80/23

EISCAT
TECHNICAL
NOTE

EISCAT Data Gathering and Dissemination

by

Alan Farmer

KIRUNA
Sweden

EISCAT Data Gathering and Dissemination

by

Alan Farmer

EISCAT Technical Note 80/23
Printed in Sweden
EISCAT Scientific Association
Kiruna, October 1980
ISSN 0349-2710

E I S C A T DATA GATHERING AND DISSEMINATION

Alan Farmer, EISCAT

15 - Sept - 80

EISCAT Data Gathering and Dissemination

Contents

1. Introduction

EISCAT, the European Incoherent SCATter Scientific Organisation, was set up in 1975 to provide ground stations for use by the member countries, for studying the ionosphere by means of the incoherent scatter radar technique. The radar stations included are a VHF transmitter-receiver and a UHF transmitter-receiver at Tromso in Norway, and UHF receivers at Kiruna in Sweden and Sodankyla in Finland. Countries contributing to the scheme, besides the three Nordic ones mentioned are Great Britain, France and West Germany. Each country is, by EISCAT rules, to be allowed an appropriate amount of their own observation time on the antennae, and there will, in addition, be a series of so-called 'Common' experiments. These will be mutually agreed experiments, run at agreed times, and the results will be freely available to all the member countries.

The incoherent scatter technique is one that tends to collect a lot of raw data. This is particularly likely to be the case with EISCAT since it is especially intended for long-period survey work and is flexible enough to provide a lot of different types of information simultaneously. Some in-situ reduction and analysis of the data is carried out, but it is still envisaged that the amount of data leaving the sites will be collosal. Add to that the fact that much of it must be copied at least six times, and that different levels of reduction and pre-analysis may be requested, and the full size of the data gathering and dissemination problem can begin to be appreciated.

This paper attempts to give EISCAT's current philosophy on its data-gathering and dissemination system, and is also intended as an outline manual for use of that system.

2. Getting Data to the User

EISCAT produces large volumes of data, subsets of which must be distributed to the collaborating scientists in at least each of the six member countries. Each of the three ground stations, and the HQ separately, have NORD-10 computers. At the sites these are used to control the experiments and equipment, and as the first 'catchment' area for the data. It is, therefore, also logical that the computer should be the focal point of the system by which the data is handled. The EISCAT computers therefore have a multiple role in relation to the data collected:

- (i) They normally would control the experiments being run.
- (ii) They control the initial data capture to magnetic tape and, in the case of Sodanyla and Kiruna, the data transfer to Tromso when communications line capacity between the sites allows.
- (iii) They are responsible for any initial reduction and simplification of the data (taking 'data' here as, for instance, what comes out of the correlator - the correlator itself could also be said to have done a reduction).
- (iv) They are responsible for any initial analysis, compression, merging and extraction of data.
- (v) They will also be used for copying of the data and control of the database collected at all three sites.

This report will consider all these aspects in turn.

For a description of the NORD-10 computer system as configured for EISCAT see J. L. Armstrong "The EISCAT Computer System".

A special command processor - in many ways like an alternative operating system for the experimental control aspects of the NORDs jobs - has been written to front end the software of the site machines used for controlling and monitoring the equipment. This commmand processor runs under EROS, the EISCAT Real-Time Operating System. The WTAPE program described in Chapter 5 is normally controlled by this command processor. For details of EROS and the other experiment preparation and execution software see the "EISCAT Experimental Preparation Manual", J.L. Armstrong.

The data collection during normal experimental conditions will be done via the WTAPE program (see Chapter 5). This ensures the raw data tapes are generated in a specified format (see Chapter 3). This format and the WTAPE program have been generalized as much as possible to make WTAPE useful for a wide range of experiments run under EROS.

The system by which the data is disseminated is affected by many factors. Some of these are laid out below. In many cases decisions have not yet been taken, affecting the way EISCAT is to act in these areas, and so there are some uncertainties yet in how the final system will be structured. Where possible the current (Sept 80) thinking is described.

A. The way the data is collected and stored at the experimental level is more or less settled. Analysis of the requirements of the first Common Program and other experiments have shown that generally there is far too much data to be held on the small discs available to EISCAT's NORD-10s. For this reason initial data gathering will normally be to tape.

B. The amount of pre-processing expected on the data before it is sent out from EISCAT will have a great influence on the form it will be sent out in, the turnaround time to the user, etc. EISCAT is limited in this respect both in terms of equipment and personnel. Chapter 4 gives an outline of a possible data reduction and dissemination pathway. How much of that is performable by EISCAT, or expected of EISCAT, is still under active consideration. This report for now will only consider the parts that so far are clear - the generation and dissemination of 'RAW' and 'ARCHIVE' data, with reference to the rest only in passing. The standard laid out in chapter 3 will be generally applicable, though, if later stages of the processing chain are also to be included. Wherever there appeared to be a conflict we have concentrated on solving the problems at the earlier parts of this chain.

C. The compatibility problems - that is ensuring that the data is supplied to end-users in a format that is usable without too much effort on their part, but which does not take up too much of EISCAT's time and effort, have figured large in the considerations of the tape-production programs. EISCAT has settled on an ANSI-standard data tape transfer format. This is discussed in chapter 3, with further notes in chapter 5.

Test tapes have been sent to all member countries (mid June 80) with representative data, and in the final formats, to check that there are no foreseeable problems. We are still (sept 80) awaiting replies from some on this, but have heard of no insurmountable problems as yet.

D. The amount and depth of the 'after sales' service expected of EISCAT once data leaves the organization will also have a large influence on how much else can be achieved by the personnel involved. This also relates to point B. How often requests are made for repeats of all or parts of our archived data, or for slightly modified processed subsets of it will probably not be known until the full-scale system is in operation. To try to pre-empt some of the requests to give help with de-convolving the data formats, EISCAT have prepared a set of

tape data-extraction subroutines which any prospective users can be supplied with. These are at two levels depending on the machine the users have. See section 7. Using the subroutines supplied, extracting the data in a usable form should be a minor programming job involving only the assembly in various combinations of our basic 'building-block' routines. Each member country will , however, have to be responsible for supplying the 'key' machine dependent elements (see chapter 7) for their own computers if they do not have NORD-10s.
The backup EISCAT gives obviously depends on what levels of the scheme shown in chapter 4 they operate. The simplest scheme would be if EISCAT just collected the data, sorted it, archived it, copied it and then sent out the copies. What could be added to this, though, include various sort, merge, reduction and analysis jobs and this would probably also mean sending out different 'levels' of reduced data. Whichever way the final system is configured a data base of tapes stored and available will be necessary. A simple trial database is in operation at EISCAT HQ at present - the XTAPE program being the front-end to the data (see chapter 6) . Once some experience has been gained with this it will be developed to make it more comprehensive, give it better selection and reporting capabilities, and expand it in several areas. The present intention is that each site will hold their own version of this, keeping it updated on a daily basis with respect to their own tapes. At given intervals - say once a week - these individual databases will be copied over the communcations lines to the HQ machine where the central database of all EISCAT tapes will be kept.

E. We do not at this time know how much 'other' data EISCAT is likely to be asked to provide along with the radar data. There are allowances built into the WTAPE programs to insert 'Special' data during the main data gathering process - either as extra parameters of the parameter set or as special data blocks on the tape. When other instrumentation is introduced to the system, however, deficiencies may be found in this methodology. Other ground instrumentation, or information from other sources may also in some cases be required to be sent together with the tapes - information it may be difficult to digitize with the other data, or specifically required in non-digital form.

F. At the moment it is assumed that sending out the tapes from Kiruna by post will present no problems. The envisaged system will, of course, be revised if problems become apparent with the time scales in delivery of data this gives, or if data is fond to get lost in transit. (See Appendix A)

3. The EISCAT Tape Standards

It has been decided to build the EISCAT tape formatting standard around the ANSI X3.27 and British BS4732 Standards. See Appendix B for a condensation of these standards, with notes on how these have been applied to EISCAT. Appendix D gives a list of definitions applied both there and in the rest of this text. EISCAT will use a 'parochial' label format, with the letter 'E' in the 80th column of the volume label to denote this.

It is hoped the redundancy inherent in the system as envisaged at present will allow for any future enhancements. It has been possible, however, to make some simplifying decisions that reduce the complexity of the tape handling without, we hope, detracting too much from this flexibility.

NOTE that tapes from EISCAT will normally be 1600b.p.i., 9-track and with symbolic files and labels written in ASCII characters, and data written in NORD-10 integer (16-bit, 2 byte) or, occasionally double integer (32 bit, 4 byte) characters. At the moment we hope it will be unnecessary, but if unavoidable the NORD-10 Reals may also used in the data files. These are 48 bit, 6 byte reals, and their format is explained in Appendix E. Any other, special formats will be announced at the time of the experiment. (This is one envisaged use of the 'NEWS' file in the experimental header - see Chapter 5, section on the routine for writing Experimental Header to tape.)

Simplifying assumptions

The tapes' labels will be written to British (BS) and ANSI Standards (see below). Some simplifications have been possible - viz:

1) only "multi-file volume" type tapes will be produces - this means that all tapes will be of the type

VOL HDR* ---FILE A---- * EOF * HDR * ----FILE B ...ETC...* EOF **

Where VOL represents a set of volume labels, HDR a set of file header labels, EOF a set of file end-of-file labels and '* ' represents a tape mark.

For the definitions of 'record', 'block', 'file', 'label' 'end-of-file' and 'tape-mark' see Appendix D. The ANSI and British standards allow for different formats if the data is arranged so that a file goes over several volumes or if eov coincides with eof etc. For simplicity the programs generating EISCAT tapes will always be arranged so that the tape has the format of a "multi-file volume" type as shown above. That is, if the data is so long that it will spread over two or more tapes, each of those tapes will be rounded off and begun as if they were separate experiments. Only the user labels within the tapes will specify that, and how, they are to be linked together.

2) The volume label sets will always be of a fixed type, that is there will always be two (80byte - 80 ASCII character) labels which are, in terms of the British standard, first the 'required' volume label (starting "VOL1") and then a user-volume-label (starting "UVL1"). The full definitions of the contents of these is given below.

3) The header label set will similarly be simplified to be always of a fixed type, a two (80-char) label grouping of a 'required' label (starts "HDR1") and a user-header-label (starts "UHL1"). Again see below for details.

4) Similarly to 2 and 3, the eof label group will be a fixed, 2-label group of 'required' eof label (starts "EOF1") and a user-terminator-label (starts "UTL1"). See again below for details of the internal structure of these labels.

The Data Files' Structure

NOTE: All the structural features below are given in terms of 'words', where a 'word' refers to the NORD-10 word of 16 bits (2bytes).

