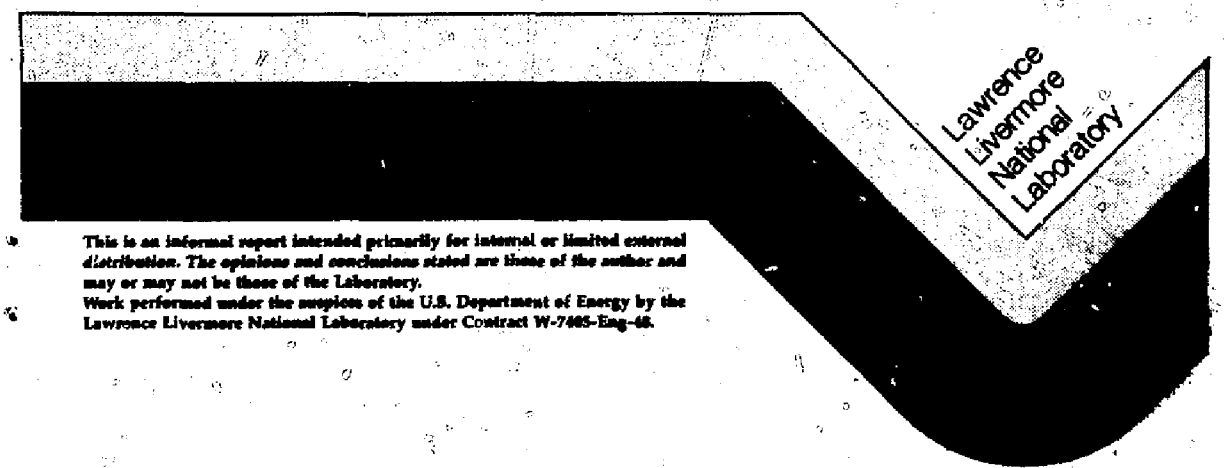
UCID- 30206

A USER'S GUIDE TO REVERT
A CDC 7600 PROGRAM FOR CONVERT NG SPENT FUEL TEST - CLIMAX DATA TO ENGINEERING UNITS, WITH CORRECTIONS

GLENN HAGE

OCTOBER 1984



UCID--30206

DE85 002690

CONTENTS

Introduction 1
Major Routines and General Flow 4
Operation 10
ALTER Operations and Error Messages 13
Acknowledgments 19
References 20
Appendix A Glossary 21
Appendix B Access to Programs and Data 25
Appendix C Subroutine Library Files 26
Appendix D Conversion Coefficient Database Files 32
Appendix E ITEMNAME File 36
Appendix F HPTP Logical File Formats 38
Appendix G Program Library and Associated Routines 41
Appendix H Error Messages 45
Appendix I TAPECOPY 48

LIST OF FIGURES

Fig. 1	Generalized flow of control showing responsibilities of the major routines in REVERT	3
Fig. 2a,b,c	Subroutines in SFTLIB, their interaction and use of common blocks	27,28,29
Fig. 3	Subroutine use in SFTLIB	30
Fig. 4	Common block use by subroutines in SFTLIB	31
Fig. 5	The ITEMNAME file	37

LIST OF TABLES

Table 1	Physical device types and their use by PROCS subroutine	6
Table 2	Some contents of the .SFT directory	25
Table 3	A comparison of floating point, binary numbers on the HP 1000 and the CDC 7600	40

MICROFICHE

Fiche 1a	Printed output from a sample run	microfiche
Fiche 1b	Listing of REVERT AND SFTLIB subprograms	microfiche

ABSTRACT

A CDC 7600 computer program, REVERT, can revise Spent Fuel Test-Climax data files using one of several algorithms, depending on the type of data. The algorithms use coefficients from a separate file organized by data type identifier. REVERT can also make that file of coefficients, using data from tapes made by Hewlett-Packard equipment employed for data acquisition on the Spent Fuel Test - Climax at NTS.

INTRODUCTION

The Spent Fuel Test-Climax (SFT-C) is being conducted under the technical direction of the Lawrence Livermore National Laboratory for the U.S. Department of Energy (DOE). As part of the Nevada Nuclear Waste Storage Investigations, it is managed by the Nevada Operations Office of the DOE.

The SFT-C is located 420 m below surface in the Climax stock granite where facilities were constructed between June 1978 (when funding for the test was initiated) and April 18, 1980 (when spent-fuel emplacement began). Spent-fuel emplacement between April 18 and May 28, 1980, spent-fuel exchanges in January and October 1981 and August 1982, and spent-fuel retrieval between March 3 and April 6, 1983 combined to meet the operational objective of the test: demonstration of safe and reliable packaging, transport, short-term storage, and retrieval of spent nuclear reactor fuel.

A technical measurements program was undertaken to acquire data concerning the ultimate qualification of granitic rock as a repository medium, as well as the design and prediction of the response of such a repository in granitic rock.[1] Temperature, displacement, stress, radiation, and test status data have been recorded continuously on more than 900 channels during the three year storage phase of the test and continued for six months following retrieval to record thermal and thermomechanical responses during cooling of the rock surrounding the facility. The data acquisition system (DAS) consisted of dual HP1000 minicomputers and associated peripherals.[2] Copies of magnetic tapes of the acquired data are periodically shipped to Livermore for processing, archiving and data analysis.

