system was developed to meet a number of specifications.

(i) The format was to consist of simple structures so that it could be used by single scientific users and small institutions, as well as large data centers.

(ii) The format was to be largely self-documenting through the provision of plain language comment capabilities at all levels of the structure and through inclusion of formatting information and character coding information on the tape.

(iii) The format was to be capable of being processed automatically by the user or data centre receiving the tape.

(iv) The structures of the format were to be capable of transmitting complex, multi-disciplinary data, as well as the most simple data collections.

(v) The format was to be a magnetic tape format for the exchange of data and in many cases was to be suitable for archiving of these data.

GF3 has been designed to facilitate automatic processing. The self-documenting aspect of the system is one of its more useful and elegant features. To the user receiving a GF3 tape, only the recording density and the fact that the tape is in the GF3 format need to be known in advance. Foreknowledge of the detailed formatting is not required. All the necessary information to interpret and understand the contents of the tape is included on the tape in fixed positions in the various record structures.

The flexibility of the format results from the variety and number of possible usages and combinations of the GF3 record types. This makes it possible to include, within the format scheme, structures from the very simple to those which can contain multi-disciplinary data with several levels of hierarchy. It has been found possible to encode in GF3 physical, chemical, biological, geological, meteorological and geophysical data.

1.4 THE CONCEPT OF STANDARD SUBSETS

The international collection and exchange of oceanographic data is often by type of data. For example, many countries collect and exchange CTD and drifting buoy data. To simplify this kind of exchange, the concept of standard subsets was introduced. The standard subset is a predefined format within the GF3 system for use as appropriate in the exchange of standard types of data.

A standard subset is constructed by predefining definition records on a magnetic tape which describe the contents of the tape and its detailed format.

The user is thus spared the task of preparing the content for these header fields and can proceed directly with the necessary programming to create data tapes.

Another advantage of the standard subset concept is that it presents the user with a detailed format on which more complicated structures can be built. For example, if a format is required for a dataset which is quite similar to a standard subset except that more variables were observed, then it is relatively easy to add parameters to a standard subset, provided the hierarchical structure of the dataset is not altered.

Information on the availability of standard subsets and documentation is available from the RNODC (Formats).

2. HOW DATA ARE ORGANIZED IN GF3

2.1 BASIC TAPE STRUCTURE

GF3 tapes are character oriented digital magnetic tapes. The information carried on a tape is contained in the most basic element of the structure, the GF3 record. Records are organized into files according to the rules of GF3, and files are organized into tapes. A GF3 dataset can be contained on one tape or, if large, it can occupy several tapes.

Not all files on a GF3 tape need have the same structure. If convenient, a multi-disciplinary dataset from a large experiment might contain physical and biological oceanographic data in some files, meteorological and surface wave data in still other files, or indeed a mixture of these. No two files would necessarily be formatted identically or contain the same parameters. Each file would have in it all the formatting information to decode it and interpret the data.

2.2 TAPE FILES

Figure 1 shows the basic file structure of all GF3 tapes. There are 4 types of files:

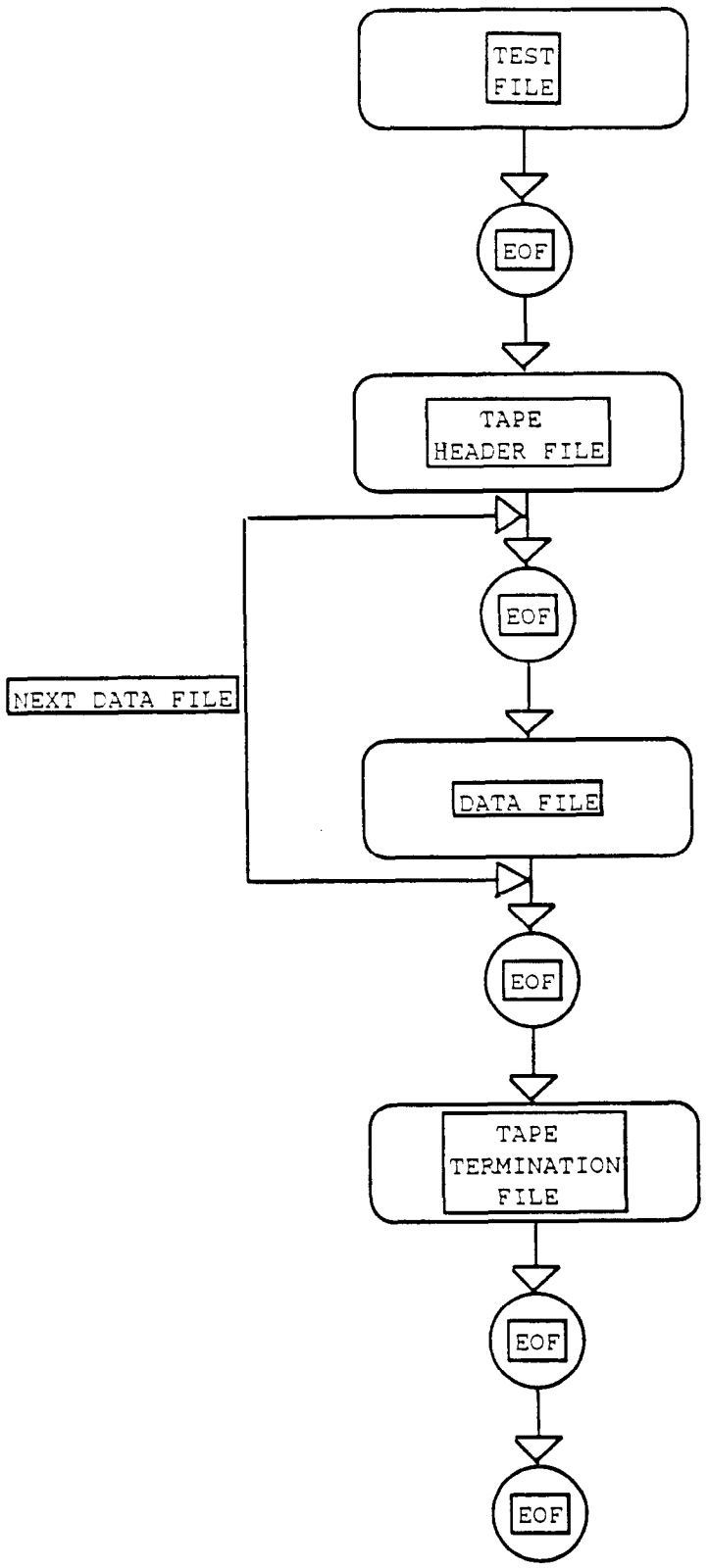

FIGURE 1. GF-3 TAPE STRUCTURE

(i) The Test File is the first file and the primary purpose is to protect the leading portion on the tape from damage. It contains a many times repeated character which may also be helpful in detecting misalignment.

(ii) The Tape Header File provides administrative information about the tape and the originating data center. Examples of this information include tape number, the originating country and institution, the type of computer used, and the date the tape was created. It also identifies the version of GF3 used in setting up this tape.

(iii) The Data Files contain the data. These files are constructed using the record types provided by GF3. The records employed and their arrangement depend on the data being exchanged and their logical assembly into some sort of hierarchy.

(iv) The Tape Termination File appears as the last file on the tape and is used to terminate the present tape and indicate the next tape if the data are contained on another tape.

A GF3 tape will always contain one and only one Test File, Tape Header File, and Tape Termination File. There will be one or more Data Files, as necessary.

2.3 RECORD STRUCTURES AND THEIR USAGE

GF3 recognizes 7 types of physical records:

Test Record
Plain Language Record
Tape Header Record
Definition Record
File/Series Header Record
Data Cycle Record
End of Tape Record

The relationship between these records and the files is illustrated in Figure 2.

(i) The Test Record consists of the character A repeated over 1920 bytes.

(ii) The Plain Language Record consists of free format textual information and is used to provide information and comments. These records can be inserted at the tape level to provide information pertinent to the whole tape, at the file level to provide information pertinent to a file, and at the series level to provide information on a single series.

Liberal use of comments is encouraged in GF3. For example, the scientists can describe their experiments in their own words to convey the guidance and opinions that are important to the user of the data. One user has structured the Plain Language Records so that when printed, a diagram of the mooring used on the wave buoy is presented on the listing. Other users have included in the comments coding tables for use by the persons receiving the data.

(iii) The Tape Header Record is a completely fixed-field record which is used to provide administrative information about the originating data center and the tape, as well as comments at the tape level.

(iv) The Definition Record is the key to the GF3 formatting system. It is used to define the contents of the user-formatted portion of the Series Header Records and of the Data Cycle Records. These records can be interpreted by a computer to determine what parameters are on the tape, where they are located in the records, what units they are in, and what format they are in. The Definition Record makes possible the automatic processing features of GF3.

(v) The File and Series Header Records contain a variety of fixed format information about the data in the file and series which follow. Examples include position, time and platform or ship information. The Series Header Record also contains a user-formatted area which includes other data pertaining to the series not provided for in the fixed format area.

(vi) The Data Cycle Record is used to carry data. This record is entirely user formatted, except for 20 bytes of administration information.

(vii) The End of Tape Record appears only in the Tape Termination File and tells the user if the data are continued on another tape or if this is the last tape in the set.

2.4 THE HIERARCHICAL NATURE OF GF3

Scientific projects generally produce data which can be arranged in a hierarchical manner. For a given experiment, there may be certain data which are constant for the whole experiment, such as the calibration data for the instruments. Within the experiment, there may be a number of runs. For each run, there will probably be variables, such as initial conditions, which are constant for the run. Within a run, many measurements of the phenomenon of interest will be made.