Data files on the tapes will be, in terms of the BS/ANSI standards, in a 'parochial' format, such that the data will always be in fixed length blocks of 1024 16-bit words (2048 bytes) - the 'physical' block structure. The internal (logical record) structure of the data is not within the scope of the ANSI standard but will also be standardised as follows:

The data files are composed of 1024 word physical data blocks(PDBs), which contain the data and pointers for extracting the logical data information within.

The logical data record (LDR) is the primary unit in which data comes from the generating program (e g WTAPE), and the LDRs are packed, without separators, into the physical blocks' final 1022 words, the first two words of these physical blocks being used as follows:

1st word (2bytes)= block number (within file)

2nd word (2bytes) = pointer to beginning of next logical-data-record (LDR) within that (physical) block. Since the LDR can be bigger than 1022 words it may be that no LDR starts within the physical block, and in that case the 2nd word will be zero. Otherwise it equals the word no, in that block, where the next LDR starts. For LDRs of less than 1022 words length it is possible to have two LDRs start in one physical block - in which case the 2nd word of the physical block will point to the beginning of the 1st LDR to start in that block, and the start of the next must be calculated from that and the length of the first complete LDR in the block. (That length will be the first value in the LDR array.)

In any one experiment the LDRs are kept as far as possible consistent no matter what type of data is contained within them. Thus, for example, the WTAPE program can dump to tape three types of data - the correlator output, the communications from another site (itself the correlator output of that site), and the special data of, say, an experimenter's own instruments. All three of these data types will have an LDR made up as follows:

If the LDR is M words long
The first word will be an integer value equal to the length of the LDR (i e M)
The next part of the LDR will be a dump of parameter information 128 words long, containing, in initial versions at least, integer values (see Appendix C) of system parameters during an experiment.
The last word of the parameter set is its version number, and associated with each of these will be a given format - Appendix C is the format for version 1.
After the parameters , the remainder of the LDR (M-129 words) will be the data.

As an example, if we have an LDR which is the result of dumping a 2048word correlator output dump, and assuming a 128 word parameter set, then the LDR will be 2177words long:

Word 1 integer value 2177 (length of LDR)
Words 2 to 129= parameter set (=128 2-byte integers)
Bytes 129 to 2177= correlator data (probably 2048

Contents of EISCAT user-specific labels

In line with the ANSI and BS standards outlined in Appendix B, certain user-specific elements exist both in the compulsory labels and in (especially) the user-specific labels of our labelling system.

In the compulsory label starting 'VOL1':
characters 38-51 will be originating site:
'EISCAT-KIRUNA '
'EISCAT-TROMSO '
'EISCAT-SODANKY'
or 'EISCAT-HQ '
character 80 will be 'E' to show data format is parochial, as defined by E(iscat).

In the compulsory labels starting HDR1 and EOF1:
characters 9-21 will be 'EISCAT-K-DATA','EISCAT-T-DATA', 'EISCAT-S-DATA', or 'EISCAT-H-DATA' depending on the originating site.
Characters 61-73 (operating sys) will be ' SINTRAN III '

In the user-specific label starting UVL1:

characters 38-51 as VOL1
characters 5-10 as VOL1 (ie tape number)
characters 12-17 tape type (= 'RAWON ','EMPTY ','ARCHIV', 'DATON' etc)
characters 18-23 is date in format YYMMDD
characters 24-27 is '1600' (tape density - 1600bpi)
characters 28-31 is tape length (usually 1200 or 2400)

In the user-specific label starting UHL1:

characters 18-28 is time in format YYDDMMHHMMSS
characters 32-35 as HDR1 (file no)
characters 12-17 file type ('EXHDR ', 'WTFIL ' or 'DTST ')
characters 38-47 experimenter's name
characters 48-51 = '/EIS'
characters 52-72 experiment title for expt header label & data start label. File name if a write-file.

In the user-specific label starting UTL1:

characters 18-28 as UHL1
characters 32-35 as EOF1 (file no)
characters 12-17 label type ('HDREND', 'WTFIL ','DATEND')
characters 38-51 as UHL1
characters 52-72 as corresponding UHL1
characters 73-76 tape length used so far (integer, feet)

Examples of EISCAT Labels

Examples of the labels on a tape o/p from the WTAPE program (see above for full explanations of the fields etc.):

Compulsory volume label
```
'OL1   130                         EISCAT-KIRUNA
```

User volume label
```
JVL1   130 EMPTY 80040816001200    EISCAT-KIRUNA
```

Compulsory header label
```
HDR1    EISCAT-K-DATA      00010001000100 80113 99365 000000 SINTRAN III
   or,
HDR1    EISCAT-K-DATA      00010002000100 80113 99365 000000 SINTRAN III
   or,
HDR1    EISCAT-K-DATA      00010003000100 80113 99365 000000 SINTRAN III
```

User header label
```
JHL1       EXHDR 800422133638  0001  ALANTES   /EISTEST OF WTAP
   or,
JHL1       WTFIL 800422133641  0002  ALANTES   /EIS(ALAN)ANAL-REP:S
   or,
JHL1       DTST  800422133645  0003  ALANTES   /EISTEST OF WTAP
```

Compulsory end-of-file label
```
EOF1    EISCAT-K-DATA      00010001000100 80113 99365 000001 SINTRAN III
   or,
```

```
EOF1    EISCAT-K-DATA       00010002000100 80113 99365 000001 SINTRAN III
    or,
EOF1    EISCAT-K-DATA       00010003000100 80113 99365 000082 SINTRAN III


     User end-of-file label
UTL1       HDREND800422133640  0001  ALANTES   /EISTEST OF WTAP               0
    or,
UTL1       WTFIL 800422133645  0002  ALANTES   /EIS(ALAN)ANAL-REP:S           0
    or,
UTL1       DATEND800422134513  0003  ALANTES   /EISTEST OF WTAP              14
```

Programs and routines to maintain this standard

The standard is maintained in the following way:

A. Within EISCAT

Each tape, on entry to the 'system' is labelled by program:
TAPE-INIT <unit no>,<tape no>,<tape length>
which puts the 'VOL1' and 'UVL1' labels on the tape, then two EOF marks.

Data tapes are generated by the program WTAPE (see Chapter 5). This is itself based around a set of standardised subroutines such as the MOUNT-TAPE, EXPERIMENT-HEADER etc routines which are responsible for maintaining the file formatting standards.

Most of the subroutines will be internal to EISCAT, since users should never have to write to the data tapes. However some, like the MOUNT subroutine, can be included in the subroutine library released to participating scientists. See Chapter 7.

Each 'RAW' data tape is checked by running program HTEST on it. This confirms readability, counts the numbers of files and records etc. After this a tape is copied onto archive tape using program ARCHIVE-COPY, which does a one-for-one copy of all but the user volume label. This it leaves as it found it on the recipient tape, except for changing the 'type' field to 'ARCHIV'. RAW tapes are put permanently to store and never used again, unless an archive copy becomes corrupted.

If raw data is to be sent out it is the archive tapes which are copied. It may be that some (see Chapter 4 and above) tidying, sorting, merging and/or analysis will be carried out before copies are made. If so (it is not done at present) then tape labelling will still follow the scheme given above, with tape 'type' as given in Chapter 4.

Note that each step in the above chain from initialisation onward includes a label check at each step to ensure tapes are not overwritten or otherwise rendered suspect. In general, tapes will only ever be copied from, rarely if ever re-written, and the label checking will be on this basis. For this reason too the write ring will be removed from each tape immediately it is created.

As an example of the label checking, WTAPE will only accept tapes of type 'EMPTY' (see UVL1 label details). Only the HTEST and ARCHIVE-COPY programs will accept 'RAW' type tapes, and then only read-only. Only ARCHIVE-COPY can create a tape of type 'ARCHIV'. Any program other than 'ARCHIVE-COPY' dealing with 'ARCHIV' type tapes can only be read-only. And so on.

B. Outside EISCAT

The security of the data on the tapes once they leave EISCAT is, of course, the responsibility of the recipient. EISCAT will have generated a label structure which should ensure that properly written software on a recipient machine cannot corrupt the tapes. To be 'properly written' that software must use the labelling in the way EISCAT (following the ANSI and British Standards) intends. To help users EISCAT will supply a series of subroutines which, when implemented with the correct machine-dependent core (see Chapter 7) will make the label checking and the extraction of data into easily-readable arrays, extremely easy.

We also strongly recommend the following: never put the write ring back in the tapes; only ever read from them, never write, and append only if absolutely necessary. Tapes coming to recipient establishments should be copied as they come in, so that an 'archive' independent of EISCAT's is maintained. Follow the safeguards in Appendix A. If in doubt about anything contact the EISCAT programming staff. Try any software out first on a test tape . EISCAT can supply one on request.

Note: all EISCAT supplied routines are written in FORTRAN. We have tried to confine what has been written for external use to as simple a subset of the language as possible (equivalent to say FORTRAN IV). This has meant not using, for example, the NORD-10's character string handling, and its Real number structure. However we cannot guarantee that routines will run without some minimum of conversion effort at each external site, even once the 'core' routines have been implemented. Similarly there may be occasions when even the things we had intended to screen out 'slip' through . Please bear with us in such cases.

4. The EISCAT Data Gathering Chain and Dissemination Points

At present the data gathering chain is envisaged to be as shown graphically below. An attempt is made to show at which points in this data is most likely to be sent out. It must be stressed again that at present only the sections down as far as the archiving have been implemented. No definite decision has been taken as yet on how much further down this chain EISCAT is supposed to go. With two programmers and half a NORD-10 at HQ, plus the site contingents (normally taken up with preparations for experiments etc), what EISCAT can accomplish is limited. So from the 'archive' level down this scheme can be considered preliminary only.

This system has been tested to the extent of making tapes of 'special' data (simple ADC output from the receivers, the 'St.Santin' experiment), which has been copied, archived and used for simple analyses. Also test tapes have been generated which were copied and sent to EISCAT member countries' representatives. (This was with dummy data. We are considering following up with some 'representative' data.) This was to check compatibility of our tapes with designated recipients' systems in the member countries.