REVERT*, a CDC 7600 program, is used by the Spent Fuel Test - Climax Project to re-convert data values that were collected on a Hewlett-Packard 1000 database at NTS.[6] That system also converts raw data (electrical units of measure) to engineering units. Both the raw and engineering data values, along with time (Julian day, to five decimal digits) and two device identifying fields (a "license plate" and "channel number"), are periodically written on magnetic tape. This data is processed, after files are made from the delivered tapes, by the BREAD program[3] which produces a sorted, BCD file with incorrect data removed. One or several of these files are merged and converted to binary (five words per data point) by the MFAB program.

Data conversion coefficients and control parameters from several tables on the H-P system are written on magnetic tape before significant changes affecting data conversion are made. These tapes are delivered to Livermore for the making of a printed record and for use by REVERT in more sophisticated conversion algorithms.

REVERT has two independent paths of control. One is to read the dump tapes described in the paragraph above and make a file that, two, can be randomly read for coefficients during the "re-conversion" process.

The source language of REVERT is LRLTRAN,[9] which is an LLNL version of FORTRAN.

---

* A glossary is provided in Appendix A to define all names that are unique to this program, and other computer-oriented terms. It is directed mainly to readers who are not programmers.

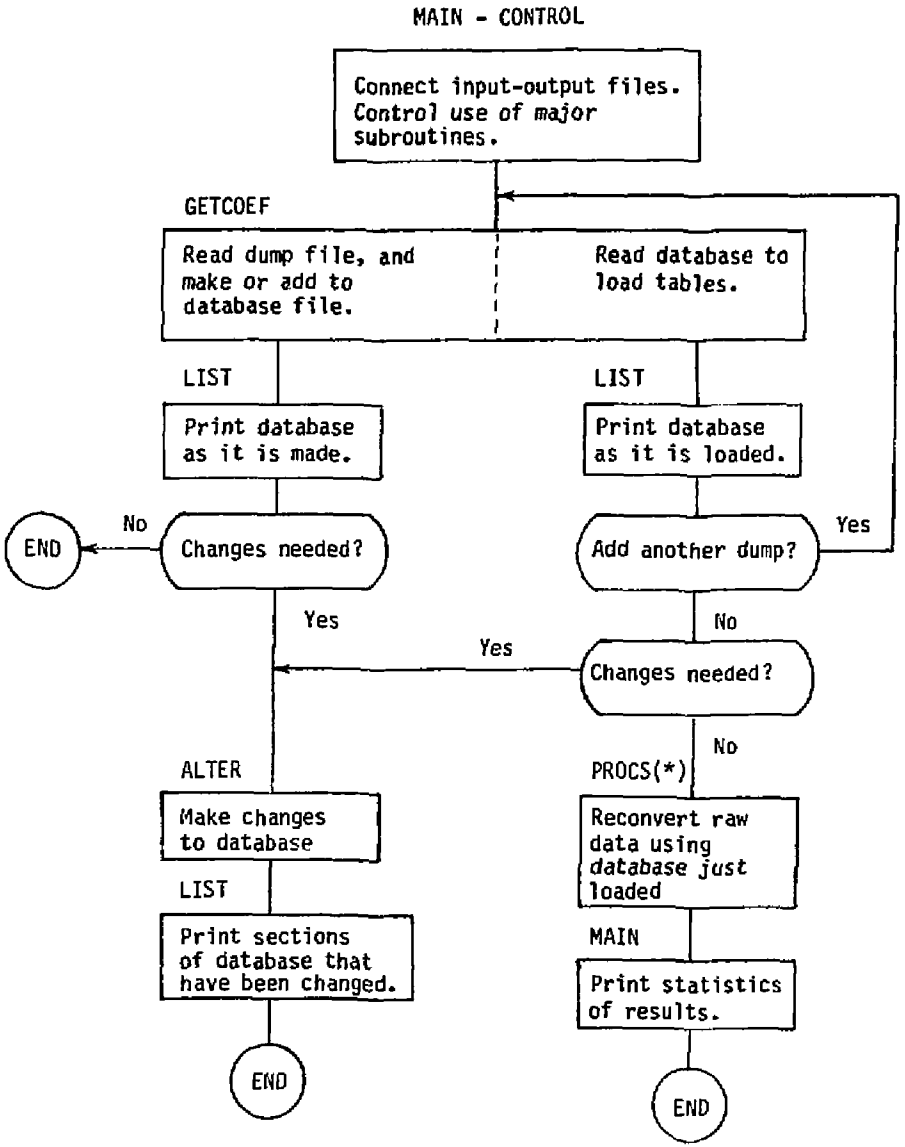


Figure 1. Generalized flow of control showing responsibilities of the major routines in REVERT

MAJOR ROUTINES
General Flow

Figure 1 shows in a very general manner the major routines in REVERT, their purposes and interactions. Placement is roughly in sequence of use but some routines are actually internal to others, since all except MAIN are subroutines. Most data is shared in numbered or named common blocks, but there exist several subroutines that communicate entirely via argument lists. Each major routine is briefly described here. Further details can be found in later sections of the report describing files made or used, the user dialog, and error messages. Complete listings of all programs are on the attached microfiche.

MAIN - uses GETCOEF and PROCS.

The program identifies itself and types the present time and date on the user's terminal. After system services have been used, optional user input is requested. GETCOEF is always called. PROCS is called if GETCOEF doesn't terminate due to changes having been made to the database. After PROCS is used, tables are printed showing the total number of data records on the data file, and total number of references to them, listed in order of license and device type. A few summary totals are then listed: grand totals and error-anomaly counts of missing reference data, data time outside database range, and instrument licenses and device types not in the database used. The time-range of the data file is printed last. An example is shown on the attached microfiche.