The structures available in GF3 are well suited to represent the logical nature of many datasets. The following section contains examples of how various types of data can be loaded into GF3. Some examples demonstrate up to 4 levels of hierarchy: the tape level, the file level, the series level and the data cycle level. In fact, a fifth level is available with the use of header parameters in the data cycle records.

3. EXAMPLES OF THE USE OF GF3

The following examples of the use of GF3 have been drawn from existing standard subsets. The first of these is discussed in considerable detail to illustrate the structure of GF3.

3.1 SAMPLE STRUCTURE OF A TAPE CONTAINING CTD DATA COLLECTED DURING AN OCEANOGRAPHIC CRUISE

The characteristics of a cruise with CTD (Conductivity-Temperature-Depth) observations are generally that the ship follows a track, stops at certain points and lowers the CTD instrument package. The package measures pressure, temperature, conductivity (convertible to salinity) and possibly oxygen content as it descends through the water column. The package will also generally have a number of water bottles attached. At each of several preplanned depths, a bottle will close, capturing a water sample. These samples are analyzed for other parameters of the water and also for an independent salinity measurement to serve as a calibration point for the CTD probe.

A lowering of the CTD at a location is referred to as an oceanographic station. The data from a station are considered to be a series in the GF3 standard subset. A data file is composed of a file header record and a number of series equal to the number of stations in the cruise. If the tape contains data from several cruises, then there will be several data files, one for each cruise.

The structure of a tape in the CTD standard subset format is shown in Figure 2.

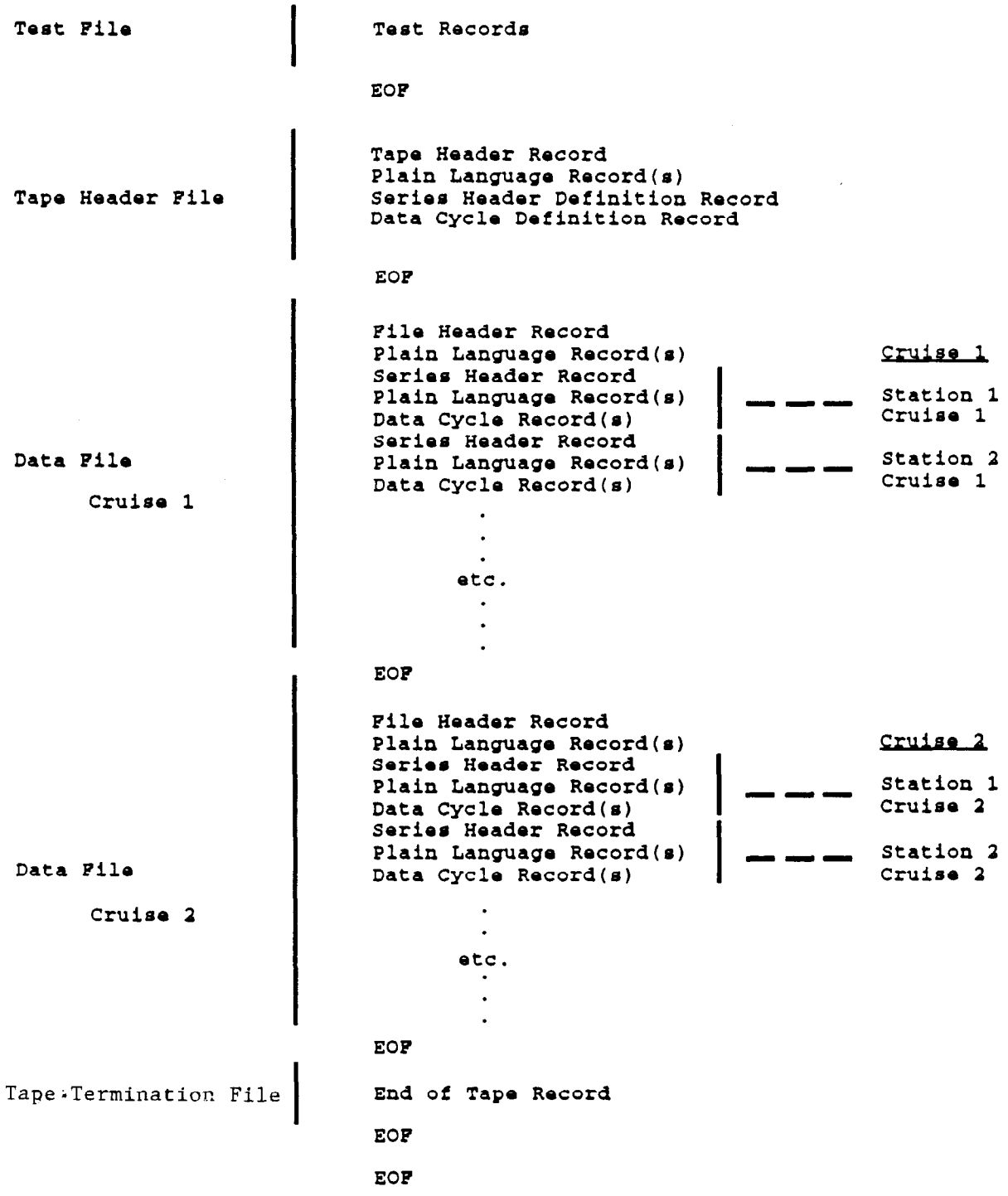

FIGURE 2. STRUCTURE OF A GF-3 TAPE FOR THE CTD STANDARD SUBSET

The Tape Header File contains the detailed information on the format of the user specified portions of the Series Header and Data Cycle Records. The detailed formatting of the data is obtained by printing and inspecting the pertinent fixed fields in this file. Alternatively, these fields are read by the computer and used to automatically process the Series Header and Data Cycle Records.

In the example in Figure 2, the Definition Records all appear in the Tape Header File and nowhere else. This means that, in this case, all Series Header Records and Data Cycle Records must be in the same format. This is not a necessary restriction. The format and content of the Series Header and Data Cycle Records can be redefined at any point in the tape by including a new Definition Record before the first occurrence of the altered record structure.

Figure 3 shows the Series Header Definition Record for the CTD standard subset. The user-formatted area in the Series Header Records is being used for the calibration data from the water bottles. Positions 3 to 10 of each record sequence contains the 8-character parameter code. The parameter code is used to identify the parameter that has been measured, to differentiate between the
same parameter measured by different techniques and to indicate whether the parameter is reported in standard GF3 units (SI). Note that the parameter codes for pressure, temperature and salinity measured by the CTD are different from those for the same parameters measured by the water bottles and reversing thermometers.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

34  0    6I        (38(I5,lX,I5,lX,I5,lX,I5,lX,I5,lX,I5,5X))                        001
3                                                                                   002
3                                                                                   003
3   PRES7PRD      SEA PRESSURE (CTD)          DBAR.I  5-94  0.1    0              004
3   TEMP7STD      SEA TEMPERATURE (CTD)       DEG.CI  5-94  0.001  0              005
3   PSAL7PRD      PRAC.SALINITY (CTD)         I       5-94  0.001  0              006
3   PRES7RTD      SEA PRESSURE (THERM)        DBAR.I  5-94  0.1    0              007
3   TEMP7RTD      SEA TEMPERATURE (THERM)     DEG.CI  5-94  0.001  0              008
3   PSAL7BSD      PRAC.SALINITY (BOTTLE)      I       5-94  0.001  0              009
3                                                                                   010
3                                                                                   011
3                                                                                   012
3                                                                                   013
3                                                                                   014
3                                                                                   015
3                                                                                   016
3                                                                                   017
3                                                                                   018
3                                                                                   019
3                                                                                   020
3                                                                                   021
3                                                                                   022
3                                                                                   023
3                                                                                   024
```

FIGURE 3. SERIES HEADER DEFINITION RECORD (CTD STANDARD SUBSET)

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

45  0    6P        (60X,                                                          001
4                  92(I5,Al,I5,Al,I5,Al,2X))                                      002
4                                                                                 003
4   PRES7PRD       SEA PRESSURE DB=10KPASCAL  I  5-94   0.1     0               004
4   FFFF7AAN       QUAL.FLAG PRESSURE         A  1                               005
4   TEMP7STD       SEA TEMPERATURE DEG.C      I  5-94   0.001   0               006
4   FFFF7AAN       QUAL.FLAG TEMPERATURE      A  1                               007
4   PSAL7PRD       PRACTICAL SALINITY         I  5-94   0.001   0               008
4   FFFF7AAN       QUAL.FLAG SALINITY         A  1                               009
4                                                                                 010
4                                                                                 011
4                                                                                 012
4                                                                                 013
4                                                                                 014
4                                                                                 015
4                                                                                 016
4                                                                                 017
4                                                                                 018
4                                                                                 019
4                                                                                 020
4                                                                                 021
4                                                                                 022
4                                                                                 023
4                                                                                  24
```

FIGURE 4. DATA CYCLE DEFINITION RECORD (CTD STANDARD SUBSET)

The final character of the parameter code is used to identify the sphere in which the measurement was made. Values include: A (Atmosphere), B (Air/Sea Interface), D (Hydrosphere), and E (Sea Bottom Interface).

The Definition Record also contains, for each parameter, its name and units, its mode (integer, alphanumeric, etc.), the length, the value entered when a data value is missing, and two factors to be applied to the data value. One factor is a multiplication factor and the other is an addition factor. They can be used to indicate the decimal place, as in the present example to convert units to SI or to apply a simple calibration to the parameter value. In the example in Figure 3, the value of sea temperature is in both cases multiplied by 0.001 and then 0 is added. A value on the tape of 11567 is therefore to be interpreted as 11.567 in the computer.

The first 80 characters of the Series Header Definition Record in Figure 3 contain the format for the user-formatted area of the series header. This format is identical to any FORTRAN format statement. In fact, it is read by the computer and used to read the data. The format provides for 38 occurrences of the 6 parameters described in the definition record.