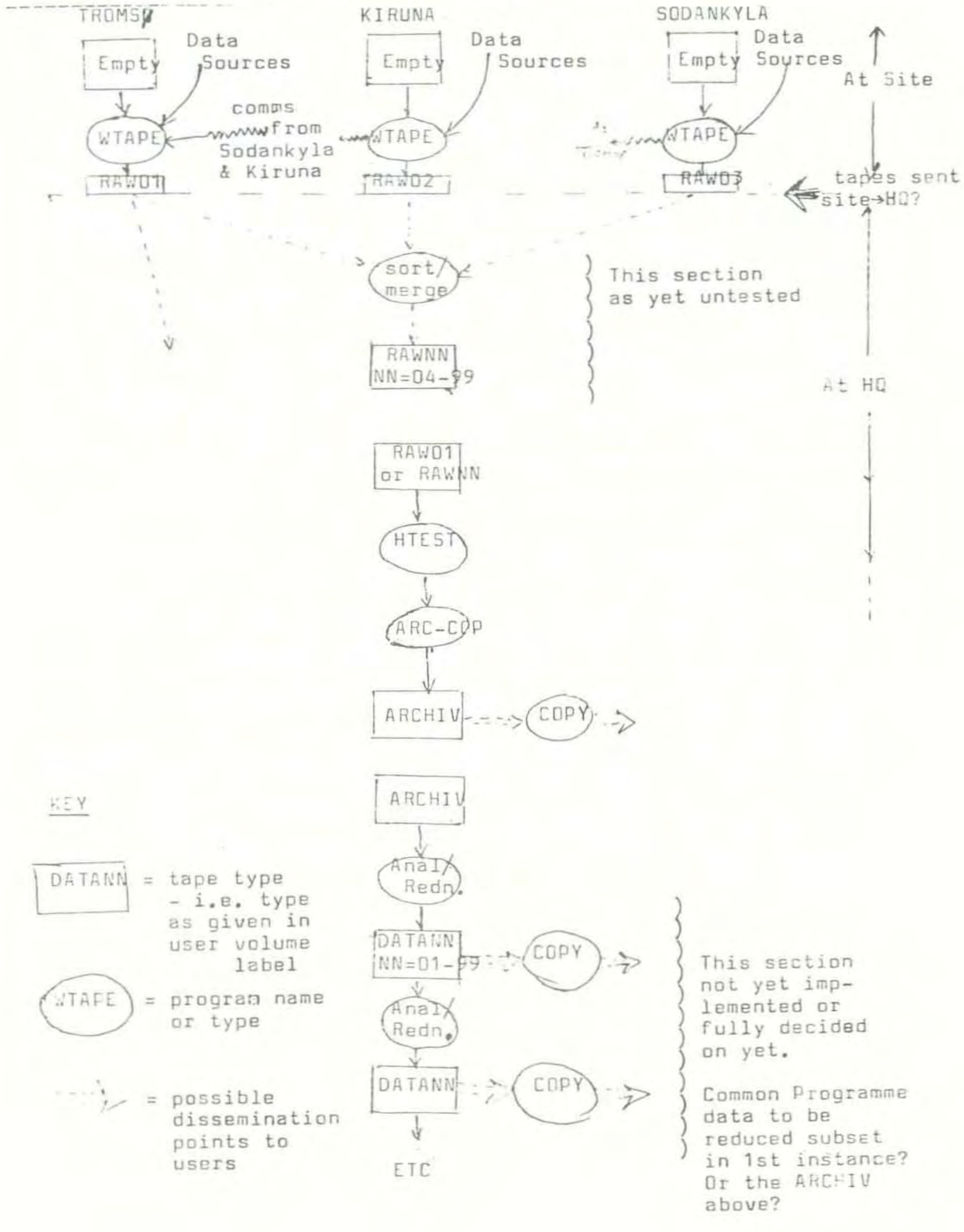
As yet undefined:

At what stage the work on the raw data is transferred from Tromso and the other sites to HQ (and how).

To what extent EISCAT is expected to carry on down this chain, or provide programs for the users to do so.

What sort of sort/merge will be needed when all three (or two of the three) sites are taking data simultaneously. We don't know how successful the data transfer along the communications line will be, and even when the data has been nominally interleaved at Tromso during the experimental run, it may need some post-sorting operation. It seems certain that there will be times when the data from Kiruna and Sodankyla cannot be sent along the communications line to Tromso during an experiment. Then tapes will probably have to be collected in one place after the experiment and sort/merged. So far (Sept 80) no attempt has been made to use the communications transfer during an experiment. This is to be tried at the next suitable opportunity.

The data flow, and dissemination points



5. WTAPE in Detail

A. The object of WTAPE

WTAPE is an RT (Real Time) program, run under EROS, which controls the writing away of data to tape during an experiment. For details of the NORD-10 systems used in EISCAT, see J.L.Armstrong, 'the EISCAT Computer System' and J.L.Armstrong, 'EISCAT Experiment Preparation Manual'.

WTAPE is written in a set of subroutines, the most important of which are:

- MOUNT - to mount a tape, checking its volume labels and that it is empty, and is the right number. Mounts the tape so that the write pointer is positioned after the 'VOL1' and 'UVL1' labels.
- EXHDR - to write the file signifying experiment start, which contains the symbolic file 'NEWS:SYMB', into which all messages to the user, news on the system, and special points to note, have been laid out.
- WTFL - controls writing of symbolic files to tape.
- DTST - sets up the start of data taking, writing away the header labels for the data file.
- CORRD, COMD and SPECD - initiated by DTST, these control the writing of respectively correlator DMA, communications and special data.

WTAPE is normally called via EROS, but can be invoked separately, in which case it is activated, in the same way EROS does it, by an 'RT WTAPE' command, and controlled by passing parameters via RT Common. (See the description of the NORD-10 system for explanations of these concepts). The way this is done is described below.

WTAPE has two main jobs:

- To check, maintain and supervise the tape labels
- To format and pack away the data, possibly from several sources simultaneously, to the tape in the manner specified in Chapter 4.

B. Communication with the WTAPE program

WTAPE is controlled via RT-COMMON, the common area which is accessible to all RT programs. Its different routines are started by setting appropriate flags there, then giving the start interrupt by:

RT WTAPE at system level

or CALL RT(WTAPE) from a program.

It communicates out to the external world in two ways:

- Condition codes, also in RT-COMMON, are set to signal successful or unsuccesful completion of a routine

- There is some output to the master control terminal, especially during error conditions, via the 'OUTSTR' routine. This 'chooses' the terminal it will output on by reading a device number from the RT-COMMON area /SITE/. Normally EROS takes care of setting this up, but if control is exercised without going through EROS the user must ensure this is set to a valid value.

The flags controlling WTAPE are in the RT-COMMON area /TPCONT/. There are two 'levels' on which WTAPE runs, marked by two different 'RTWT' commands in the program. The RTWT is a NORD-10 command that causes the program to pause at a given place until another interrupt is given. The interrupt needed to set it off again is the 'RT WTAPE' mentioned above. The 'top' level is that at which the tape is mounted, checked, has file labels written away and symbolic files copied to it. The other level - 'level two' - is entered once the labels have been written for the start of data gathering, and it is this level at which the writing away of data is controlled, i e it is here that the program is in 'data-gathering' mode.

The 'Top' Level and the Routines it Calls

At the 'top' level WTAPE is controlled by the flag ITSET. Any program wishing to activate it should set ITSET to the right value, then do a 'RT WTAPE' or 'CALL RT(WTAPE)' to kick it off from its RTWT. Valid values of ITSET are:

0 - No action
1 - mount-tape (MOUNT routine)
2 - write exptl header (EXHDR routine)
3 - write a file (WTFL)
4 - start data recording (DTST)
5 - stop data recording (DATEND)
6 - unload tape

As each routine is successfully executed another flag IMAG is updated to give a running indication of the last successful ITSET serviced. This enables checking to be carried out to ensure operations go only in the right order (otherwise error messages are generated). Thus we can set out a table of the allowable values for IMAG for each ITSET request value:

ITSET	IMAG must be
1	0 or 6
2	1,2,3 or 5
3	2,3 or 5
4	2,3 or 5
5	4
6	1,2,3 or 5

There is also a flag which calling RT programs can access to check on successful completion of routines. If an error condition occured it will tell of what type. This is the ICOND flag which is set to 1 whenever a routine is begun (and will thus be that if the routine aborts in some unexpected fashion) and to 0 when it finishes successfully. If it finishes in a controlled, but unsuccessful fashion, ICOND is set to a unique error code (see WTAPE listings).

MOUNT

The MOUNT routine needs to be passed the tape unit number and the tape number. This is done through /TPCONT/. It then opens the tape unit specified, rewinds the tape to start, reads the label, checks it is of the approved 'type', that it is an 'EMPTY' tape, that its number is that input, etc. If all is OK it will write 'RAW'' to the volume label and position the tape ready for writing after the 'VOL1' and 'UVL1' labels. See Chapter 3.

EXHDR

This needs to be passed Expt. No., Experimenter's Name and Expt. Name, again in the common area /TPCONT/. It writes a header file on to the tape, containing file (RT)NEWS:SYMB which the user should have prepared with any data, information or news which the potential tape recipient might want or need. This would be a symbolic file. This file represents the start of an experiment. There can be more than one experiment on a tape, each starting with such an experiment header file. Each experiment can contain any number of data or symbolic files (see below).

WTFL

This needs to be passed a file name in /TPCONT/. This file will be written away, together with the necessary labels, to tape. It cannot be written between a data start and end (i e it cannot interfere with data taking), and cannot be written away until an experiment-start header has been written to tape.

DTST

This writes the header label for the start of a data file, and sets the WTAPE program down into the level 2 described below - i e waiting for data interrupts at the second level RTWT. The WTAPE program itself supplies this routine with current file number etc. At this stage the calling program should also have specified the lengths of the data areas to be dumped to tape (in RT-COMMON in area /BUFCON/.)

DATEND

When data collection is interrupted (see 'level 2' below), the data collection loop is aborted and WTAPE returns to the 'top' level RTWT, and then goes to the DATA-END routine to write the end-of-file labels. Written away to the labels is the number of blocks in the file, the length of tape so far used etc. All these are generated within the WTAPE program as the data dumping routines on level 2 are executed.

UNLOAD

This will unload the tape and zero all tape-status flags, close the tape unit , etc.

'Level two' and the Routines it Calls

Once the WTAPE program has been set into the data-collection phase, it waits at an RTWT from which its branching is controlled by the flags IDMA, ISPEC and ICOM which are all also in /TPCONT/.

When called (by RT WTAPE or CALL RT(WTAPE)), WTAPE scans these flags in the order IDMA,ISPEC,ICOM, and calls appropriate routines depending on which it finds 'set' to 1.

IDMA=1 means a DMA dump (from the correlator) is required (calls routine CORRD).
ISPEC=1 means a special data dump is required (routine SPECD).
ISPEC>1 means a special data dump is required. This will be controlled by a routine written and inserted in WTAPE specially by a user or EISCAT staff.
ICOM=1 means a communications dump is required (routine COMD, only at Tromso).