GETCOEF - uses LIST and ALTER

Notice on Figure 1 that GETCOEF has two exit paths, from two independent functions. Both functions involve the coefficient database - making or reading it, modifying it and making a printed listing - which is why the functions were combined. The form and number of filename(s) entered by the user determines which of four options in the program will be exercised. Database files made by REVERT for its own use have names beginning with "HPDB". The four options are:

1. Load tables from an HPDB file, with options for listing it and reconverting a data file.
2. Load tables from an HPDB file, listing it if desired; then modify the HPDB file with a dump file from NTS, with an ALTER file, or by interaction from the terminal. No reconversion run can be made in this mode.
3. Read a dump file from NTS and make an HPDB database file. Changes can be made from the terminal or an ALTER file. Listings are always made. No reconversion run can be made in this mode.
4. Entering two HPDB filenames implies a compare of the two is desired. This use has been replaced with DBCOMP program (available in the same library file as REVERT), which is dedicated to this task for efficiency.

PROCS - uses GCBFR (indirectly)

This routine controls the reconversion of the data file, storing the results in the fifth word associated with each datum. The work is done in three passes over the data file, as shown on Table 1.

Pass one reads the data file sequentially, recording the first and last record number of each license found in the data file (which is sorted by instrument license and by time within each license) in a table. The table, containing licenses, channel numbers and device types was previously loaded from an HPDB database file by GETCOEF. Also stored in the table at this time are the first and last times from the data file for each license. Complicating this pass is the possibility of duplicate licenses in the database. The first copy of a license is used until the time-range in the database associated with that copy is exceeded by the time associated with the datum. Then the next copy is used, and the table of licenses and record numbers is revised. Conversion of raw data to engineering units is accomplished for those device types which do not need converted data from other device types. None of the conversion subroutines used in this pass use GCBFR.

Pass two is run sequentially over the stored table of licenses, looking for device type TC. Other device types are ignored in this pass. Since duplicate entries of some licenses are in the table, if data for a license are

TABLE 1. Physical device types determine which FIT subroutines are used by the PROCS subroutine for conversion of raw data to engineering units. They use data from other devices collected at or near the same time, if necessary.

device type	pass	uses data from type(s)	definition
ES	1	-	Excitation supply
LD	1	-	Linear devices
PN	1	-	Polynomial coefficient devices
RT	1	-	Reference block temperatures
SD	1	-	Status devices
WT	1	-	Watt transducer
TC	2	RT	Thermocouples
RX	3	TC, ES	Rod extensometers
VW	3	TC	Irad gauges
WX	3	ES, RT	Wire extensometers
CA	*		Continuous air monitors
MD	*		Miscellaneous devices
RA	*		Air monitors
SY	*		System-data acquisition use

* Data from these devices are not saved in the present processing system, but remains on the archived raw data tapes.[3]

outside the time-range for one copy in the table, the proper one is found and used.

Pass three is similar to pass two, except that device types RX, VW, and WX are converted. Some reference data must be converted by pass two before it can be used here.

GCBFR - is used by the conversion subroutines that are called by PROCS.

This is the innermost major subroutine. It is used to retrieve a pair of records of data bracketing a given time, for a given license. The calling conversion (FIT) subroutine provides a license, time, and a time-range. GCBFR searches the data file for the license and loads a buffer with about 100 five-word data values having that license near the given time. Then it searches for two times that bracket the given time, and returns the two times and two data values to the FIT subroutine that called it. If no pair of bracketing times come within the time-range provided, the data is considered too old, and an error switch is set. Each FIT subroutine may handle errors uniquely, but for this type of error a value -9999.0 is made the "result" of the reconversion of that datum. During the search, data values of -9999.0 are ignored, since this is a code used by the data acquisition system to represent the absence of useful data. Each FIT subroutine calls the RVAL function to interpolate linearly and return a reference datum at the exact time desired. Data collected within two minutes of the required time bypass the bracketing search and are delivered as both the high and low-side values. This simplifies the linear interpolation done in RVAL for the FIT subroutines. The ten FIT subroutines and the RVAL function were designed by R. C. Carlson.

This program is necessary because data files written by BREAD[3] are sorted by license to facilitate long term storage, analysis, and plotting.

LIST - is used by GETCOEF and ALTER

The LIST subroutine is a collection of printing formats for the coefficient blocks in the database, one for each device type. LIST may be optionally used each time a coefficient block is read from an HPDB database file preparatory to a reconversion run. Either a full double-spaced listing

of the block is made (up to 12 lines) or a single-line listing consisting only of references to other blocks (licenses) can be produced. When a new HPDB database is being made or added to, a full listing of each block is made. In this latter case, reference channel numbers are printed instead of licenses since they cannot be translated until a table of license - channel number pairs has been filled.

When a coefficient block has been altered, either through the use of an alter control file or via the user's terminal, a full listing (with reference licenses, if any) is made.

The full listing runs to 180 pages at present; the merge and alter records are not only longer but less orderly, because during file merging the listing is added to as new licenses are randomly found and during altering many coefficient blocks are repeatedly altered in the course of reading hundreds of alter commands. For this reason, three tables are printed following the first-described use (loading preparatory to a reconversion run). They are 1) a multicolumned table of channel numbers, licenses and "line" numbers, sorted by channel number, 2) the same items, sorted by license, and 3) a cross-index of all reference licenses, showing references to each coefficient block (license). See the attached microfiche for an example.