Figure 4 is the Data Cycle Definition Record. It describes the pressure, temperature and salinity fields which will be carried in the data cycle records. It also specifies a quality flag for each field. The quality flag is identified by the parameter identifier code FFFF. The 7 means that the values of the code are standard and can be found in the GF3 coding tables. The method code AA specifies which GF3 code table is used. The sphere character N means sphere not applicable.

The format statement in record sequence 1 of the Data Cycle Definition Record provides for up to 92 occurrences of the 6 defined parameters.

For both the Series Header and Data Cycle Records, missing data values or unused data positions should be set to the value specified in character positions 46-48 of the Definition Records. This is in each case for CTD data -94, which means -9999.

The remainder of the tape consists of a data file for each cruise and the Tape Termination File.

Figure 5 is an example of a Series Header Record. The first 5 record sequences contain various administrative and fixed field information, such as date and time of station, ship name, and the latitude and longitude of the station.

The value of 10 in record sequence 5 indicates there are 10 data cycles in the user-formatted portion of the Series Header immediately following. These data cycles contain the calibration data at 10 levels. Note that several missing values have been replaced by the dummy value code -9999.

Figure 6 is a Data Cycle Record. This record contains 92 values of pressure, temperature and salinity as defined in the data cycle definition record. Not all the data cycles for the station are included in the one Data Cycle Record. This is indicated by the 77 in character positions 1 & 2 of the record. The first 7 means this is a data cycle record and the second 7 means the next record is also a data cycle record.

The final file on the tape is the Tape Termination File. The content of this file is fixed and is described in the GF3 technical specifications.

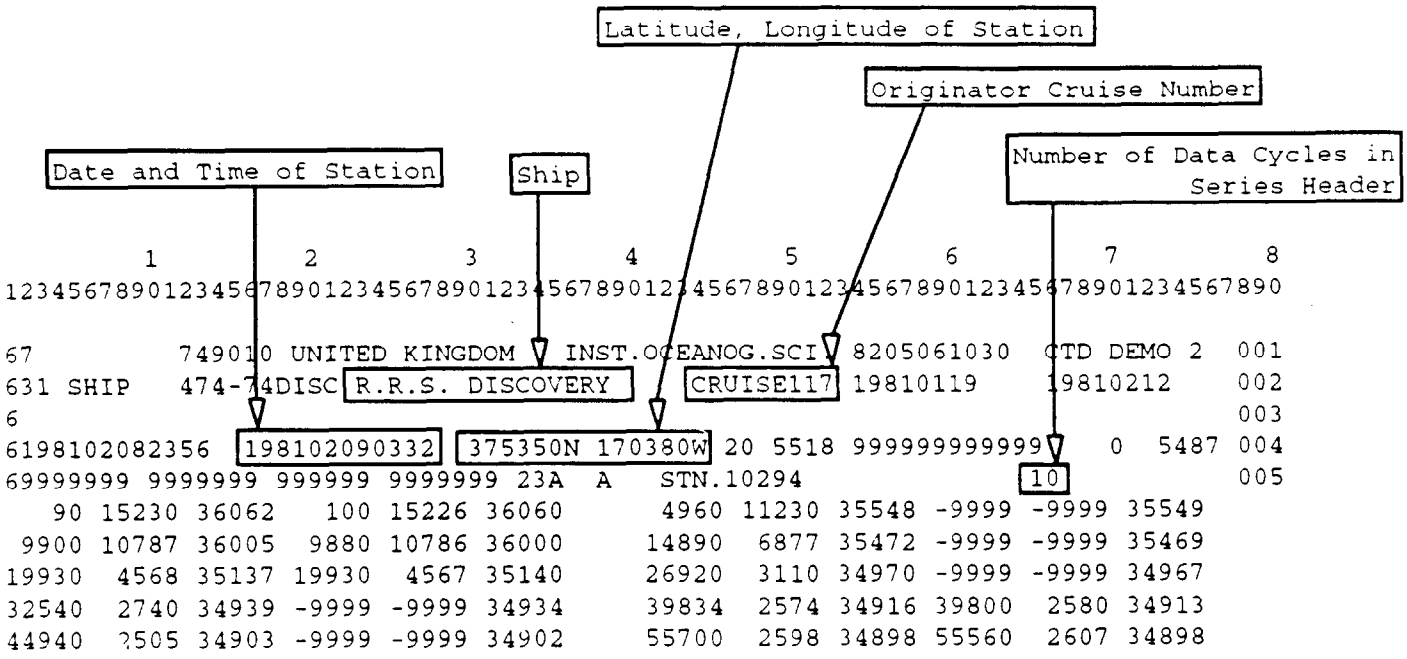

FIGURE 5. TAPE DUMP OF SERIES HEADER RECORD - GF3 STANDARD SUBSET FOR CTD DATA

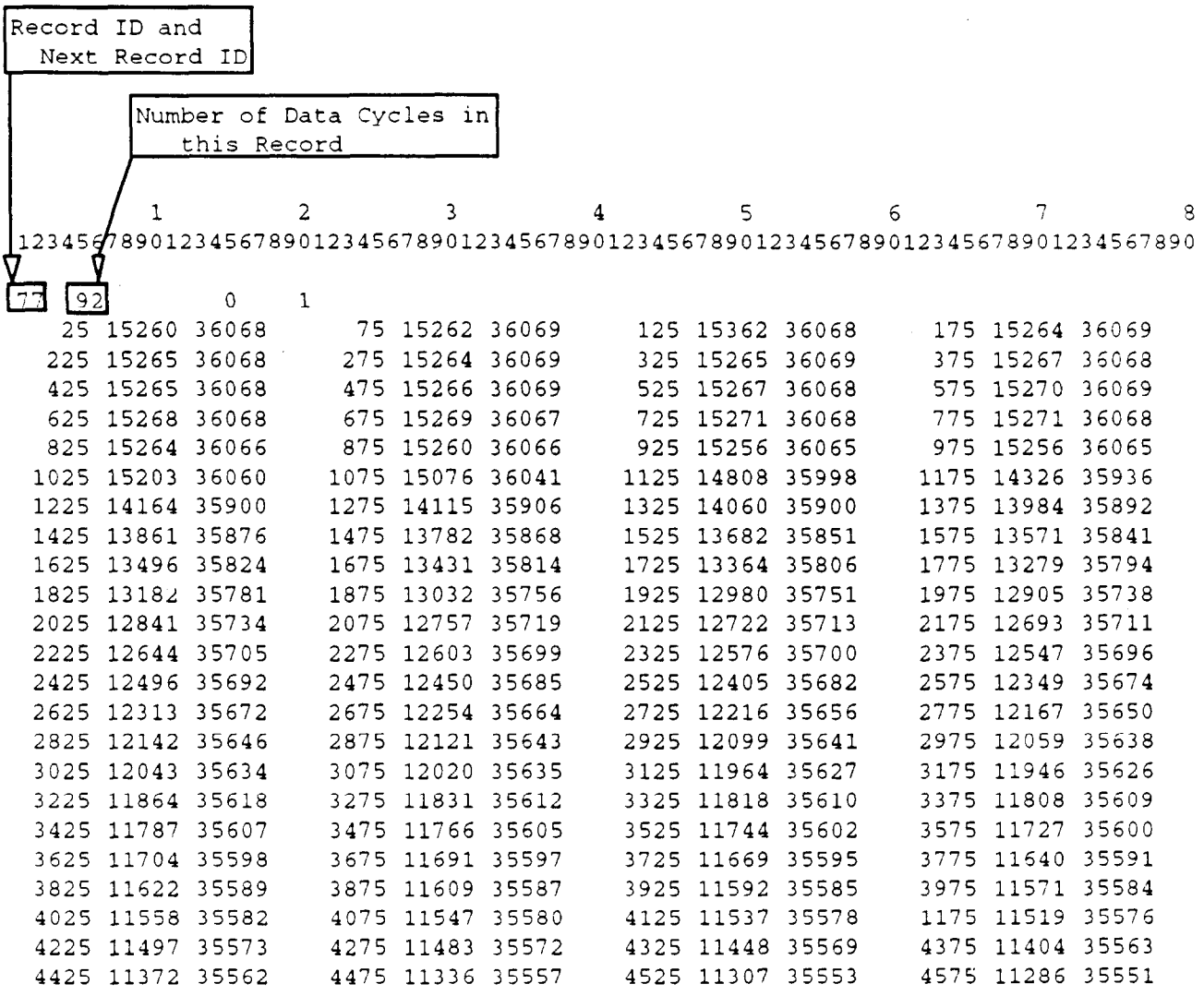

FIGURE 6. TAPE DUMP OF DATA CYCLE RECORD - GF3 STANDARD SUBSET FOR CTD DATA

3.2 SAMPLE STRUCTURE OF A TAPE CONTAINING WAVE SPECTRA COLLECTED AT SEVERAL OBSERVING STATIONS

The dataset consists of time series of wave spectra collected at several locations. The wave spectra collected at each location logically form a GF3 series. Each Data Cycle Record contains one wave spectrum and the date and time and various other pertinent information about the measurement, such as digital sampling frequency and bandwidth.

There is only one data file on the tape. It contains one series from each measurement location. Figure 7 shows the structure of this dataset.

Figure 8 is the Data Cycle Definition Record for the GF3 wave spectra subset. In this case, only the fixed format portion of the Series Header is used. All user supplied information is carried in the Data Cycle Records. Therefore, the tape does not include a Series Header Definition Record.

The parameter EEEE7XXN, which occurs several times in Figure 8, is used to replace the E type format familiar to many but which is not allowed in GF3. The parameter represents the power of ten, which is to be applied to the immediately following parameter. This permits the format to accommodate numbers of large range which could not be easily accommodated otherwise.