All these routines work in roughly the same way. They expect to find their data on a fixed segment, and in a given area of that segemnt (with the possible exception of ISPEC>1). This segment is specified by common areas DMABUF (DMA data area), SPECBUF (Special data area) and COMBUF (Communications data area. On being started these routines take the data, length specified by LENDMA, respectively LENSPEC,LENCOM, format it in the format specified in Chapter 3 (DMA and Special dumps also automatically prefixing a parameter block by a call to a routine PARSET. The communications already has its own parameter block from the sending site.), then outputs it to tape in 1024word blocks - again according to the formats specified in Chapter 3. See Appendix C for details of the parameter set.

An internal common area /INTL/ holds the necessary counts and variables to keep running information on the amount of each physical block filled, starting position of next logical block within the current physical block etc.

A flag ICMOUT defines, at Kiruna and Sodankyla, if at the same time as a tape dump is carried out, a dump is also to be made to the communications line. This must be switched 'on' (=1) by the controlling program if data from Kiruna/Sodankyla is to be sent down the communications line to Tromso.

CORRD and SPECD also have an inherent check to see if any data has been lost since the last data dump, or is being lost during the current data transfer. Counts of the number of DMA dumps (NDMA) and Special data dumps (NSPEC) are kept. These must be initialised and incremented by the routines calling WTAPE. As an example of their use we can see what happens in the CORRD routine:

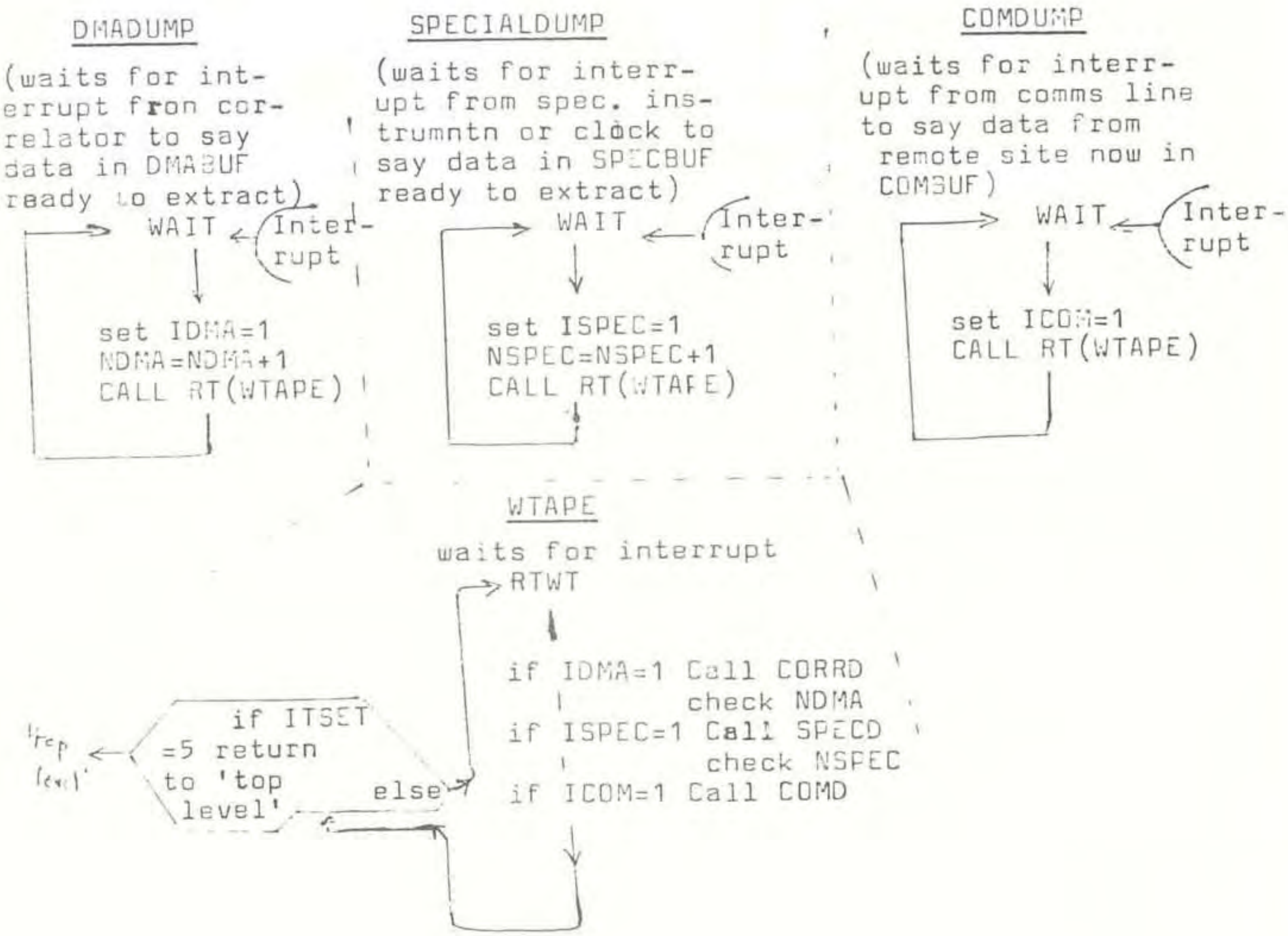
When called, CORRD copies NDMA to a variable IODMA. It then copies the data array from the special segment into its own buffer and adds the parameter set to the front. It then checks the current value of NDMA against IODMA to see if the calling routine has updated the data area while it was copying. If so (ie NDMA not equal to IODMA) it goes back and copies another buffer full. It will repeat this operation twenty times before it gives up. Once it has an uncorrupted buffer, however, it formats and writes the data away to tape (and comms line if

ICMOUT=1), then checks NDMA against ONUM, the value of NDMA at the last call to CORRD. If NDMA is not equal to ONUM+1, a warning is issued that some data dumps have been lost. It ends by setting ONUM equal to current NDMA.

The common area /TPSTAT/ beside holding flag IMAG (see above) also holds variables IFLFT, the number of feet of tape left - a warning is given when within 400feet of the end -, NBLKS, the number of blocks recorded since the last DTST, and IFLNO, the number of the current file.

To set the program to return to the 'top' level from 'level two', ITSET should be set to 5 (and RT(WTAPE) issued). All other changes of ITSET while in the data collection loop at level two lead to error warnings and are normally ignored.

Normally the program issuing the starts for the DMA dumps is a program called DMADUMP. Various Special data dumping programs have been run with this system. The name of the program requesting a communications dump to tape at Tromso has not yet been finalised. If we call the Special data dumping program SPECIALDUMP and the communications program COMDUMP, we can represent the level 2 system as:



Note: none of the calling programs have holds on semaphores or internal devices, and there is no allowance within them for them to wait until WTAPE has finished clearing the last buffer of data. They dump data as fast as they can, without regard to one another, or to whether WTAPE can handle data at that rate. Similarly WTAPE writes the data away as fast as it can and does not get 'hung' if it is too slow to deal with all the calls or if some other routine has not yet finished. If it is too slow this will just be shown during the checks on NDMA,NSPEC and NCOM, for which error messages will be issued. This was felt to be the fastest way to get all these routines working together, and give the cleanest, fastest, smallest routines possible. This methodology runs the risk of losing data when the system is running at its fastest, and multiple updating of data areas (leading to data corruption) is then possible. However if these small, fast routines could not deal with things at that speed, routines with checks and stops in would have even less chance. Provided some way is written in to show when corruption has taken place, it was felt the speed of these routines more than made up for their lack of 'tightness'. One set of problems they do get totally around is any problem with deadlock situations with holdups on semaphores and internal devices.

Current State of Testing

A routine DMADUMP exists and has been tested with the WTAPE programs satisfactorily. A number of different Special routines have also been run with WTAPE using data dumped through the SPECBUF area. The communications dump has not yet been tested (Sept 80) but is next on the list of test runs. After that combinations will be tested, and speed runs carried out using realistic combinations. We already know we can dump away 8k data from the DMABUF area easily every couple of seconds if it is the only thing running. (Results of test earlier in 1980).

6. EISCAT's Tapes Database

On user name TAPES-EISCAT on the HQ NORD-10 is a program XTAPE which controls access to the database of EISCAT tapes. This is used as a simple test of a tapes database for handling the tapes going through the EISCAT system.

At present the following fields are used:

- Tape number
- Name (10chars)
- Description (30chars)
- Date (YYMMDD)
- B.p.i.(800/1600)
- Length (<500,500,1200,2400)
- Free-to-use (Y/N)
- Comments (92 chars)

The comments field has proved to perhaps be the most useful so far!

The program is written in structured (top down) style. At the terminal:

```
@XTAPE
A(DD),D(ELETE),C(HANGE) OR R(EPORT):
```

Answer A - will give a routine that asks for the number of tapes to be input, then finds the next free numbers and asks for the tapes' details.

D - will give a routine that asks for the tape number then prints its details and asks for a confirmation before deleting it.

C - enables changes to be made to the individual fields of a given tape's information

Eg

```
A(DD),D(ELETE),C(HANGE) OR R(EPORT) : C

TAPE NO : 10

THIS IS TAPE .......
                 (goes on to print out details
                  of the tape)


OK (Y/N) : Y
FIELD NO. TO CHANGE : 6

OLD LENGTH 2400
NEW LENGTH : 1200


                 Etc
```

R - gives an option to run up to 10 different numbered reports (of which 2 written) which will give different sorts/selections of the tapes data

The eventual aim is that each site will keep its own data on its own version of the database, which HQ will copy over at regular intervals (like once a week), to keep a central directory. This will probably mean adding a 'site' field to the record definition.

Experience so far shows that the two reports currently written seem to be enough to keep sufficient track of the number of tapes we now have in the system (<200). However, as experimentation proper starts this will probably not be the case and other reports will have to be included to search on different parameters . The current reports give a listing by number for a given number range, ande a list of free-to-use tapes. We will probably eventually have to search on different parameters, and it may become necessary to code up information currently held in the comments field, to allow searching on that information.

Even with this reduced system there are many cases where a second date field would have been an advantage. It will probably be best in any later system to have at least two dates - probably a date-entered-in-the-system and a date for the last change made to that tape's details.

Before any tapes database is generally implemented 'in earnest', it is obvious that extensive rewriting of the current system will have to take place to take into account the above points. While this is happening it would probably also be best to replace the current fixed length comments field with codes for some most-used comments, and a variable length field, which will save some disc space, for the rest. A tape type field will also have to be added (and then reports can be run which select only archive or raw tapes etc.)

Suggestions as to how the database should be constructed, or ideas on what should be expected of it etc, would be gratefully received at EISCAT HQ before the major job of re-writing takes place.