The data for the cross-index are collected while printing the full or reference listing, transposed in one pass and printed in a third pass. The sortings are done by channel first, then by license, so that the table necessary during execution of a reconversion run will be sorted appropriately for subsequent use.

ALTER - is used by GETCOEF - uses LIST

ALTER was originally a part of GETCOEF, used for making a few changes to the coefficient database from the user's terminal. Consequently it is not programmed to run efficiently for its present use. Reconstruction of the coefficient database has been required numerous times to accommodate modifications and replacements of instruments. This led us to make the process file-driven, and experience has led to the creation of more sophisticated operations.

The five operations are:

1. A - add a single license and its conversion constants
2. C - change the conversion constants of a single license
3. CD - change all blocks of a specified device type

4. CM - change all blocks for instruments which have a partial (masked) license in common
5. CO - change as in CM with data from another block

The section titled ALTER OPERATIONS contains a complete description of the rules involved.

As currently used, the ALTER file contains about 1000 operations, some of which modify all blocks.

SHELF - is used by GCBFR and PROCS

This is not a major subroutine, but it is crucial to efficient execution during a reconversion run. The purpose of SHELF is to minimize the number of disk reads and writes (I/O) from and to the binary data file (B-file) that is being reconverted. The buffer used by PROCS in its first pass, and by GCBFR for reference data, is one disk sector (512 words) long. On a typical one million word data file, the first pass of PROCS would read and write this buffer about 2000 times. GCBFR, in reading reference device data would read into the buffer hundreds of thousands of times. In an effort to reduce this I/O time, which slowed reconversion runs to about 30 minutes, without adding the logic of double-buffering or upsetting the "look-up" logic of GCBFR, the SHELF subroutine was devised.

SHELF remembers disk sectors that have been read by putting them in an LCM block 80 sectors in length. Actually, calls to SHELF replace I/O calls (BUFFER IN and BUFFER OUT). Fit subroutines that use GCBFR to get reference data need as many as 16 different devices. Since some devices have 400 or more data points per data file, 50 or more sectors are often searched to reconvert a sequence of data points. A list of disk addresses is saved, as well as a serial number, in order to determine if a wanted sector is available on the LCM shelf. If not, the space with the lowest (oldest) serial number has its data replaced. "Thrashing" of the sector spaces occurs whenever a reference device has many more than the average number of data points, and experience has caused us to make the shelf a little longer than the worst expected case. The present version runs about seven times faster than the no-shelf version.

Regarding PROCS, the shelf space would sit idle during the first pass, since there is no data being referenced, so it was used as one grand buffer, reducing the number of disk reads and writes from about 2000 to about 25.

OPERATION

REVERT is able to process several different files, as shown on Figure 1. Depending on how REVERT is used, the user will need differing mixes of the various files. The section ACCESS TO PROGRAMS AND DATA demonstrates our present library of resources and raw material, which is changing with use. Basically, there are five uses for REVERT:

1. Make a 7600 compatible database from an HPTP file
2. Merge* a database
3. Modify a database, using ALTER mode
4. List a database
5. Use a database to process a data file.

This section describes the dialog and files necessary for each use. REVERT'S mode of operation is determined by the response to "ENTER COEFFICIENTS DATA FILE NAME(S)". A single file name, beginning "HPOB" forces the list and/or reconvert modes. No ALTER or merge is allowed. A second name on this line determines the type of ALTER or merge desired and prevents a reconversion run. An HPTP... file name triggers the merge mode. "FILE" or "TTY" determine the source of ALTER information and trigger the ALTER mode. Parts of messages in brackets below are variable; they may contain an example or description of their contents. Note that "U:" is a label for a line that contains a user entry, which is underlined; "R:" is for a message from REVERT; and "R/U:" is a label for a line that contains both a message from REVERT and the user's response, which is also underlined. Each entry or response is terminated with the RETURN key.

MAKE A DATABASE

```
U:    REVERT / I I
R:    REVERT [10E 02/03/84R 14:02:24]          R is the computer identifier
R:    THIS RUN [R 02/03/84 14:10:00]
```

* Merging of a database means the adding of coefficient blocks from a dump file to an HPOB database, but only when a new license is encountered.

```
R/U:  ENTER COEFFICIENTS DATA FILE NAME(S)  HPTP831024
R/U:  REPORT GENERATED [83/10/24 18:05 GMT (45632.25401)]...OK? (Y/N) Y
R/U:  ENTER ITEM-ALTER FILENAME IN12L
      . . . about five minutes elapse . . .
R/U:  [HPDB831024] READY.  CHANGE OR END? END
```

MODIFY A DATABASE

```
U:    REVERT                          The time request is an implied one minute
R:    REVERT [IOE date time]
R:    THIS RUN [R date time]                          R is the computer identifier.
R/U:  ENTER COEFFICIENT DATA FILE NAME(S) HPDB831024 FILE
R/U:  ENTER ITEM-ALTER FILENAME RCB
R/U:  [HPDBdate] READY.  CHANGE OR END?  E            Type C to make TTY changes
```

MERGE A DATABASE

```
U:    REVERT                          The time request is an implied one minute
R:    REVERT [IOE date time]
R:    THIS RUN [R date time]                          R is the computer identifier
R/U:  ENTER COEFFICIENTS DATA FILE NAME(S)  HPDB831024 HPTPdate
      . . . a few minutes elapse while the HPDB file is loaded . . .
R/U:  REPORT GENERATED [date time GMT (Julian day)] . . . OK? (Y/N) Y
R/U:  ENTER ITEM-ALTER FILENAME IN12L
      . . . a few minutes elapse . . .
R/U:  [HPDBdate] READY.  CHANGE OR END? C
      . . . Dialog with ALTER section begins . . .
```