Note that the Data Cycle Definition Record specifies that there are 16 header parameters and 4 data cycle parameters in a Data Cycle Record. This is the fifth level of hierarchy noted earlier. The header parameters occur first in the data cycle and occur only once in a Data Cycle Record. The data cycle parameters can occur more than once in a Data Cycle Record. In the example given in Figure 8, the format statement specifies that 23 occurrences of 6 groups of the 4 data cycle parameters can be contained in a Data Cycle Record, along with the 16 header parameters. Therefore, a Data Cycle Record can contain up to 23 x 6 or 138 values of spectral density.

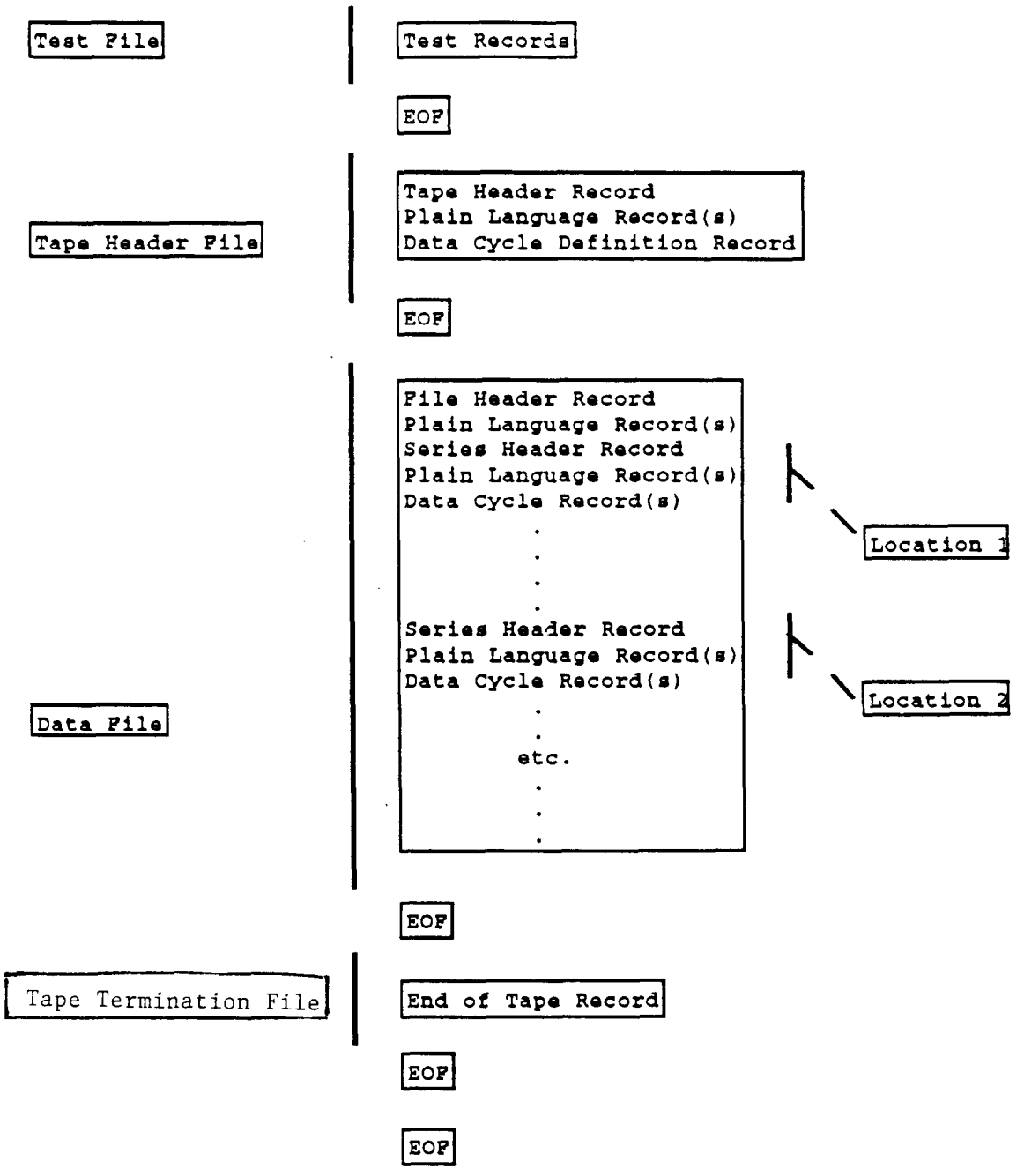

FIGURE 7. STRUCTURE OF A GF-3 TAPE FOR MEASURED WAVE SPECTRA SUBSET

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

45   16   4P        (3I4,I5,3I4,2A1,I1,2I4,2(I3,A1),12X,                          001
4                   23 (2X,6(I2,I4,I3,I4)))                                      002
4                                                                                003
4    YEAR7ZSN       YEAR (START OF WAVE RECORD)          I   4     1.0   0.0     004
4    DATE7ZSN       DATE (MMDD) GMT (START REC.)         I   4     1.0   0.0     005
4    HHMM7ZSN       TIME (HHMM) GMT (START REC.)         I   4     1.0   0.0     006
4    DRSC7PRN       DURATION OF RECORD (SEC)             I   5 95  1.0   0.0     007
4    FREQ7SSN       DIGITAL SAMPLING FREQ. (HZ)          I   4 94  0.01  0.0     008
4    EEEE7XXN       POWER OF TEN FOR BANDWIDTH           I   4 94  1.0   0.0     009
4    BAND7XXN       BANDWIDTH OF ANALYSIS (HZ)           I   4 94  1.0   0.0     010
4    FLAG2XXN       1USER DEFINED FLAG                   A   1                   011
4    FLAG2XXN       2USER DEFINED FLAG                   A   1                   012
4    CCCC7AAN       DATA CYCLE OVERFLOW INDIC.           I   1     1.0   0.0     013
4    VCAR7FAD       CHARACTERISTIC WAVE HT. (M)          I   4 94  0.01  0.0     014
4    VTPK7FAD       WAVE SPEC. PEAK PERIOD (SEC)         I   4 94  0.01  0.0     015
4    WSPD7XXA       WIND SPEED (M/SEC)                   I   3 93  0.1   0.0     016
4    FFFF7AAN       Q.C. FLAG FOR WIND SPEED             A   1                   017
4    WDIR7XXA       WIND DIRECTION (DEG. TRUE N.)        I   3 93  1.0   0.0     018
4    FFFF7AAN       Q.C. FLAG FOR WIND DIRECTIONA            1                   019
4    EEEE7XXN       POWER OF TEN FOR FREQUENCY           I   2     1.0   0.0     020
4    SPCF7XXN       FREQ. OF SPECTRAL EST. (HZ)          I   4     1.0   0.0     021
4    EEEE7XXN       POWER OF TEN FOR SPEC. DENS          I   3 93  1.0   0.0     022
4    VSDN7FAD       SPECTRAL DENSITY (M**2/HZ)           I   4 94  1.0   0.0     023
4                                                                                024
```

FIGURE 8. DATA CYCLE DEFINITION RECORD (MEASURED WAVE SPECTRA)

3.3 SAMPLE STRUCTURE OF A TAPE CONTAINING DRIFTING BUOY DATA

A drifting buoy is a buoy which is, as its name implies, free to drift in the ocean. The buoy drifts with the winds and currents and reports certain environmental parameters to ground stations via satellites. The satellites also deduce the position (latitude and longitude) of the buoy at the time of the report and add that to the message. A drifting buoy dataset, therefore, consists of a series of latitudes, longitudes, and environmental parameter values, each with its date and time.

It is customary to organize such a dataset chronologically by the buoy reporting the data. Figure 9 shows the organization of such a dataset in GF3.

In this case, the data from each buoy is a series on the GF3 tape. All the data are included in the Data Cycle Records. The Series Header is not required and does not appear on the tape.

Figure 10 is the Data Cycle Definition Record for the GF3 drifting buoy standard subset. There is one header parameter, the year. The data cycle parameters include latitude and longitude, since the position changes with every observation. Much numerical satellite data in GF3 would have a similar structure, with the position varying for each set of observed parameters.

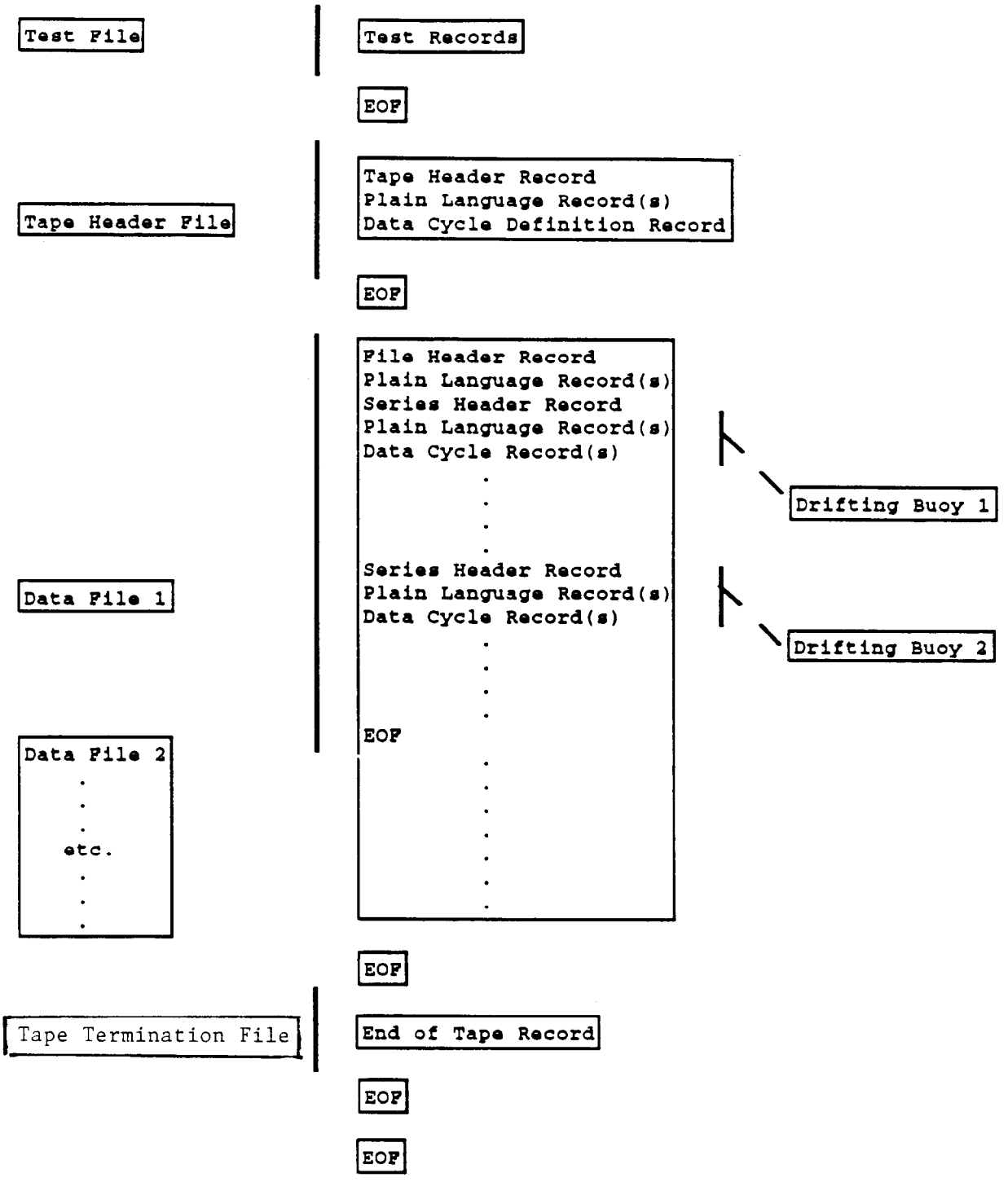

FIGURE 9. STRUCTURE OF A GF-3 TAPE FOR DRIFTING BUOY STANDARD SUBSET