7. Libraries of Routines Available From EISCAT

All EISCAT programs dealing with the processing of tapes that will be available to users, will be built up from a standard set of subroutines. At a 'bottom' level, these will call machine-dependent subroutines for doing things such as opening tape units, reading one block from tape and so on. Let us call this level ML.

Built onto the ML level routines will be intermediate level routines (IL), which apart from calling the ML routines will use only a very simple subset of Fortran - i e a standard of Fortran as machine-independent as possible.

Further up still in the order of complexity EISCAT will also have written full programs (FP), themselves built on the previous two levels of software.

EISCAT will make available to users on request the routines and programs on any and all three of these levels. The ML level routines, of course, are only of use to those who have NORD-10s, since their peripheral calls are machine-dependent, but the IL and FP software should be usable by all, once they have themselves generated a set of software that simulates what the ML level software does, on their own machine. Only a couple FP level programs are currently available, as shown below, and the availablility of more will depend on decisions yet to be taken on how much EISCAT HQ is expected to supply in this area.

For those without NORD-10s, a list of the ML routines should be obtained from EISCAT HQ, with their calling format and operating characteristics etc.

ML Routines

These are:

- FSPB (forwards-space-blocks)
- BSPB (backwards " ")
- FSPF (forwards-space-files)
- BSPF (backwards " ")
- OPENU (open tape unit)
- RDBL (read-block)
- WRBL (write-block)
- UNLD1 (unload tape)
- ENFL1 (write end-file)
- REWND1 (rewind to start of tape)
- (OUTSTR) (an output - to - terminal routine)

Detail definitions are given below.

IL Routines

These are (Sept 80):

SPF (forward/backwards space files)
SPB (" " " blocks)
MOUNT (mount a tape, with all necessary label checking)
GETLR (get a logical record)
GETPHYS (get a physical block)

Again, detail definitions are given below.

FP Programs

At present the programs available under this heading to external bodies are programs:

HTEST - to check an EISCAT tape's formats and count no of files, no of blocks etc (Also gives quick look at SYMB files and first blocks of data files)

ARCHIVE-COPY - a program to copy whole tapes, with all necessary label checking

A test program "LEARN-ABOUT-TAPE" can also be made available, together with a test tape, for anyone learning about the EISCAT tape labelling and formatting standard. This program illustrates the use of the ML and IL routines by example of reading a test tape. The test tape (which was the same as that sent out in June 80 by EISCAT to each member country, also has examples of NORD-10 integer, double integer and real numbers in its data blocks.

ML routines detailed definitions

FSPB

call as:

```
CALL FSPB(N,ITUNIT,IRET)
```

where:

N = no of blocks to forward-space
ITUNIT = mag-tape unit no.
IRET = error return code

Values of IRET:

0 - OK
1 - EOF encountered
2 - End-of-Tape encountered
3 - No more blocks on this tape
10 - Value of N passed is invalid
>10 - 10*(Machine-returned error code other than above)

Under the last category above, users wishing to make their FSPB routine on another type of machine emulate the NORD-10 ones exactly can get from EISCAT a complete list of the NORD-10 file system error codes, which are what our version of the routines pass back.

BSPB

Call as:
    CALL BSPB(N,ITUNIT,IRET)
where:
    parameters as FSPB, except N is the no of blocks to back-space, not forward-space.

Error return codes (IRET):
- 0 - OK
- 1 - Backspace to BOT (not yet implemented)
- 10 - value of N is invalid
- >10 - as FSPB

FSPF

Call as:
    CALL FSPF(N,ITUNIT,IRET)
where:
    parameters as in FSPB, N here being the no of files to skip forward.

Error codes:
- 0 - OK
- 1 - Unused
- 2 - End-of-Tape encountered
- 3 - No more blocks on this tape
- 10 - Value of N passed is invalid
- >10 - as FSPB

BSPF

Call as:
    CALL BSPF(N,ITUNIT,IRET)
where:
    parameters as FSPB, but N being no of files to backspace

Error codes:
- 0 - OK
- 1 - Backspace to Beginning of Tape (not yet implemented)
- 10 - Value of N passed is invalid
- >10 - as FSPB

OPENU

Call as:
    CALL OPENU(IDEV,ITUNIT,IN,IRET)
where:
    IDEV = Device no on which tape is to be mounted
    ITUNIT = tape unit no (internal) passed back to be used by the other routines
    IN = Read/Write flag (0=R,1=W)
    IRET = Error return code

On EISCAT NORD-10s IDEV can be 0,1 or 2 at HQ, has only one allowed value at Sodankyla and Kiruna sites, and two allowed values at Tromso.

Error codes (IRET):

- 0 - OK
- 999 - Invalid value of IDEV passed
- 998 - Invalid value of IN passed
- 997 - Tape not on-line
- 996 - Write requested but no Write-ring in tape
- 995 - Read only requested, but tape has a Write-ring in (it is up to individual users if they wish to consider this an error condition)
- 994 - Tape not at load point, and issuing a Rewind wont get it there.
- Else - internal error codes ofmachine (see under FSPB for comments)

RDBL

Call as:

```
CALL RDBL(ITUNIT,IARRAY,IWANT,IGET,IRET)
```

where:

- ITUNIT = tape unit no passed back by OPENU routine
- IARRAY = an integer array dimensioned to 1024 (max) or IWANT at least, into which read block will be returned
- IWANT = No of (16-bit NOTE) words to be returned, i e size of blocks, in words, to be read
- IGET = No of words actually returned
- IRET = error return code

Error return codes (IRET):

- 0 - OK
- <0 - IRET = (-IGET) if IGET not equal to IWANT
- 999 - trying to read blank tape at this point
- 998 - read overflow
- 997 - Tape unit not set to right density (which would normally be 1600bpi)
- else - equal to the particular machines error return code (see FSPB for comments)

WRBL

Call as:

```
CALL WRBL(ITUNIT,IARRAY,IWANT,IGET,IRET)
```

where:

Parameters as in RDBL except array IARRAY is to be written not read, and IGET is a dummy to make this and the RDBL routine consistent. IWANT becomes the no of (16bit) words to be written.

Error return codes:

- 0 - OK
- else - set equal to the machine's own return code on failure (see FSPB for comments on this).

UNLD1

Call as:
```
     CALL UNLD1(ITUNIT,IRET)
```
where:
ITUNIT = tape unit no passed back by OPENU
IRET = error return code (all values same as error return code from machine, except 0=OK. Users wishing to emulate EISCAT's NORD-10 routines exactly should contact us for details of the error codes.

REWND1

Call as:
```
     CALL REWND1(ITUNIT,IRET)
```
where:
parameters as UNLD1, ITUNIT now being tape unit on which rewind to be performed. IRET comments as UNLD1.

ENFL1

Call as:
```
     CALL ENFL1(ITUNIT,IRET)
```
where:
parameters as UNLD1

OUTSTR

Call as:
```
     CALL OUTSTR(CH)
```
where:
CH is a character string, to be output to the terminal

Some alternative to this will have to be supplied for any machine not having character array manipulation in FORTRAN.

IL routines in detail

SPF

Call as:
```
     CALL SPF(N,ITUNIT,IRET)
```
where:
parameters as in FSPF and BSPF - this is simply a routine that does both calls, a negative N being used for back-spacing. Error codes are those consistent with whichever of these operations were being attempted

SPB

```
  Call as:
      CALL SPB(N,ITUNIT,IRET)
  where:
      parameters as in FSPB and BSPB - this is simply a routine
                        that does both calls, a negative N being
                        used for back-spacing. Eror codes are those
                        consistent with whichever of these operations
                        were being attempted.
```

MOUNT

```
  Call as:
      CALL MOUNT(IDR,ITUNIT,ITNO,IRWA,ITY,ILAB,IRET)
  where:
      IDR = the tape drive no. (See OPENU above)
      ITUNIT = tape unit no (internal) passed back - see
                        OPENU above
      ITNO = tape no (will be checked against tape)
      IRWA = read/write/append flag
                        =0 for read
                        =1 for write
                        =2 for append
      ITY = tape type expected
                        ='AR' for 'ARCHIVE'
                        ='RA' for 'RAW'
                        ='EM' for 'EMPTY'
                        ='SC' for 'SCRATCH'
                        ='DA' for 'DATA'
                        ='BA' for 'BACKUP'
                        ='CO' for 'CO-US-FI'
      ILAB = (dimensioned ilab(3) ). Three two-character words
                        containing the tape type, to be written
                        over tape field of  VOL label if IRWA=1.
                        Otherwise a dummy.
      IRET = error return code
```

If 'append' is requested the tape will mount tape to end of all current files - ie to the double tape-mark following an EOF label group that signifies the end of tape's data. Whether IRWA=1 and 2 is allowed is checked against the tape type. 1 And 2 not allowed for tapes of type 'raw', while 1 (write) not allowed on 'archive' and 'data' tapes.

```
 Error codes returned (IRET):
      0 - OK
      1 to 999 - same as OPENU
      1000 - wrong tape no
                        (then the tape no found is passed back in ITUNIT
                         unless ITNO passed was 0, when ITUNIT passed
                         back is 9999, or if ITNO passed was -ve, when
                         ITUNIT passed back is 9998)
      1001 - wrong tape type, warning only
                        tape unit is opened, but user is warned the tape
                        type is wrong - happens if 'empty' or 'scratch'
                        tape mounted in place of 'raw', 'archiv' or'data',
                        or 'scratch' in place of 'archive' etc -
                        provided IRWA is 0.
      1002 - Wrong tape type, fatal
                        Tape unit closed and tape dismounted
      1003 - IRWA not equal to 0 but trying to mount a 'raw' tape
      1004 - ITY is invalid
      1005 - Tape incorrectly labelled
                        (ITUNIT passed back then contains the
                         error code from RDBL, or 500 if 'VOL1' is not the
                         1st characters of 1st block on tape, 501 if 'HDR1
                         is not the first characters of 3rd block on tape,
                         and 505 if unrecognized tape type..)
```

GETLR

```
  Call as:
      CALL GETLR(ITUNIT,IPAR,IDAT,LENDAT,IRET)
  where:
      ITUNIT is the tape unit no passed back by OPENU or MOUNT
      IPAR is a 128 (16bit) word array into which the parameter
                        array will be passed back
      IDAT is the data passed back - an array that should be
                        dimensioned at least as big as the data array
                        expected
      LENDAT is the size of the data array found, passed back
      IRET is the error code returned
```

Once positioned at a given place on the tape, this routine will read blocks until it finds one containing the start of a logical record (ie 2nd element of that physical block NE 0). If it has to read more than ten blocks to find this it abandons the search. Once it has found the start of a logical record it reads the record off, putting the parameter array from the start of it into IPAR, and the data into IDAT, with the length of the data found into LENDAT. It then resets the tape read pointer ready to read the next logical record. Any routine calling this needs also to contain a common area /INTPHYS/, defined as:

```
      COMMON /INTPHYS/ IATFL,IATBLK,IBLK(1024),ISIZE
```