LIST A DATABASE

```
U:    REVERT                          The time request is an implied one minute
R:    REVERT [IOE date time]
R:    THIS RUN [R date time]                          R is the computer identifier
R/U:  ENTER COEFFICIENTS DATA FILE NAMES(S) HPDB4/83M8
R/U:  LISTING OF [HPDB4/83M8]?
R/U:  - FULL, PEFERENCES, OR NONE (F/R/N) F
R/U:  [HPDB4/83M8] READY.  RUN OR END?  END
```

```
USE A DATABASE TO PROCESS A DATA FILE
U:    REVERT / 6 6                          NOTE the larger time request!
R:    REVERT [10E date time]
R:    THIS RUN [U date time]
R/U:  ENTER COEFFICIENTS DATA FILE NAME(S)  HPDB831024
R:    LISTING OF [HPDB831024] ?
R/U:  - FULL, REFERENCES, OR NONE (F/R/N) R
      . . . a few minutes elapse . . .
R/U:  [HPDB831024] READY.  RUN OR END?  RUN
R/U:  TYPE DATA FILENAME B44733           This filename and the N or R below
                                          could have been typed on the execute
                                          line (the first line), eliminating the
                                          requests.
R:    TYPE N TO PRE-FILL WITH -9999.,
R/U:       R TO USE UNCHANGED . . . N
R:    RECONVERTING DEVICE TYPES WT ES RT PN LD SD        (This is pass one.)
R:    RECONVERTING DEVICE TYPE TC                       (This is pass two.)
R:    RECONVERTING DEVICE TYPES WX VW RX                (This is pass three.)
      . . . a longer wait . . .
R:    ERRORS RECONVERTING
R:    [226 RX]                            Only non-zero totals are printed.
        .  .
        .  .
        .  .
R:    TIME RANGE - [44732.00227 44732.99999]
R:    DATA REPROCESSING COMPLETE
```

First, it must be noted that REVERT does not in all cases expect an ALTER or prompt the user for one. Quality assurance demands that an execution of REVERT that *modifies a database not be used to reconvert* a data file using that database. This is alluded to in the description of GETCOEF and in Fig. 1. Control is implemented via responses to the prompt "ENTER COEFFICIENTS DATA FILE NAME(S)". When a single name of an "HPD8"-prefixed database file is given, no modifications are permitted.

To ALTER such a file, a second word must be entered. If an NTS dump file is to be added, that name is entered on the same line. Changes via the user's terminal will be requested after the files are merged. To ALTER an HPDB file directly, type "FILE" or "TTY" on the same line as the HPDB filename. Changes or the ALTER filename will be requested after license tables are loaded. When loading an NTS dump file to make an HPDB database, the fact that the filename does not begin with "HPDB" allows the ALTER option to be requested after the database has been made. The OPERATION section has a more detailed description of the messages involved.

ALTER FILE

Five operations are available: add, change, change device, change with mask, and copy. Each uses the same format, shown below.

```
                                                    1                   2
column numbers| 1 2 3|4 5 6 7 8 9|0|1 2 3|4 5 6|7|8|9|0
field labels  | Op   |L  or   D  |S|    X|    N| |T| |Data
```

where,

- Op - an operation: A, C, CD, CM, or CO, left justified
- L - a license e.g., CAM101
- D - a device type, left justified e.g., RX
- S - a suffix used to identify duplicate copies of licenses (any character)
- X - a right justified integer - the location in a block at which to start storing data

N - a right justified integer - the number of items to store. Maxima are listed with data, below. If N = 0 the line is ignored.

T - the type of data to follow: I for integer; E for real, floating point; A for ASCII (BCD)

Data - maximum 11 integers (11I10), 8 real (8E13.6), or 8 BCD (8(A10,1X)) Commas may terminate the I and E fields; the 10-column ASCII fields are separated by single space, which is ignored. One line can have only one type of data.

Some general rules and opportunities -

- Blank lines are acceptable (and are ignored).
- Comments may be effected by leaving lines blank up to column 17, with an A in column 18. The data field, columns 20 through 120, are available for comments. At the beginning of the file, only the first three columns need be blank.
- An operation may be continued on subsequent lines by leaving Op and L fields blank.

The five operations -

A - Add a single coefficient block identified by a new license, and its conversion constants. A duplicate license may be created by adding a unique non-blank character in column 10 (a suffix). This character is used only during the ALTER and LIST phases, not during data reconverting. This allows different coefficient blocks to be applied to the same license.

C - Change data for a single coefficient block identified by its license. Duplicate licenses cannot be changed as a group with this operation. To change data for a particular copy of a duplicate license, use the S-field character that matches the one used when it was added. With a blank in the S-field, data is changed for the first copy of the license.

CD - Change all blocks for the specified device type. Each coefficient block is identified by a channel number, license, and physical device type. This operation allows one to broadcast certain values over the entire database.

CM - Change all blocks for which the un-masked portion(s) of the license matches. The mask character is *. For example, if a masked license were G*E011, all coefficient blocks identified by licenses GAE011, GBE011, etc. will be changed. The mask character may be in any position, or all positions. The latter case, license ******, would make changes to every block in the database. A license without any * is also accepted, since the effect is to change the CM operation to a C.