```
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

45  1    9P       (2X,I4,54X,                                                    001
4                 46(2(I4,1X),2(I6,1X),Al,lX,I5,Al,lX,I4,Al,2X))                 002
4                                                                                003
4   YEAR7ZTN      YEAR                              I   4     1       0          004
4   DATE7ZTN      MONTH/DAY IN FORMAT MMDD          I   4     1       0          005
4   HHMM7ZTN      HOURS/MINS IN FORMAT HHMM         I   4     1       0          006
4   LATD7NSN      LATITUDE IN DEGS.NORTH+VE         I   6 96  0.001   0          007
4   LOND7NSN      LONGITUDE IN DEGS. 0-360EAST      I   6 96  0.001   0          008
4   FFFF6XXN      TYPE OF FIX (USER CODE)           A   1                        009
4   ATMS7XXA      SEA LEVEL PRESSURE MBARS.         I   5 95  0.1     0          010
4   FFFF7AAN      QUALITY FLAG FOR PRESSURE         A   1                        011
4   SSTP7PRD      SEA SURFACE TEMP. DEG.C.          I   4 94  0.1     0          012
4   FFFF7AAN      QUALITY FLAG FOR TEMP.            A   1                        013
4                                                                                014
4                                                                                015
4                                                                                016
4                                                                                017
4                                                                                018
4                                                                                019
4                                                                                020
4                                                                                021
4                                                                                022
4                                                                                023
4                                                                                024
```

FIGURE 10. DATA CYCLE DEFINITION RECORD (DRIFTING BUOY STANDARD SUBSET)

The examples presented above are, of course, hypothetical and present one method of structuring a given dataset within the framework of the GF3 formatting system. Because of the flexibility of GF3, any dataset can be structured in many ways, all equally valid and equally transparent to software systems incorporating the automatic processing features of GF3-Proc. It should be noted, however, that the order in which automatic processing systems extract the data from a tape will depend on the structure used.

It is also important to note the liberal occurrence of Plain Language Records and of quality flags associated with the standard subsets. Users of GF3 are strongly encouraged to take full advantage of these features to more fully document their data and provide information on the quality of the parameters, right down to the level of individual readings.

4. AN INTRODUCTION TO GF3-PROC

4.1 THE KEY FEATURES OF GF3-PROC

GF3-Proc is a suite of FORTRAN subroutines which provide the FORTRAN programmer with a simple and yet complete software interface for reading and writing data in the GF3 format. The package has been designed to exploit the full flexibility of GF3 and to relieve the user of much of the detailed coding that would otherwise be necessary to read or write a GF3 tape. A great deal of built in intelligence is contained within the package and it has been constructed to a high technical specification.

The GF3-Proc software includes an extensive level of error checking to ensure that tapes written using the package conform as closely as possible to the record sequencing and formatting rules of the GF3 system. These checks may also be applied to detect errors on an incoming GF3 tape before it is read and processed into the users own system.

One of the most important aspects of GF3-Proc is its ability to read and automatically analyze GF3 definition records and to use the knowledge gained to automatically control the reading and writing of data in the user defined areas of GF3. These areas of the GF3 Series Header and Data Cycle Records are the main data holding areas of GF3. GF3-Proc provides the user with a simple interface for reading and writing data in these areas without the need to be concerned with the mapping of the data into GF3 records.

GF3-Proc is designed to be portable across a wide range of different computer systems. This portability is intended not only to ensure the availability of the package to a wide user community, but also to provide the user with the insurance that his GF3 orientated software can be migrated to a new machine with minimum inconvenience. Where the user has a choice of different machines at his disposal, the portability enables him to select the machine most appropriate to his work or to spread his GF3 capability over a number of different machines.

Most of the design features of GF3-Proc are directed at maximizing programmer productivity. However, as it is anticipated that high data volumes will be passed through the package, particular attention was also paid in its design to making it efficient in terms of machine utilization. The highly active elements in the packages code have been designed to be as machine efficient as possible. The tape input/output of GF3 records is carried out by GF3-Proc through a single 1920 byte unformatted read/write instruction. The mapping of data between GF3 records and the users programme is handled in an internal GF3-Proc record buffer using specially tailored GF3-Proc routines without recourse to FORTRAN binary/character conversion statements.

GF3 was designed to be flexible enough to accommodate a wide range of different types of data and to enable all the necessary information to interpret and understand the contents of the tape to be included on the tape itself. Furthermore, the self-defining elements of the format were designed in such a way that they could be processed automatically. GF3-Proc provides this automatic processing capability. Although conceived originally as a data exchange format, the very design features of GF3 mean that it is also well suited for use as an archive format, particularly for multi-disciplinary data sets. The availability of GF3-Proc complements the use of GF3 in this role by providing a ready made user interface to the archived data. Furthermore, the interface can be migrated from machine to machine, together with the data archive.

4.2 PROGRAMMING ENVIRONMENT OF GF3-PROC

GF3-Proc consists of some 11,000 lines of FORTRAN code, of which about 50% is made up of comments. The code is sub-divided into approximately 165 subroutines and is written for use with FORTRAN 77 compilers on host machines with either ASCII or EBCDIC as their internal code.

The GF3-Proc subroutines form an interface between the users FORTRAN programme and the GF3 tape. Although the user programme has complete procedural control over the operation of GF3-Proc, all instructions that actually read from, or write to, the tape are carried out from within GF3-Proc itself, i.e., the user programme does not communicate directly with the GF3 tape. Only about 50 of GF3-Procs subroutines may be called directly from the users FORTRAN programme; these routines constitute the GF3-Proc User Interface. The remaining 100 or so routines operate from within GF3-Proc and are transparent to the user programme.

GF3-Proc makes extensive internal use of labelled common areas for communicating data and control information between its various subroutines. The communication of all data and control information between GF3-Proc and the users FORTRAN programme is carried out through arguments in the calls to the User Interface routines. For these subroutines there are a total of about 30 different arguments with which the user has to become familiar. On average, each subroutine has two arguments; one supplied by the user programme and the other returned to the user programme by GF3-Proc. The maximum number of arguments in any one routine is five.

Numeric data may be passed between GF3-Proc and the user programme, either in the form of a floating point variable or an integer variable, depending on which is more convenient for the user programme. GF3-Proc carries out any conversion that might be necessary. Thus, for example, if GF3-Proc retrieves a data field stored in a GF3 record as an integer with an implied decimal point but the user requires that field in floating point form, GF3-Proc will automatically set up the value as a floating point variable before returning it to the user programme. Character information is passed between GF3-Proc and the user programme through character variables.

The GF3-Proc software includes about 180 error traps. If any of these are triggered an appropriate message is generated in a standard format on the GF3-Proc error report file. All error messages are fully documented in the GF3-Proc Reference Manual where information is also provided on the likely cause of each error. If the error is the result of an inappropriate call from the user programme or the action requested is not acceptable to GF3-Proc, either because it is unrecognizable or because it is likely to corrupt further processing, then GF3-Proc will normally abort the user programme.

The core requirement of typical user programmes calling GF3-Proc to read, write or manipulate GF3 tapes is of the order of 25k words. The exact requirement depends on the users processing of the data before or after it is passed through GF3-Proc.

4.3 THE GF3-PROC USER INTERFACE

The subroutines of the GF3-Proc User Interface are designed to be closely related to the structure of the GF3 format. Just as the GF3 record is the central element of the GF3 format, the GF3-PROC record buffer is at the centre of GF3-Proc processing. The record buffer is an area in GF3-Procs labelled common which is designed to hold a single GF3 record. GF3-Proc processing is fundamentally concerned with reading data into the buffer, manipulating data within the buffer or writing out the buffer.

The User Interface routines may be classified into 8 separate categories:

(i) **Package control routines**: these are special purpose routines that enable the user to control the way in which GF3-Proc is to operate. There is a routine to initialize GF3-Proc processing while other routines enable the user to specify, for example, the FORTRAN logical unit number on which GF3-Proc is to output its error message file.

(ii) **I/O unit control routines**: these enable the user to specify the characteristics of the I/O units on which GF3-Proc is to read or write GF3 records, e.g., the FORTRAN logical unit number of the unit, the character code (i.e., ASCII or EBCDIC) in which GF3 records are stored on the unit, etc.

(iii) **File handling routines**: these enable the user to manipulate complete GF3 files. Routines are included to read (i.e., skip) or copy any number of files, to write an end of file mark, or to automatically generate a complete Test File or an end of tape file by a single user call.