Which will contain pointers to where the program is 'at' in the tape at present. These values of IATBLK and IATFL (ie the number of the block and the number of the file where read head currently located) must be initialised at the start of the program - see also GETPHYS.

```
 Error codes (IRET):
     0 - OK
   950 - have to read more than 10 blocks to find first logical
                         record
   951 - pointer to start of next logical record points to outside
                         the physical block
   952 - impossible record number (trying to read a 'symb' file
                         as data?)
   953 - no data found -just an unappended parameter block
   954 - impossible logical record length on tape
   955 - second element of physical block NE 0 when it should be a
                         spanned block
   956 - second element of physical block not correct when a logical
                         record should finish within this block
   957 - impossible value of pointer to start of next logical
                          record in this block - 0 or 1 or 2. Reading
                          a SYMB file as a data file?
   960 - (Old versions only) record length <1022 so program not able
                         to cope!!
  Else - error code returned from RDBL
```

GETPHYS

```
 Call as:
      CALL GETPHYS(ITUNIT,IFILE,IBLOCK,IBLK,IRET)
 where:
      ITUNIT = tape unit no passed back by OPENU or MOUNT
      IFILE = file no from which block to be extracted
      IBLOCK = block no in that file to be extracted
      IBLK = data block (dimensioned IBLK(1024) ) into
                      which read block is to be put
      IRET = error return code
```

The calling program must also have opened a COMMON area call INTPHYS defined as follows:

```
      COMMON /INTPHYS/ IATFIL,IATBLK,IBLOK(1024),ISIZE
```

Which will be used to keep track of current position on tape where program is 'pointing'. The pointers to which file and block the program is currently 'at' (IATFIL, IATBLK) should be initialised at the start of any sequence calling GETPHYS. Setting them to 0,0 at start after OPENU or MOUNT will suffice if the tape is to be read solely by GETPHYS. If, however, the user puts other calls to read from the tape into his program, using for example the RDBL routine, then care must also be taken to update IATBLK. Calls to GETLR automatically keep this common area updated.

NOTE: If GETPHYS is called, the read position on return will be to the block after the one specified in that call. Thus, if for example GETPHYS and GETLR are to be used together to extract data in such a way that GETPHYS will position the read roughly in the right area, then calls to GETLR made to get the logical records out, then if the block of the file specified in GETPHYS is to be included in the blocks searched for the next logical record, a BSPB must be done after the GETPHYS and the value of IATBLK reset (IATBLK=IATBLK-1) accordingly.

Error codes returned by GETPHYS

```
0 - OK
1 - unused
2 - IFILE>9999
3 - IBLOCK <1
4 - IBLOCK > no. Of blocks in file (ie EOF or EOT reached)
5 - IFILE > no of files on tape (EOT reached)
6 - IFILE < 1
7 - IBLOCK impossibly large (>14000 currently)
8 - Block read not 1024 words!!! Size returned in IBLOCK
10-999 - RDBL error code +10
1001-1999 - IRET from FSPF routine + 1000
2001-2999 - error return from BSPF routine + 2000
3001-3999 - IRET return from FSPB routine + 3000
4001-4999 - IRET return from BSPB routine + 4000
```

Programs currently available (FP)

HTEST

This program is loaded with the ML routines described above. When run, it requests a tape unit number, and a tape number, opens the relevant unit and checks the tape. It will abort if tape no is wrong or there is a write-ring in, or if the unit cant be opened for some reason. If all goes OK it then asks what level of analysis is required - two types are available, summary and detail. The summary analysis will read forward to the end of tape printing out the labels it finds, plus the number of blocks it finds in both SYMBOLIC and DATA files. If the detail output is requested the user is asked if the data files are to be output in octal, integer or not at all. If the last mentioned, then an output like the summary will be produced except that the first block (1024 words) of each symbolic file will also be output. The data files will only have their number of blocks counted, though. If octal output or integer output of the data is requested the first block of each data file will also be written out, in the appropriate format.

This program gives a good check that a tape's format is OK, and gives a good overall summary of what is on the tape for, for example, writing a program using the GETPHYS and GETLR routines to extract data. The dates and times at which the data in the file was taken is usually in the labels, which is obviously useful for locating wanted data.

ARCHIVE-COPY

This program does a simple tape copy of an EISCAT tape, checking in the meantime that the format is OK. If the tape being copied is a 'raw' tape, the copy will be given a label of type 'archive'. The program requests the in and out tape unit numbers, but everything else is automatic. The 'in' tape will be thrown off if it has a Write-ring in. If the 'in' tape ends prematurely (ie the program writing it did not give final EOF labels as it should) the copy program will 'finish off' the copy with properly formatted labels etc.

LEARN-ABOUT-TAPES

This is a test program for users trying to get used to EISCAT tape standards. It reads and prints out our June 80 test tape, and the program listing is commented heavily so that the user will understand what it is doing. Contact EISCAT HQ (Alan Farmer or Joe Armstrong) for details.

APPENDIX A

BS4783: The Care and Transportation of Magnetic Tapes

(Some notes from the standard)

Tapes used extensively for 7-track recording should not then be reused for 9-track and vice-versa.

(Eiscat intends to use only 9-track, though 7-track could be provided if specially requested.)

It is recommended that every tape to be used for information interchange be first tested over its complete usable area with an 'appropriate' test program (such as the TANDB program from CERN). This is not done at present, and no problems have as yet been encountered. It would also be fairly time-consuming considering the number of tapes EISCAT is expected to handle. However, this position is not considered closed, and we have the programs to do it if necessary.

The standard recommends that every tape, after writing, should then be wound forward until the end-of-tape marker is sensed before it is rewound. This is to reduce the possibility of 'tension discontinuity' which can cause cinching in transit. This is being considered as a possible 'tidy up' procedure at the end of the WTAPE programs, but is not as yet implemented. At present there are some uncertainties as to how well the tape machines used by EISCAT actually detect the end-of-tape marker, and further tests are required on this point.

The free end of tape round a volume should be held down by a synthetic rubber block or vinyl strip. This is, and will be, done on all tapes sent out from EISCAT. It should also be remembered by recipients of tapes from EISCAT if they wish to care for their tapes in the best possible way.

Handling care: Especial points to note include:

Care with the loose end of tape - damage to this can release oxide particles which may interfere with the recording/reading heads of the tape machine.

Putting on and taking off of tapes should involve pressure only on the hubs, never on the flanges since this can mechanically strain the wound tape.

The write ring should be taken out immediately a tape has been written. All operations on EISCAT tapes in our processing cycle will involve reading from one tape and writing to another - i.e. no overwriting anywhere in the system. This means that all applications programs can be (and are) written to detect and reject any tape from further up in the chain that is mounted thereafter with a write ring in. It is strongly recommended that this practice is also used by those receiving the tapes.

Self-adhesive labels should be affixed to all tapes, carefully, to the flange and so that they do not stick out from the flange surface. All necessary information to ensure the tape is identifiable should go on that label. Suggested in the standard are:

7 or 9 track
Packing Density
Recording Mode (NRZ or PE)
Character Code
Parity
Sender Name
Recipient Name
Volume Name and No of Volumes constituting the set
Interchange Identifier - agreed between interchanging parties.

EISCAT's tapes will have specially designed labels containing all but four of these parameters, plus a few not included in the list. The four left off, with reasons are:

1. Recording mode. The recording technique usually also defines this if 800 or 1600 b.p.i. It was considered having recording density permanently printed on the labels as 1600 (ie not have it as an optional field to be filled in) but we could not be sure that there were not users who would want 800. We are unable to supply 6400 b.p.i.
2. Character code. This was felt to be unnecessary since all recipients should have copies of our documentation where all necessary information on the tape formats is defined.
3. Volume number is included, but the number of volumes constituting the set is left off since this will only ever be one, as defined in chapter three.
4. Interchange identifier. This is also considered unnecessary in the light of the fact all potential users should already have a copy of our documentation and hence know the one format in which the tapes are sent out.

Writing on the label should be in indelible ink.

Tapes should be stored, before and after use, in their containers, which should be air and dust tight and sufficiently strong to afford some protection to the tape inside. These containers should also be labelled with the tape number/identifier, and care should be taken that they are not mixed up.

The standard recommends a transport case for tapes with an 80mm clearance around the whole of the tape(!!). This is to protect against stray magnetic fields in transit, although the authors of the standard frankly admit they do not really know how serious a problem this is. EISCAT has no plans to use such containers at present, but may reexamine this question if problems are encountered with losing data when tapes are sent out. At present they are sent in their normal plastic containers overwrapped with a protective layer of cardboard.

The standard recommends at least 6 hours deconditioning time, inside their boxes, for tapes taken out of, and subsequently returned to, the environment of the computer room. The bigger the climactic change endured, the longer should be the deconditioning time. This is most likely to be a problem within Scandinavia. In Kiruna, for instance, the difference between inside and outside temperatures can be as much as sixty Centigrade degrees.

APPENDIX B

British and ANSI Standards for Data Interchange by Magnetic Tape

BS 4732:1971 & ANSI X3.27-1978

(The applicability of the British standard BS 4732:1971 and the American standard ANSI X3.27-1978 to EISCAT tape formatting and labelling.

As far as possible we have tried to use the relevant ANSI and British Standards in the formatting and labelling standards of the EISCAT tapes. The relevant standard numbers are given above. The standards mentioned are essentially similar where they apply to what we intend to do , but both impose, in different ways, unacceptable limitations on tape formats that mean we are forced into a position where we cannot follow them exactly. These limitations come where the standards describe the format of the data itself (as opposed to the labels which we shall copy the standards in completely).

These limitations arise as follows:

A). The British standard, while allowing non-character formats in the data blocks (i.e. data can be in integer or other format if desired) specifically states that there should be an integral number of data records to a block (section 4.1, subsection (2)).

B). The ANSI standard allows spanned records (ie records and blocks do not have to bear a simple integral relationship and records can span over one or more physical blocks and start anywhere in one of those blocks) but requires specifically that all data shall be in character format - ie numbers shall , for example, be written out fully as their character representation.

The limitations described in a) and b) are unacceptable to EISCAT since, in the case of a) this severely limits our freedom of action in setting up experiments, imposing a rigidity which could impair the ways in which experiments are performed, and/or waste tape and machine time; and in the case of b) make the data much harder to analyse afterwards, waste enormous amounts of tape, and during an experiment waste valuable machine time while the number-character conversion was carried out.