CO - Change all blocks for which the un-masked portion(s) of the license matches, as in CM operations, but copy data from another masked license. The X and N fields control the amount of data to be moved. The source license is in the data field, left justified, and therefore an A is necessary in the type field. For example, if the first license is G*E011 and the data license is G*E211, data are copied from GAE211 to GAE011, GBE211 to GBE011, etc. Discontinuous locations in the coefficient blocks can be moved by the use of continuation lines, which need only their X and N fields to specify their ranges.

TERMINAL DIALOG

Only the add and change operations are available via the user's terminal dialog. This, in fact, was the first version of ALTER, before we were aware of the great number of changes that would be necessary to make a working database.

As in the use of ALTER files, described above, the dialog described below is started by adding "TTY" as the second item in response to the request "ENTER COEFFICIENTS DATA FILE NAME(S)." In all cases other than a single HPDB file name, which starts a reconversion run, REVERT finishes with the message "[HPDByymmdd] READY. CHANGE OR END?" "CHANGE" or "C" will then begin the dialog described below. If an HPTP file is being loaded, "TTY" is not necessary, since REVERT already knows that this is not a reconversion run. Similarly, if an HPDB file is being loaded with "FILE" as the second item of the response in order to use an ALTER file, REVERT knows that it is not being used for a reconversion. (Also, see the OPERATION examples starting on page 10.)

The requirements for responses to the various messages are described in detail below. Note that "R/U:" identifies a line containing a prompt and response: (Each response is ended with a RETURN key.)

R/U: CHANGE OR ADD (C/A) AND LICENSE

The first character typed must be an A or C.

A six-character license follows, within the next ten spaces, e.g.,

C MBIOIO or A AAZ999

Only the first copy of duplicate licenses can be changed.

R/U: TYPE PD-TYPE AND CHANNEL (AA N,)

The first two characters must be one of the 13 physical device identifiers used in the database.

The channel number field must end with a comma, and fall within seven spaces of the device characters.

Blanks, or a blank line, *imply no change to these fields for the* selected license; for an addition to the database, blank input is nonsense.

R TYPE LOCATION, COUNT, AND E I OR A (L, C,E) OR RETURN

R/U: &

Integers between 1 and 100 may be entered for the L and C data placement fields, each terminated with a comma.

The type of data to follow is indicated by an I, A, or E immediately following the second comma, where the I, A, and E represent integer, ASCII (BCD), or real, floating point data respectively.

R [Data that currently resides in the selected locations is typed here.]

R/U: #

Data entered following this prompt must follow FORMAT *limits depending on the data type* character entered on the previous line:

I - 10I6 No more than ten consecutive integers separated by commas (including a final one).

A - 10(A10,1X) No more than ten, ten-character fields, space separated. Licenses as data must be right justified.

E - 10E14.0 Real, floating point data may be packed on a line by using decimal points in the fraction parts and terminating *each datum (with or without an exponent) with a comma,* e.g., 5.0, 10.E10, .01, 1.E-2. Spaces are allowed everywhere except before a comma *following an E, as in the second and fourth examples. They are not necessary* anywhere.

R/U: &

Response is the same as for the & above.
To terminate work on a license, hit RETURN.
To terminate all add/change work, hit RETURN again after the first prompt.

ERROR MESSAGES - Causes and Results

[License] NOT FOUND

A change line from terminal has referenced a non-existent license. The request will be repeated.

[License] ALREADY EXISTS

An add line from terminal would make a duplicate license, which is not allowed except from ALTER file lines with license suffixes. The request will be repeated.

[License] NOT FOUND. C-LINE [n]

A change line from an ALTER file referenced a non-existent license, or the suffix given did not match one of a duplicate license. The next line used will be one with a non-blank operation field.

[License] ALREADY EXISTS. A-LINE [n]

An add line from an ALTER file would make a duplicate license, or a duplicate one with a duplicate suffix, which is not allowed. Use a new suffix character on this line. The next line used will be one with a non-blank operation field.

ERROR IN ALTER FILE, LINE [n]

The data type is not an A, I or E. The number of items on the line (the N field) is greater than 100, or negative. The location (X field) is greater than 100, 0, or negative. Or the sum X + N - 1 is greater than 100 (the location of the last datum). The license field is blank. The operation is not A, C, CD, CM, or CO. The next line used will be one with a non-blank operation field.

I/O ERROR DURING UPDATE, LINE [n]

Any hardware error reading or writing the HPDB database file during the ALTER that is detected by the software or operating system will cause this message to be typed. There are several different causes. REVERT does not try to recover, but stops immediately since analysis is not worth the effort for these rare events. This message has never been used.

[Device] NOT FOUND. CD LINE [n]

The device type for a CD operation was not in this HPDB database. Probably the ALTER line was mistyped, but it is possible to create a database without a particular device type. The next data line in the ALTER file that will be interpreted by the ALTER subroutine will be the next one with a non-blank operation field.

[License] NOT FOUND. CM LINE [n]

The masked license in the license field did not match any similarly masked license in the HPDB database. The next data line in the ALTER file that will be interpreted by the ALTER subroutine will be the next one with a non-blank operation field.

[License] NOT FOUND. CO LINE [n]

The masked license in the license field did not match any similarly masked license in the HPDB database. If there is no similar match of a data field license, no data is copied, and no notice is given. The next line used will be one with a non-blank operation field.