(iv) **Record handling routines**: GF3-Proc processes one GF3 record at a time and the current record is held in the record buffer. Routines are provided to read the next record into the buffer from an input unit, to write out the record in the buffer to an output unit or to copy a record through the buffer (i.e., read it in from an input unit and then write it out to the output unit). A special routine is available to validate the contents of the entire record in the record buffer against the GF3 technical specification for its record type, checking, for example, that all fields are correctly formatted, that they contain plausible entries, that mandatory fields are present, etc. Another routine may be called to initialize the record buffer with a predefined skeleton appropriate to the GF3 record type being created by the user, e.g., pre-filling line sequence numbers and record ID fields.

(v) **Fixed field handling routines**: Once a GF3 record has been read into the record buffer, routines are available to extract specified fields out of the fixed format area of the record and into the user programme. Each fixed format field in GF3 is known to GF3-Proc by a unique identifier. The user simply supplies that identifier and GF3-Proc returns the value of the field to the user programme. Similarly, once a record has been initialized in the buffer, routines are available to enable the user to set up fields in the record by supplying the identifier of each field and its corresponding field value.

(vi) **Cycle handling routines**: These routines provide the interface to data held in the user defined area of Series Header or Data Cycle Records. Information on the formatting and content of the user defined areas is automatically picked up, analyzed and stored by GF3-Proc as the definition records pass through the record buffer. GF3-Proc closely monitors the various definition records and is able to retrieve automatically from its store the definition appropriate to the user defined area currently being read or written by the user programme. Data in these areas are manipulated by the user through a special cycle buffer maintained by GF3-Proc. At any given time the cycle buffer will contain the header parameters of the user defined area (referred to as the header cycle) or the current data cycle. The cycle handling routines enable the user to read the next cycle into the cycle buffer or to write out the cycle buffer to the GF3 output device. The mapping of the cycles to and from GF3 records is handled automatically by GF3-Proc and need not be considered by the user programme.

(vii) **Parameter handling routines**: Once a cycle has been read into the cycle buffer a routine is available to inform the user programme whether it is a header cycle or a data cycle. Further routines enable the values of specified parameters to be read out of the cycle and into the user programme. The parameters may be identified either by their GF3 parameter code or by the sequential position of the parameter in the appropriate Definition Record. Analogous routines are available to enable the user programme to set up parameter values in the cycle buffer. It should be noted that as parameter values are passed between the user programme and the cycle buffer, GF3-Proc automatically applies the scaling factors appropriate to the parameter (as defined in the Definition Record) and converts numeric values into the appropriate format, i.e., floating point or integer. If, in writing cycles, a parameter value is missing, or the user simply omits to pass a value to GF3-Proc, the software then automatically inserts the dummy value appropriate for the parameter.

(viii) **Special utility routines**: this is a small collection of miscellaneous routines providing utility functions that the user might find useful when preparing or reading data in GF3.

4.4 PROGRAMMING BENEFITS OF GF3-PROC

By providing a high level interface to GF3, GF3-Proc relieves the programmer of most of the GF3 orientated coding that would otherwise occupy much of his programme. In converting data between GF3 and the formats of the users own files, the bulk of the users GF3-Proc programme will normally be concerned with coding against the users own formats rather than against GF3. For example:

(i) GF3-Proc automatically handles the next record byte and the series header continuation flag, both of which require knowledge of what is to follow before they can be set.

(ii) GF3-Proc automatically handles the data cycle accounting fields in the Series Header and Data Cycle Records.

(iii) In handling user defined areas the user need only be concerned with cycles and parameters. GF3-Proc looks after their mapping, scaling and formatting into GF3 records and their overflow onto succeeding records, as required.

(iv) When data cycles overflow the user defined area of the Series Header Record, GF3-Proc automatically sets up the fixed format part (i.e., first 400 bytes) of the continuation Series Header Records.

(v) The user need not be concerned about coding for the detailed structure of GF3 records. Instead it is necessary to know only the individual field identifiers, the units of the fields and whether they are stored in numeric or character form. In the latter case awareness of field length is also required.

(vi) The user handles character information in the internal code of his FORTRAN compiler i.e., ASCII or EBCDIC. GF3-Proc automatically looks after its transliteration to or from the GF3 tape if conversion between ASCII and EBCDIC is required.

(vii) In creating GF3 records many fields can be initialized by a single call to the User Interface.

Although GF3-Proc is designed to handle most of the features of GF3 automatically, the option is usually available for the user to exercise control over these features if desired. In this sense, GF3-Proc is flexible to user requirements and allows the user as much control as needed.

GF3-Proc substantially reduces the code the programmer has to write and thereby minimizes the number of mistakes that might otherwise be made in writing a GF3 tape.

The level of checking built into GF3-Proc far exceeds that which would be cost-effective to write in one GF3 programme. This means that programmes using the package may be written quickly, but to a high specification.

Although magnetic tape is the normal medium on which GF3 data is stored, GF3-Proc also allows GF3 records to be read to, or written from, disk files. This provides a means of assembling GF3 files prior to their transfer to tape. It also enables programmes to be developed and tested interactively without the inevitable delays associated with tape mounting in a batch environment. Once the programme has reached operational status, input/output can be easily switched from disk to tape simply by modifying a couple of calls in the user programme.

GF3-Proc stores GF3 records on disk in a form particularly suited for manipulation by, for example, text editors. One or more GF3 records may be prepared and edited on disk before being read into GF3-Proc for incorporating into GF3 tapes. This technique is very useful for preparing Definition Records which can then be read through GF3-Proc in order to check for errors. The technique may be used to either read in complete records or to read in partially completed records which may then be completed by the users GF3-Proc programme before being written out. It is particularly helpful for preparing text information to be inserted into Plain Language Records. As far as the user programme is concerned, GF3 records can be read in from disk files in an identical fashion to their input from tape and GF3 records can be merged from
different input streams.

GF3-Proc allows GF3 records to be output to a printer as an alternative to tape output during the development of programmes. Once development is complete it is a simple matter to switch the output to tape.

4.5 GF3-PROC PORTABILITY

The Level 4 version of GF3-Proc is a FORTRAN 77 implementation designed for an environment where GF3 tapes are coded in either ASCII or EBCDIC. The host computer must have either ASCII or EBCDIC for its internal code, 6 significant figure floating point precision and at least 32 bits assigned to variables declared as INTEGER. Within these constraints it is estimated that about 99% of the GF3-Proc code is fully portable and that only about 1% of the code need be adapted to the particular characteristics of the machine on which it is installed. Such characteristics include tape input/output specifics and programme abort and traceback facilities. Those items of the GF3-Proc code that are dependent on such characteristics have been clearly isolated within the package design so as to facilitate their modification. Experience on installing the package on a range of machines of IBM, Honeywell, GEC, Norsk data, NEC, Univac, Sun, Data General, DEC and CDC manufacture has shown that only a couple of man days of effort are required to adapt the package for each new system. It should be noted that the modifications are carried out within the internal code of GF3-Proc and do not affect the User Interface to GF3-Proc which is designed to be totally portable.

5. EXAMPLES OF THE USE OF GF3-PROC

5.1 INTRODUCTION

The best way to both illustrate the capabilities of GF3-Proc and explain how to use the package is through example programmes. The programmes (or, where more convenient subroutines), selected are as follows.

EBCASC: This is a programme which copies the first data series from an EBCDIC GF3 tape onto an ASCII disk file and produces a lineprinter listing of the series. The programme is set up to apply the checks built into GF3-Proc to the input tape. It is assumed that the programme is to be run on an ASCII computer. If it is run on other types of computer, the disk file produced will be in the internal character code of that computer, not ASCII.

CMDGF3: This programme converts a simple current meter data format into GF3. Header records are set up using special GF3-Proc calls and the definition record is read from a disk file.

5.2 COPYING A DATA SERIES FROM TAPE TO DISK

The function of this programme is to copy the first series of data on an EBCDIC GF3 tape to an ASCII disk file and then list that file to a lineprinter. As a secondary function, the programme reports any errors it encounters on the source tape. The tape physical structure is defined as:

Test Records	
<EOF>	
Tape Header Record	
Plain Language Records	
<EOF>	
File Header Record	
Plain Language Records	
Series Header Record	Start of first data series
Plain Language Records	
Data Cycle Definition Record	
Data Cycle Records	End of first data series
Further data series	
<EOF>	
End of Tape Record	
<EOF>	
<EOF>	

It is assumed for the purposes of this example that this structure is rigidly followed and that the tape is written at a density of 6250 bpi.

The programme can be subdivided into a series of tasks as follows:

1) Initialization of GF3-Proc
2) Definition of the input tape
3) Definition of the output disk
4) Positioning of the input tape at the start of the first series
5) Copying of the series to the disk file
6) Repositioning of the disk file
7) Definition of the output print unit
8) Copying of the disk file to the printer

These tasks are shown in the programme listing which follows. Within the comments of the programme are included pointers to notes. Explanations of each of these notes follows the programme listing.