It is therefore proposed that all EISCAT tapes should follow the ANSI/BS Standards in as many respects as possible but that the data shall be of a non-standard (or parochial, in their own nomenclature) type . It will, infact, be in binary number form, and with a spanned record format. The labels will follow the standards as closely as possible, and will show the 'parochiality' of the tape formats by having an 'E' in the 80th character position of the volume label.('VOL1' - see below.)

The notes following are intended as a condensation of the ANSI and BS Standards as they affect EISCAT tapes - anyone interested in the complete details, unsure what is meant in parts and/or needful of fuller details, should refer to the relevant standard(s).

BS 4732:1971 - "Magnetic Tape Labelling and File Structure for Data Interchange"

(Its applicability to EISCAT)

Each label shall be an 80 character block.

Classes of labels

Operating system labels

User labels
- also classified as 'required' or 'optional'

Types of labels

```
Volume labels - operating sys/required  ......VOL1 (& EOV1)
              - user/optional ................UVL1-9 (& UTL@)

File labels   - operating sys/required........HDR1, EOF1
              - user/optional.................UHL@, UTL@
```

(The label types in brackets do not concern EISCAT. There are also,in the standards, some defined operating sys/optional label types which we shall ignore here since they will never be needed by EISCAT.)

Note: above '1-9' in the label means that any of the single digit integers from 1 to 9 can be used here, while '@' means than any character 0-9 or A-Z can be used. These characters shall be in ASCII code. This definition is sufficient, but more limiting, than that of the BS4732.

Approved 'Macro' Structure for Files Recorded on Magnetic Tape

One approved macro structure for mag-tape files (in BS4732:1971 known as the 'multi-file' volume type) is the one chosen for EISCAT's formats. The tape-producing programs will be written in such a way that all tapes produced will be of this type (ie files do not span volumes, nor does insufficient room get left at the end of a tape to write the terminating labels.) This structure is as follows:

```
VOL HDR * ----FILE A ----* EOF * HDR * ----FILE B ----* ....ETC..* EOF**
```

Where * = tape-mark (end-of-file marker)
'VOL' is a volume header label group
'HDR' is a file header label group
'EOF'is an end of file label group

Every volume must have a volume header label (starts 'VOL1') as the first block on the volume and this label is not used anywhere else.

An optional user volume label may be appended immediately after the required volume header ('VOL1'),and will start with the characters UVL@ (in the case of EISCAT UVL1).

Every file shall be preceeded by a file header (starts 'HDR1')

User (optional) file header labels may be appended immediately after the required file header labels, and shall start with the characters 'UHL@' (in EISCAT's case 'UHL1'). Each header label group shall be immediately followed by a tape mark. (Note : at the beginning of a tape the volume header group is immediately succeeded by the first file header group with no tape mark between so that the first tape mark comes after the first header label group on the tape. In the case of a tape empty of all but the volume labels the volume label group will be followed by two tape marks (denoting end-of-useful-tape).

The last block of every file shall be succeeded by an end-of- file label (starts 'EOF1'). A tape mark shall immediately preceed every end-of-file label (ie there should be a tape mark between the data and the eof label). A user end-of-file label (starts 'UTL@') can follow immediately after the required file label ('EOF1') , forming an end-of-file group which shall be immediately followed by a tape mark.

The end of file group that immediately follows the last (or only) file on a tape (volume) shall be followed by two tape marks.

(There are further rules in the standards concerning the use of end of volume labels and other allowed labels which we shall not be concerned with since our tapes will be formatted such that a knowledge of these others is unnecessary.)

Volume header label structure ('VOL1')

Field 1	Label Identifier	Length 3	Shall Be 'VOL'
Field 2	Label Number	Length 1	Shall Be '1'
Field 3	Vol Serial No.	Length 6	Six @ characters to identify the volume
Field 4	Accessibility	Length 1	Space - means unlimited access. There are other rules concerning use of this field we shall not be concerned with since we use a space only.
Fields 5&6	Reserved for future Standardization	Lengths 20 &6	(All Spaces)
Field 7	Owner Identification	Length 14	Any @ characters - identifies the volume's owner
Field 8	Reserved for Future Standardization	Length 28	- Shall be spaces
Field 9	Label Standard Format	Length 1	See note below

NOTE: (for field 9) In BS4732 this is specified to be '1', as it was in the first, original ANSI standard. ANSI X3.27 (1978) (see below) is the third ANSI standard, and so defines a '3' to be put into this position. Since we are using a 'parochial' standard - ie one of our our own - EISCAT will define this to be 'E' (denoting E(iscat) as originators).

irst file header label ('HDR1')

ield 1	Label Identifier	Length 3	Shall be 'HDR'
ield 2	Label Number	Length 1	Shall be '1'
ield 3	FIle identifier	Length 17	Any @ characters
ield 4	Set Identifier	Length 6	Shall be spaces (not used by EISCAT)
ield 5	File Section No.	Length 4	'0001' - EISCAT shall not deal with any other type since anything else is only used in subsequent volumes of a multi-volume file.
ield 6	File Sequence Number	Length 4	numeric characters - ie '0001' to '9999' - denotes sequence of this file in the volume
ield 7/8	Generation No. And nversion	lengths 4 & 2	'0001' & '00' , fixed in all EISCAT productions
ield 9	Creation Date	Length 6	A space followed by the date in format YYDDD
ield 10	Expiration Date	Length 6	Format as field 9 - on EISCAT tapes normally set to ' 99365'
ield 11	Accessibility	Length 1	A space - meaning unlimited access - EISCAT not being concerned with anything else
ield 12	Block Count	Length 6	'000000' - See details of EOF1 label
ield 13	System Code (optional)	Length 13	Thirteen '@' characters to identify the operating system that created the tape - with EISCAT tapes this will be ' SINTRAN III ' (with blanks included as shown)
ield 14	Reserved for Future Standardization	Length 7	All spaces

irst end of file label ('EOF1')

Is identical to the HDR1 label over character numbers 5-54 and 61-80. The differences, in the other character positions, are as follows:

Field 1	Label Identifier	Length 3	Shall be 'EOF'
Field 2	Label Number	Length 1	Shall be '1'
Field 12	BloCk count	Length 6	Six numeric characters used to give the number of blocks within the just-ended file. From '000000' (if empty) to '999999'

User labels

Field 1	Label Identifier	Length 3	Shall be 'UVL' for user volume label Shall be 'UHL' for user header label Shall be 'UTL' for user termination (eof) label
Field 2	Label No	Length 1	Shall be '1' for EISCAT tapes. (Others permitted but not used by us.)
Field 3	User Optional	Length 76	Any '@' characters (see below on EISCAT user label formats)

Block structure of the data

Grouping records into blocks: no explicit indication of the boundaries between records is required.

Note : we do not follow this British standard where it states that there shall be an integral number of records to each data block.

ANSI X3.27-1978 Magnetic Tape Labels and File Structures for Information Interchange

(Points in addition to, or at odds with, BS 4732)

The standard introduces the idea of records spanning blocks and even tapes. This concept was not present in the British standard or the original ANSI standard 1 (X3.27-1969).

It is pointed out that a double tape mark can also occur if there is an empty data file - in which case it would follow a hdr label group not an eof.

Specifies that all data shall be recorded as characters as specified in the ANSI standard X3.4-1977. (This is essentially the ASCII codes.) EISCAT shall not generally follow this recommendation, except where character information is to be recorded, as , for instance, when symbolic files are copied.

ANSI X3.27 introduces a way of formatting spanned record type data, which EISCAT will not use, since, as it is diverging from the standard in not using ASCII character data files, it was decided it might also as well use its own data formatting method. See chapter three for the actual formats.

A system may override file attribute parameters found in labels being processed by that system with new values for those parameters provided by other sources. However, information found in the volume header (VOL1) is never overridden or ignored. The volume label may not be changed, added to or deleted. This does not preclude its being rewritten, along with the UVL1 label, with contents unchanged.

Creation date - shall be space followed by date in format YYDDD, and a valid date is ' 00000' (note not '000000')

Expiration date - format as creation date. Note ' 00000' is a valid default date but ' ' should not be. Note also that the default implies the file has from creation been expired - there is no default to say the file never expires. For this reason EISCAT tapes will have ' 99365' in this positon.

Accessibility: if extra control is ever needed within the EISCAT system, allowance is made in the standard for this since the accessibility field could be changed from blank to any of the characters A-Z, and this could be linked to an EISCAT generated standard.

When writing to a tape the procedure that should be followed is that the volume label is read first, then the header label to be sure that any file that exists already has expired, and is otherwise OK to be overwritten. This is not really applicable to HQ or the sites when gathering the data, since all stages of the tape copying/analysis chain will be by copying the previous stage over onto fresh, or freshly initialised tapes. Users should be aware that they should follow this standard, however, if they are to protect the integrity of their data.

The ANSI standard introduces the concept of the 'level' of tape label handling, depending on how many of the possible label types are dealt with, and which fields in these checked, modified etc. Within this concept EISCAT could be said to be supporting the standard to level 2 as defined. (This means dealing with VOL1, EOF1 and HDR1 labels, though not necessarily with all the fields in them.)

Before accessing a file for read, users should ensure that the volume accessibility field (pos 11) and the file accessibility field (pos 54) are blanks. For the present this will always be the case with tapes sent out by EISCAT, but needs for better security later may change this, and applications programs written now should take this possible change into account.

Appendix B of X3.27 (utilization of the standard) is a useful guide to how a tape should be dealt with , in terms of actions to be performed on encountering file boundaries of various forms.

The standard specifies that a tape sent out for the purpose of information interchange should first be initialised not only with a volume label, but also with a first file header, a zero-length file and a first file trailer (EOF). EISCAT , if asked for tapes for this purpose, can supply either that, or what may seem as good an alternative, which is a tape with a volume label set immediately followed by the end-of-tape double tape-mark. The potential user should specify which is required.

The standard points out that copying a volume in its entirety, labels and all, is allowed, but if done indiscriminately makes it very hard to keep track of which volumes are which, since there would be so much duplication. EISCAT will attempt to avoid this by, where possible, never carrying out such an operation. At the very least a copied tape shall have a different tape volume number.

The accessing procedures within the EISCAT system have been simplified by the addition of a field in the user volume label which tells at which stage of the EISCAT processing cycle that tape resides. This should make it unnecessary to carry out the complete range of checks on accessibility fields etc that the ANSI standard specifies although EISCAT will endeavour to ensure that the labels it produces, right through the cycle, continue to conform to the standard, for those who have software that automatically carries out the full range of checks.