ACKNOWLEDGMENTS

The program originated with R. A. Nyholm on the CDC 7600. Many subroutines came from the HP 1000 Data Acquisition System.[6] Some details for this report also came from the data acquisition system report. The many features added since were done at the request of R. C. Carlson or done to make his use of this program easier or more accurate. Other subroutines were supplied by D. N. Montan and J. M. Richter. The gracious ladies of the Correspondence Center deserve a special thanks for putting this document into readable form.

REFERENCES

1. L. D. Ramspott, et al., Technical Concept for a Test of Geologic Storage of Spent Reactor Fuel in the Climax Granite, Nevada Test Site, LLNL UCRL-52796, 1979.
2. Nyholm, R. A., Brough, W. G., and Rector, N. L., Spent Fuel Test-Climax Data Acquisition System Integration Report, LLNL UCRL-53304, June 1982.
3. Hage, G. L., Bread A CDC 7600 Program That Processes Spent Fuel Test-Climax Data, LLNL UCID-19814, April, 1983.
4. Control Data 7600 Computer System, Reference Manual, Control Data Corporation, Arden Hills, MN, Publication 60258200, 1971.
5. Hewlett-Packard 1000 RTE-IV Assembler Reference Manual, Hewlett-Packard, Cupertino, CA, Publication 92067-90003, 1980.
6. Nyholm, R. A., Spent Fuel Test-Climax Data Acquisition System Operations Manual, LLNL UCID-19834, January 1983.
7. Gary Long, The BCON Interpreter for the CDC 7600 and the CDC STAR-100, LLNL UCID-30101, January 6, 1975 plus undated revisions.
8. Edna Carpenter, et al., ORDERLIB Subroutine Library, Livermore Computing Systems Document M-026 LTSS Part IV, Chapter 304, January 1977.
9. William Derby, et al., LRLTRAN Language Used with the CHAT and CIVIC Compilers, Livermore Computing Systems Document LSCD-302, March 1981.
10. Neale Smith and Rich Belles, XPORT REFERENCE MANUAL, Livermore Computing Systems Document LCSD-1196, December 1981.
11. Pierre Du Bois, et al., CDC 7600 System Calls and I/O Requests, Livermore Computing Systems Document M-026, LTSS Part I, Chapter 10, November 1977.
12. Dusenbury, Karl, et al., Summary Sheets, Livermore Computing Center Document LCSD-50, Rev. 1, May 1982.

APPENDIX A - GLOSSARY

Some names of subroutines and other computer-oriented terms are defined here, as well as I know them, in order to speed the reading of this document by those who are already familiar with the terms. Capitalized terms in the definitions are themselves defined here. All subroutine names were coined by R. A. Nyholm unless credited otherwise.

ALTER - This subroutine, named by this author, alters the coefficient database, thus the name.

ASCII - American Standard Code for Information Interchange is a type of BCD, a 6-bit variation of which is used in LTSS on the CDC 7600.

BCD - Binary Coded Decimal is a method of storing letters, numbers and characters in a binary computer. In a 6-bit code, 2^6 or 64 different arrangements may be made equivalent to 64 different symbols. For example, 010 001 (or 21_8) is used for a 1 (one) and 111 010 (or 72_8) is used for a Z.

BREAD - This is a program that processes data files into a form usable by REVERT. The data is originally in binary form, which is read and output later in ASCII form. The program was written by R. A. Nyholm, J. M. Richter, P. E. Harben, and G. L. Hage.

BUFFER - A block of memory in a program that is usually reserved for data being read from or written to a disk file.

CALL - The use of a subroutine by another subroutine or the MAIN part of a program is called a CALL.

CHANNEL - Each data collection device is identified to the data recorder, the HP 1000, via a logical channel number. Channels are numbered from 1 through 512 and 1001 through 1512. Every datum is recorded along with its channel number.

COEFFICIENT BLOCK - The HPDB DATABASES have a coefficient block of 100 words for each LICENSE. There are about 1200 in an HPDB. The appropriate one is loaded into REVERT during the reconversion of data having a particular LICENSE.

DATABASE - Our use of the term means a file of data that is organized so that it can be referenced randomly (as opposed to serially). In REVERT, it is the file called the HPDB which contains the coefficients that are used by the FIT subroutine formulas.

DUMP FILE - The HP 1000 data collection system contains a database of 1) coefficients used in fit routines for the conversion of data to engineering units 2) alarm messages 3) other data useful for the HP 1000 data collection system. When there is a change to the coefficients, a tape is made containing the previous values. Copies of these tapes are delivered to LLNL and are processed by TAPECOPY and REVERT to make a database of coefficients for reconversion runs. See HPTP and HPDB.

FIT - The subroutines that reconvert data using specific formulas and coefficients from the DATABASE are called FIT routines.

GCBFR - This subroutine gets the constant from BUFFER for all FIT subroutines.

GETCOEF - This subroutine's name is a contraction of "get the coefficients," that is, to read the database file of coefficients that are necessary to reconvert data to engineering units.

HP - Hewlett Packard Company is the manufacturer of the HP 1000 that is used to collect data at the NTS.

HPDB - Hewlett Packard database files are derived from HPTP files using REVERT. These files are named HPDByymmdd, where yy are the last two digits of the year, mm are the month (January is 01, December is 12), and dd are the day. REVERT originally made a file HPDATABASE for this purpose, but since we needed to save a continuing series of such files, the date was taken from within the HPTP file to create a unique name. This convention was established by G. L. Hage. The current database is HPDB4/83M8. The 4/83 indicates that it was merged from a collection of HPTP files ending with one from April, 1983, and was subsequently ALTERED with file RC8.