```
C
C    ***************************************************************************************************
C
C    Title - Programme EBCASC
C
C    Author - BODC/UK
C
C    Programme copies the first data series on a GF3 tape coded in EBCDIC onto disk and produces a lineprinter
C    listing of the data. The disk file is coded in the internal character code of the host computer.
C
C    ***************************************************************************************************
C
C    Logical unit numbers
C    LGT -      Input tape
C    LGD -      Output disk
C    LGPR -     Lineprinter
C
          DATA LGT,LGD,LGPR /10,11,6/
C
C    Tape density is ITDEN
C
          ITDEN = 6250
C
C    Initialize GF3-Proc
C** NOTE 1
C
          CALL GFPROC
          CALL GFPCST (7,2)
C
C    Define the input tape
C** NOTE 2
C
          CALL GFUNCR (ITKEY)
          CALL GFPCST (5,ITKEY)
          CALL GFUNST (1,1)
          CALL GFUNST (2,2)
          CALL GFUNST (6,1)
          CALL GFUNST (7,LGT)
          CALL GFUNST (8,ITDEN)
          CALL GFUNST (9,2)
C
C    Tell GF3-Proc that the tape is to be the current input unit
C
          CALL GFPCST (3,ITKEY)
C
C    Define the output disk
C** NOTE 3
C
          CALL GFUNCR (IDKEY)
          CALL GFPCST (5,IDKEY)
          CALL GFUNST (1,2)
          CALL GFUNST (7,LGD)
C
C    Tell GF3-Proc that the disk is to be the current output unit
C
          CALL GFPCST (4,IDKEY)
C
C    Read tape until Series Header Record is encountered.
C    This will not be until the third file of the tape.
C** NOTE 4
```

```
C
          CALL GFFLRD (2)
100       CALL GFRCRD (1)
          CALL GFRTGT (IRTY)
          IF (IRTY.NE.6) GOTO 100
C
C    Output the Series Header Record to the disk
C
          CALL GFRCWT
C
C    Copy the series to disk
C** NOTE 5
C
110       CALL GFRCRD (1)
          CALL GFRTGT (IRTY)
          IF (IRTY.EQ.6.0R.IRTY.GE.9) GOTO 120
          CALL GFRCWT
          GOTO 110
C
C    Terminate series on disk
C
120       CALL GFEFWT
          CALL GFEFWT
C
C    Rewind the disk file
C    Note that this redefines the disk as the current input unit.
C** NOTE 6
C
          CALL GFUNRW (IDKEY)

C    We have now finished with the tape so it can be released
C
          CALL GFUNRL (ITKEY)
C
C    Create the lineprinter unit and define it as the current output unit
C** NOTE 7
C
          CALL GFUNCR (IPKEY)
          CALL GFUNST (1,2)
          CALL GFUNST (6,3)
          CALL GFUNST (7,6)
          CALL GFUNST (11,3)
          CALL GFPCST (4,IPKEY)
C
C    Copy the disk file to the printer
C** NOTE 8
C
          CALL GFFLCP (1)
```

PROGRAMME NOTES

NOTE 1: Initialization of GF3-Proc

This is the first task of any programme using the package and is easily accomplished by a call to routine GFPROC. Note that this routine is called before any other calls to the package and is only called once.

As reports of any errors on the input tape are required, it is advisable to tell the package not to abort if it encounters any non-fatal data errors (errors in the data which do not prevent GF3-Proc from working). This is accomplished by a call to GFPCST with arguments (7,2).

NOTE 2: Definition of the input tape

The input tape is a unit containing GF3 data to be manipulated by the package and must therefore be made known to the package by a call to GFUNCR. Next we need to tell GF3-Proc the properties of this unit with the following series of calls.

```
CALL GFUNCR (ITKEY)        Set up the unit and store its key in the local variable ITKEY.
CALL GFPCST (5,ITKEY)      Define the unit as the current unit i.e., the unit affected by subsequent
                           calls to GFUNST.
CALL GFUNST (1,1)          Unit is an input unit.
CALL GFUNST (2,2)          Turn automatic processing on (tells GF3-Proc to check the input tape
                           as it is read).
CALL GFUNST (6,1)          Unit is a tape.
CALL GFUNST (7,LGT)        Unit is equated to FORTRAN logical unit LGT=10.
CALL GFUNST (8,ITDEN)      Tape written at ITDEN = 6250 bpi.
CALL GFUNST (9,2)          Tape coded in EBCDIC.
CALL GFPCST (3,ITKEY)      Set up the tape as the current input unit.
```

The first call to GFPCST is not strictly necessary as the current unit is set up automatically by the call to GFUNCR. However, it is included here for the sake of clarity.

NOTE 3: Definition of the output disk

This is set up in exactly the same manner as the input tape, only this time automatic processing is not selected and more of the values may be left as defaults. The sequence of calls is.

```
CALL GFUNCR (IDKEY)
CALL GFPCST (5,IDKEY)
CALL GFUNST (1,2)
CALL GFUNST (7,LGT)        Logical unit is LGT=11
CALL GFPCST (4,IDEY)
```

NOTE 4: Positioning of the input tape

The input tape must be positioned until the Series Header Record of the first data series has been read by GF3-Proc. This involves skipping two files (the Test File and the Tape Header File) on the input tape then reading records until a Series Header Record has been read. This can then be output to disk leaving the package ready to start copying the rest of the series. The following series of GF3-Proc calls may be used:

```
      CALL GFFLRD (2)          Skip over the first 2 files
100   CALL GFRCRD (1)          Read a GF3 record
      CALL GFRTGT (IRTY)       Ascertain the type of the record
      IF (IRTY.NE.6) GOTO 100  If the record is not a Series Header then read the next record.
      CALL GFRCWT              Write out the series header record.
```

NOTE 5: Copying the series to disk

GF3-Proc includes a record copying routine (GFRCCP). However, use of this routine requires foreknowledge of the number of records to be copied. In this case the number of records is not known so a simple loop must be used to read and write records until a Series Header or end-of-file (EOF) is encountered. The series can then be terminated by end-of-data (EOD = 2 EOFs).

Note that disk files created using GF3-Proc contain logical EOFs (records filled with 9s) rather than physical file marks. Thus it is quite possible to set up a GF3 file logically separated into files within a single physical disk file. If such a file is copied to tape using GF3-Proc each of the logical files will be written to the tape as a separate physical file.

```
110     CALL GFRCRD (1)          Read record from tape
        CALL GFRTGT (IRTY)       Determine type of record read
        IF (IRTY.EQ.6.0R.        Check for Series Header Record or EOF
        & IRTY.GE.9) GOTO 120
        CALL GFRCWT              Output record to disk
        GOTO 110
120     CALL GFEFWT              Entire series copied. Terminate with two EOFs.
        CALL GFEFWT
```

NOTE 6: Repositioning of the disk file

The disk file must now be rewound ready to be copied to the printer and then redefined as the current input unit. Both these tasks are accomplished by a single call to GFUNRW. The programme has now finished with the input tape so it can be released by a call to GFUNRL. GF3-Proc can support up to 5 GF3-Proc units open at any one time so the release of the tape unit is not strictly necessary before opening up the print unit. However, its release is an example of good programming practice.

NOTE 7: Definition of the print unit

The print unit is to be set up to give one GF3 record (24 lines of text) per page. GF3-Proc automatically generates the carriage control characters to produce output to the required specification. The calls required to set up the print unit and define it as the current output unit are:

```
CALL GFUNCR (IPKEY)
CALL GFUNST (1,2)        Unit is an output unit.
CALL GFUNST (6,3)        Unit is a print unit.
CALL GFUNST (7,6)        Map unit to FORTRAN logical unit 6.
CALL GFUNST (11,3)       Specify a page eject after each record.
CALL GFPCST (4,IPKEY)
```

NOTE 8: Copying the disk file to the printer

In this case the copying of a complete file is required which can be achieved by a single call to GFFLCP.

5.3 CREATING A CURRENT METER DATA SET USING GF3-PROC

This simple example illustrates how current meter data from a single meter can be placed on a tape. As in the last example, the programme is presented first, and various pointers to notes are placed in the comments as reference to explanatory material which follows.

The programme can be broken down into a number of logical stages. Some of these are serviced by a single GF3-Proc call. In these cases, the call is given in brackets and no further description is given.

1) Initialize the package (CALL GFPROC)
2) Set up GF3-Proc units
3) Output the test file (CALL GFXFWT)
4) Output the tape header file
5) Output the file header record for the data file
6) Output the data series
7) Output the tape terminator file (CALL GFZFWT)

```
C
C    ***************************************************************************************************
C
C    Title - Programme CMDGF3
C
C    Author - BODC/UK
C
C    This programme prepares a GF-3 tape of current meter data from a simple character format. Header records
C    are set up from a structured input stream. The definition record is read as is from a disk file. All data series
C    are included in a single data file.
C
C    ***************************************************************************************************
C
          LOGICAL LERR
          DATA LGTAPF,LGDISK,LGTN,LGDATA/10,11,5,12/
C
C    Initialize the package
C
          CALL GFPROC
C
C    Set up a GF3-Proc unit corresponding to the tape to be written. The tape is assumed to be 1600 bpi ASCII
C    running on an ASCII GF3-Proc installation and that the tape has been assigned to FORTRAN logical unit
C    10.
C** NOTE 1
C
          CALL GFUNCR (ITKEY)
          CALL GFUNSR (1,2)
          CALL GFUNST (2,2)
          CALL GFUNST (6,1)
          CALL GFUNST (7,LGTAPE)
          CALL GFPCST (4,ITKEY)