A note is included in the X3.27-1978 standard about the use of the block count field (in HDR1 and EOF1). The name has been changed to 'block count' from 'no of blocks written' in the previous standard, since in the case where a block is written, backspaced and overwritten, this term can be ambiguous. The number given should be the actual number of blocks in the file, after all write operations have finished - what, in fact, will be found by the next person to read that tape.

Alan Farmer 29-apr-80

APPENDIX C

WTAPE Parameter set

(150380)

Description of System Parameter Block

Loc.	Symbolic name.	Explanation
1	ISITE	Site Code (1=Kiruna,2=Sodankyla,4=Tromso)
2 & 3	DUMP-TIME	Internal time,secs since start of year, for the current data array dump
		Parameters Describing Antenna Pointing
4-6	AZI	Azimuth of local site antenna (deg.)
7-9	ELEV	Elevation of local site antenna (deg.)
10-12	RANGE	Local range (km) (i e distance from local antenna to intersection region)
		Parameters Describing Reception System
13	IBAND	1st local oscillator = 1 (813.5 MHz) = 2 (1053.5 MHz)
14	IPHASE	Polarisation phase (to be clarified later)
15	IAMP	Polarisation amplitude (to be clarified later)
16	IPATH	Signal path switch 0 means channel X ->5,6,7,8; Y ->1,2,3,4 1 means channel Y ->1,2,3,4,5,6,7,8 2 means channel X ->1,2,3,4,5,6,7,8 3 means channel X ->1,2,3,4; Y ->5,6,7,8
17-18	ISIGATN(2)	Signal attenuators (0-63dB)
19-26	ILOC2(8)	Second local oscillator settings
27-34	ICHATN(8)	Channel attenuators (0-63dB)
35-42	IFILT(8)	Filters (kHz)
43	NOISE	= 0 0 Deg K (nominal) = 1 30 Deg K (nominal) = 2 100 Deg K (nominal) = 3 300 Deg K (nominal) =-1 undefined or under control of radar controller
44	IRFON	= 0 no RF injection = 1 RF injection

		Parameters Describing Data Collection System
		***
45	NPROG	Correlator program number
46-61	IAPB(16)	Correlator APB stack
62-77	IAPM(16)	Correlator APM stack
78-85	IRATES(8)	ADC Conversion rates (units 0.1 micro s.)
86-93	IFRADAR(8)	Radar frequencies (freq in MHz*10)
94	NINT	Integration time (secs)
95	NMAGIC	Magic number
96-127	FREE	Free
128	IVERSN	Version number of this parameter array

APPENDIX D

Definitions

(The following definitions appertain mostly to the usage as envisaged in the notes on WTAPE, tape formats, ANSI standards etc.)

Record:

A collection of related items of data which may be treated as a unit of information.

Each record may consist of one or more fields, where a field is a unit of information (i e one item of data). Fixed groupings of fields go to make up each type of record. One or more(though usually just one) record types is repeated as necessary to give a data file.

The record format -the way in which it is broken up into fields - and the record length both only concern the data's internal structure and how it is to be handled before and after recording, i e the record is a 'logical' unit and not a 'physical' one. (The latter specifying how it is laid down on recording medium.)

Block:

A group of contiguous characters recorded on and read from tape as a unit.

The block format - the way in which data, as a bit pattern, is read from and written to, tape - doesn't necessarily concern itself with the records and record structure contained within. (I e the block is a 'physical' unit and not necessarily a 'logical' one.) therefore there need not be, though it is often the case, a one-to-one relation between block and record lay-outs on a tape.

File:

A major collection of data, consisting of all records pertaining to a certain subject, experiment, etc. Often, though not necessarily, only one record type is repeated many times with different values in the fields.

Volume:

Physical unit of storage. In the accompanying documentation this can be considered synonymous with 'a reel of magnetic tape'

Label:

An identifying block at the beginning or end of a volume on a file. Labels are usually grouped in 'label-groups', with each label of a fixed size and format. See elsewhere for internal definitions. Labels are normally 80 or morecharacters in length. (In the EISCAT system 80 is the norm).

Tape mark:

Also called the eof-mark(er) or the end-of-file mark(er), this is a special configuration recorded on magnetic tape, essentially indicating the boundary between files and labels and between certain label groups.

Spanned record:
A record contained in a file in which each record may start on one block and end on another.

Record segment:
That part of a spanned record that is contained in any one block.

APPENDIX E

NORD-10 Reals

Representation of Real Variables in NORD10

Reals in the NORD10 are represented as 48 bits (that is, as three successive integer locations).

32 Bits are used for the mantissa magnitude, one bit for the sign of the number and 15 bits for the signed exponent.

The mantissa is always normalised 0.5 .GE. MANTISSA .LE. 1.0;

For all non zero numbers bit 31 equals one. The exponent is biased with 2**14, i e 40000z (octal) is added to the actual exponent, so that a standardised floating zero contains zeros in all 48 bits.

```
WORD            N                          N+1                  N+2
     ---------------------------------------------------------------------
     ! +/- !  EXPONENT            !      MANT . . .  ! . . . ISSA        !
     ---------------------------------------------------------------------
BIT     47   46 - - - - - - - - -32 31- - - - - - - - 16  15- - - - - - - - - 0
```

The accuracy is 32 bits or approximately 9 decimal digits. Any digit up to 2**32 - 1 has an exact floating point representation.

Examples:

1) WORD1: 040021Z Z IN THIS CONTEXT MEANS AN OCTAL
   WORD2: 170440Z REPRESENTAION OF THE NORD10 16
   WORD3: 000000Z WORD.

   STEP1. CALCULATE THE VALUE OF THE MANTISSA, THIS IS GIVEN BY 170440Z,000000Z I.E. BY THE BINARY NUMBER:

   .1 111 000 100 100 000 0 000 000 000 000 000

   $$= 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-8} + 2^{-11}$$

   = 0.5
   + 0.25
   + 0.125
   + 0.0625
   + 0.00390625
   + 0.000488281255

   = 0.941894531255

   STEP2. SIGN (+)IVE

```
     STEP3.   EXPONENT = 40021Z - 40000Z = 21Z = 17 (BASE 10)

     FINALLY THE NUMBER REPRESENTED IS  0.941894531255 * 2 ** 17
                                       =123456.00
2)   WORD1:   140120Z
     WORD2:   135764Z
     WORD3:   162165Z

     STEP1.   MANTISSA = 0.1 011 101 111 110 100 1 110 010 001....
                       = 0.73420551....
     STEP2.   SIGN     = (-)IVE
     STEP3.   EXPONENT = 40120Z - 40000Z = 120Z = 80 (DECIMAL)

     THUS NUMBER       =  -0.7342... * 2 ** 80
                       =  -8.876 E 23

3)   WORD1:   000000Z
     WORD2:   000000Z
     WORD3:   000000Z
     REPRESENTS 0.0
```

EISCAT publications

F. du Castel, O. Holt, B. Hultqvist, H. Kohl and M. Tiuri:
A European Incoherent Scatter Facility in the Auroral Zone (EISCAT).
A Feasibility Study ("The Green Report") June 1971. (Out of print).

O. Bratteng and A. Haug:
Model Ionosphere at High Latitude, EISCAT Feasibility Study, Report No. 9.
The Auroral Observatory, Tromsö July 1971. (Out of print).

A European Incoherent Scatter Facility in the Auroral Zone, UHF System and Organization ("The Yellow Report"), June 1974.

EISCAT Annual Report 1976.(Out of print).

P.S. Kildal and T. Hagfors:
Balance between investment in reflector and feed in the VHF cylindrical antenna.
EISCAT Technical Notes No. 77/1, 1977.

T. Hagfors:
Least mean square fitting of data to physical models.
EISCAT Technical Notes No. 78/2, 1978.

T. Hagfors:
The effect of ice on an antenna reflector.
EISCAT Technical Notes No. 78/3, 1978.

T. Hagfors:
The bandwidth of a linear phased array with stepped delay corrections.
EISCAT Technical Notes No. 78/4, 1978.

Data Group meeting in Kiruna, Sweden, 18-20 Jan. 1978
EISCAT Meetings No. 78/1, 1978

EISCAT Annual Report 1977

H-J. Alker:
Measurement principles in the EISCAT system
EISCAT Technical Notes No. 78/5, 1978

EISCAT Data Group meeting in Tromsö, Norway 30-31 May, 1978
EISCAT Meetings No. 78/2, 1978.

P-S. Kildal:
Discrete phase steering by permuting precut phase cables.
EISCAT Technical Notes No. 78/6, 1978

EISCAT UHF antenna acceptance test.
EISCAT Technical Notes No. 78/7, 1978.

P-S. Kildal:
Feeder elements for the EISCAT VHF parabolic cylinder antenna.
EISCAT Technical Notes No. 78/8, 1978.

H-J. Alker:
Program CORRSIM: System for program development and software simulation of EISCAT digital correlator, User's Manual.
EISCAT Technical Notes No. 79/9, 1979.

H-J. Alker:
Instruction manual for EISCAT digital correlator.
EISCAT Technical Notes No. 79/10, 1979

H-J. Alker:
A programmable correlator module for the EISCAT radar system.
EISCAT Technical Notes No. 79/11, 1979.

T. Ho and H-J. Alker:
Scientific programming of the EISCAT digital correlator.
EISCAT Technical Notes No. 79/12, 1979.

S. Westerlund (editor):
Proceedings EISCAT Annual Review Meeting 1969. Part I and II,
Abisko, Sweden, 12-16 March 1979.
EISCAT Meetings No. 79/3, 1979.

J. Murdin:
EISCAT UHF Geometry.
EISCAT Technical Notes No. 79/13, 1979.

T. Hagfors:
Transmitter Polarization Control in the EISCAT UHF System.
EISCAT Technical Notes No. 79/14, 1979.

B. Törustad:
A description of the assembly language for the EISCAT digital correlator.
EISCAT Technical Notes No. 79/15, 1979.

J. Murdin:
Errors in incoherent scatter radar measurements.
EISCAT Technical Notes No. 79/16, 1979.

EISCAT Digital Correlator. TEST MANUAL.
EISCAT Technical Notes No. 79/17, 1979.

G. Lejeune:
A program library for incoherent scatter calculation.
EISCAT Technical Notes No. 79/18, 1979.

K. Folkestad:
Lectures for EISCAT Personnel, Volume I
EISCAT Technical Notes No. 79/19, 1979.

Svein A. Kvalvik:
Correlator Buffer-Memory for the EISCAT Radar system
EISCAT Technical Notes. No. 80/20.

P-S. Kildal:
EISCAT VHF Antenna Tests
EISCAT Technical Notes No. 80/21

J. Armstrong
EISCAT Experiment Preparation Manual
EISCAT Technical Notes No. 80/22