C    Set up a GF3-Proc unit corresponding to the disk file which contains the Data Cycle Definition Record which
C    is to be included in the output tape.
C
          CALL GFUNCR (IDKEY)
          CALL GFUNST (1,1)
          CALL GFUNST (7,LGDISK)
          CALL GFPCST (3,IDKEY)
C
C    Output the Test File
C
          CALL GFXFWT
C
C    Set up the tape header together with any associated plain text from the C       structured input stream.
C** NOTE 2
C
          CALL HDRPRP (LGIN,1,LERR,IERR)
          CALL PLNTXT (LGIN,1,IERR)
C
C    Copy the definition record from disk to tape and output EOF
C
          CALL GFRCCP (1)
          CALL GFEFWT
C
C    Set up the file header and plain text
C** NOTE 3
C
          CALL HDRPRP (LGIN,5,LERR,IERR)
          CALL PLNTXT (LGIN,5,IERR)
```

```
C
C     Series copying loop
C** NOTE 4
C
          READ (LGDATA,1210) IYEAR
1210 FORMAT (I4)
          IF (IYEAR.EQ.9999) GOTO 150
          IYEARL = IYEAR
C
C     Set up the series header and output
C
          CALL HDRPRP (LGIN,6,LERR,IERR)
          CALL GFRCWT
C
C     Insert any plain text
C
          CALL GFRCIN (0,1)
          CALL PLNTXT (LGIN,0,IERR)
C
C     Open automatic cycle writing
C** NOTE 5
C
          CALL GFCWOP (7)
C
C     Cycle output loop
C     Read an input record
C** NOTE 6
C
110       READ (LGDATA,1220) IYEAR,IDATE,ITIME,SPD,DIR,TEMP
1220      FORMAT (2I4,I6,F6.2,F5.1,F6.3)
C
C     Test for end of data
C
          IF (IYEAR.EQ.9999) GOTO 140
C
C     If year has changed, force the start of a new GF-3 record
C
          IF (IYEAR.EQ.IYEARL) GOTO 120
          CALL GFCCFL
          IYEARL = IYEAR
C
C     Ascertain whether a header cycle needs to be written
C** NOTE 7
C
120       CALL GFCXGT (ICTY)
          IF (ICTY.NE.1) GOTO 130
C
C     Output header cycle
C** NOTE 8
C
          CALL GFCIPT (1,IYEAR)
          CALL GFCYWT
C
C     Set up date and time fields
C
130       CALL GFCIPT (2,IDATE)
          CALL GFCIPT (3,ITIME)
C
C     Convert speed to m/s
C
```

```
            IF (SPD.GE.0.0) SPD = SPD/100.0
C
C     Set speed direction and temperature unless set to dummy value code
C
            IF (SPD.GE.0.0) CALL GFCFPT (4,SPD)
            IF (DIR.GE.0.0) CALL GFCFPT (6,DIR)
            IF (TEMP.GT.-9.0) CALL GFCFPT (8,TEMP)
C
C     Write out the data cycle
C
            CALL GFCYWT
            GOTO 110
C
C     Series output complete
C     Close automatic cycle writing
C** NOTE 9
C
140         CALL GFCWCL
            GOTO 100
C
C     Data file complete
C     Terminate file and output trailer file
C
150         CALL GFEFWT
            CALL GFZFWT
            STOP
            END
```

PROGRAMME NOTES

NOTE 1: Set up the GF3-Proc units

Two GF3-Proc units are required, the output tape and an input disk file containing the Definition Record. These are set up by the familiar series of GF3-Proc calls as the current output unit and the current input unit respectively.

NOTE 2: Output the Tape Header File

The tape header file is to contain a Tape Header Record, possibly some Plain Language Records and the Data Cycle Definition Record. The Tape Header Record and the associated Plain Language Records are set up by calls to HDRPRP and PLNTXT. The Definition Record is copied over from a disk file by a call to GFRCCP. It is worth noting that the complete Tape Header Record could be set up as a file on disk and then simply copied over to the tape by a single call to GFFLCP.

The tape header file is terminated by a call to GFEFWT.

NOTE 3: Output the File Header Record

The File Header Record is output in exactly the same way as the Tape Header Record by a call to HDRPRP followed by a call to PLNTXT.

NOTE 4: Output the data series

The data series are generated by a couple of nested loops, the inner one executed once per data cycle and the outer one executed once per series. The source file (this term will be used from here on to describe the file containing the data to be formatted into GF3) is read and returns either a year or the value 9999. If the latter value is encountered, the end of the data has been reached so the programme branches out of the outer loop.

The Series Header Record is generated by calls to HDRPRP and PLNTXT. Note that in this case, there is no allocation for plain text in the Series Header Record so the text must be placed in Plain Language Records. Consequently, the calls to HDRPRP and PLNTXT are separated by a call to GFRCWT (which writes out the Series Header Record) and GFRCIN (which initializes the GF3-Proc buffer with a Plain Language Record skeleton).

NOTE 5:

The programme is now ready to start reformatting data cycles and it informs GF3-Proc of this by opening automatic cycle writing (call to GFCWOP).

NOTE 6:

At this point, the programme enters the inner loop. A record is read from the source file and the value of the year is compared with the value from the previous cycle or, in the case of the first cycle, the header record. If it has changed, a fresh GF3 Data Cycle Record must be started (remember that in the current meter subset, the year only occurs once per Data Cycle Record and therefore data from two different years may not share the same Data Cycle Record) which is accomplished by a single call to GFCCFL.

NOTE 7:

A call to GFCXGT tells the programme whether the next cycle required by GF3-Proc is a header cycle or a data cycle. If a header cycle is required (ICTY=1) then the year is inserted by a call to GFCIPT and the cycle written by a call to GFCYWT. Note that the programme has been designed in such a way that the header cycle and the first data cycle are written by a single pass through the inner loop.

NOTE 8:

Next a data cycle is written. Date and time are inserted into the data cycle by calls to GFCIPT. Speed (after conversion to m/s), direction, and temperature are inserted by calls to GFCFPT but only if they are not equal to their dummy value code and the cycle is written by a call to GFCYWT. This shows one of the useful features of GF3-Proc. Any data cycle fields which are not explicitly set when a call is issued to GFCYWT are filled with the dummy value code taken from the definition record (if no dummy value code is defined then an error results). The Data Cycle Definition Record has entered the programme as data, and therefore the formatting of the data cycles is data driven. Consequently, any changes you require (e.g,. changing the dummy value code for a channel) can be accomplished by editing data and not the programme. Further, if you require several formats to be converted into GF3 and use the same definition record file for each conversion programme then consistent GF3 data cycle layout is guaranteed.

NOTE 9:

The inner loop is continued until the end of series record is encountered. The only task remaining in this pass through the outer loop is to close automatic cycle writing (telling GF3-Proc that the series is complete) by a call to GFCWCL. When the last series has been processed, the data file is terminated by a call to GFEFWT.

6. AVAILABLE SUPPORT FOR GF3 AND GF3-PROC USERS

From 1981, the Service Hydrographique of ICES acts as the RNODC (Formats). As such, one of its responsibilities is to act as a referral centre for individuals and agencies wanting information about the GF3 formatting system, including a description of the format system, standard subsets and GF3-Proc. It acts as an archive centre for code tables for GF3. It manages the expansion of the existing GF3 parameter code tables as necessary under the guidance of the IOC Committee on IODE (through GETADE or its successor) and to provide a focal point to which user requirements for new parameter codes may be directed. It maintains user aids for GF3, including a programme library for processing of GF3, guidance notes and user guides, documentation of standard and experimental subsets of GF3, and sample data tapes of GF3 subsets. It works closely with the GETADE to ensure the provision of expert knowledge on and the promotion of GF3 as an exchange format.

The responsibilities related to GF3 are for the purpose of simplifying and expediting the implementation of GF3 in the international community.

In particular, the programme library function is seen as useful to those implementing GF3 for the first time. The following sections discuss briefly some of the software which is available through the RNODC (Formats). Up-to-date information on the contents of the library, and the availability of guidance notes, user guides and other user aids can be obtained by writing to RNODC (Formats).

6.1 GF3 TAPE INSPECTION UTILITIES

Two portable FORTRAN programmes are available to inspect a tape in the GF3 format: a tape summary programme and a tape listing programme. Data centres receiving a tape written in GF3 can, by running these programmes, make an immediate check on the amount and the structure of the data on the tape and can obtain all the information needed to plan the processing of it. The programmes are also very useful for checking newly written GF3 tapes before they are sent off.

The tape summary programme displays the order of the different GF3 record types on the tape and lists the full contents of the tape header and terminator files. If the tape contains too much data to be listed in full, this information can be used to choose portions of the data files to be listed using the selection options provided by the tape listing programme. In particular, header and definition records can be picked out. Both programmes always check the GF3 test file, giving a summary report.

6.2 GF3 INTERFACE PROGRAMMES

Users who develop useful programmes and those who develop GF3 standard subsets are encouraged to deposit these programmes and documentation in the RNODC (Formats). Information on the availability of such programmes can be obtained from the RNODC (Formats).

There is a variety of computer programmes presently under development or planned in several National Oceanographic Data Centers. These programmes or subroutines will be deposited in the RNODC (Formats).

7. FUTURE DEVELOPMENT OF GF3

It rests with the GETADE of the IOC Committee on IODE to recommend to the Committee any amendments to the system but will work more effectively if it has feedback from users of GF3 about problems encountered. The Group of Experts can be reached through the IOC Secretariat or the RNODC (Formats).