HPTP - All files made by TAPECOPY are renamed and stored on MASS and ATL. Files containing the database dump are named HPTPyymmdd, where yy are the last two digits of the year, mm are the month (January is 01, December is 12), and dd are the day. This convention was established by R. C. Carlson. The letters stand for Hewlett Packard tape. The date is written on the tape reel as delivered and may differ from the yymmdd of the HPDB file the tape produces.

LICENSE - Each data collection device has a unique "license plate" of three letters and three numbers in that order. Licenses in other formats are not used by REVERT. Every datum is recorded along with its license.

LCM - Large core memory is an extension of CDC 7600 memory that is used for operand storage and I/O BUFFERS. It is slower than SCM (small core memory) when accessed randomly, but faster when consecutive words are read or written. Block copy instructions transfer data between memories very quickly.

LIST - This subroutine is a collection of nearly all of the printed output (WRITE) statements. They were localized in this manner to permit their use by more than one subroutine. List, or listing, is a computing term for printed output of a program. LIST does not contain error messages to the user's terminal. This subroutine was written by this author.

MAIN - The part of REVERT that is not contained in any subroutine is called MAIN. This section of the program is executed first, and it makes use of all other subroutines directly or indirectly.

NTS - Nevada Test Site is the location of the Spent Fuel Test - Climax.

PROCS - This subroutine's name is an abbreviation of the word process. The FIT subroutines must be used in a special order, which is controlled by PROCS.

REFERENCE, license or channel - Some FIT subroutines need data from other devices for their reconversion formulas. These are called reference channels or reference licenses. All data in the HPTP files is reference channel only, and is converted to licenses in a second pass over the DATABASE when it is being made by the GETCOEF subroutine.

REVERT - The name of the program being described, and the name of the file in the Livermore Time Sharing System in which it is stored. Variations of this name have suffixes, such as REVERT10A, which is a version number and its minor revision letter; and other letters such as L, which is the compiler listing file, and S, which is the compiler source file. The name REVERT was coined from re-convert, which means that data that has been converted (in an appropriate manner) to engineering units by the HP 1000 system is being converted again, this time using individual instrument material properties and calibration coefficients.

RVAL - This subroutine, written and named by R. C. Carlson, does the linear interpolation to return the value of a datum at a given time. Many FIT routines need data from devices other than the one whose data is being reconverted.

APPENDIX B - ACCESS TO PROGRAMS AND DATA

All programs and data are stored on the LTSS ENTERPRISE file-transport system using the file-management utility routine XPORT.[10] Ownership of the root directory, .SFT, is shared by users 138000, 348500, 393200, and 746394. Seventeen directories reside in .SFT; within them the necessary files can be found. Table 2 lists some of the directories, those associated with REVERT, and their contents. To get REVERT itself, for example, run XPORT with the following line:

```
XPORT RD .SFT:LIBS:SFTC40215!END          (! is the linefeed key.)
```

To extract REVERT from the library file SFTC40215, run LIB[3] with the following line:

```
LIB SFTC40215!X REVERT!END
```

Subsequent versions of REVERT or other programs will be stored in newer copies of the libraries, named by the date made. February 15th, 1984 is used as 40215, above. Use the L S. command with LIB to get the names of all files in a library file.

TABLE 2. Selected contents of the .SFT directory related to REVERT.

DIRECTORY	Type of files in Directory
B-FILES	Data files needing reconversion by REVERT
CODES	Programs and BCON controller files
HPDB	Database files containing conversion coefficients and references
HPTP	Database dump files used in making HPDB type files
LIBS	Library files. SFTSdate contain source files. SFTCdate contain executable programs. SFTLIBn contain subroutines needed for compiling and loading.
R-FILES	Data files reconverted by REVERT
REVREC	Listings of results of REVERT reconversion runs
RVRTLST	Listings of HPDB database files
RVTINT	ITEMNAME and ALTER files

APPENDIX C - SUBROUTINE LIBRARY FILES

REVERT consists of a short MAIN program and subroutines that are stored in compiled form in two library files: SFTLIB and ORDERLIB.[8] All of the subroutines in SFTLIB (and the MAIN part of REVERT as well) are compiled with CHAT since the sources are written in the FORTRAN dialect LRLTRAN[9], as are most of the subroutines in ORDERLIB. SFTLIB contains subroutines for use by the SFT-C project programs, while ORDERLIB is available for all users of the Livermore Computer Center.

Some of the communication between the major subroutines and the remainder of REVERT can be gleaned from Fig. 2, which is an excerpt from a listing of SFTLIB made by the library maintenance program LIBMAK.[12] The indented names in the ROUTINE column are "entry point" names within the ROUTINE at the top. The names also appear in the EXTERNAL SYMBOLS columns, but these columns also contain references to routines and entry points in ORDERLIB. Figure 3 is a cross-index of SFTLIB. Notice that some of the routines are not used by REVERT, although the MAIN routine of REVERT uses PROCS and GETCOEF. Figure 4 is a cross-index of common blocks used for data exchange between routines. Complete listings of all programs are provided on the attached microfiche.

ROUTINE	EXTERNAL SYMBOLS		COMMON BLOCKS	
ALTER				
ALTER	000005	00000D	1$$	3$$
ALTERS	000005	000002	5$$	
ALTERL	000002	ASSIGN		
ALTERT	000015	000010		
LISTCL	000007	000007		
	000008	000008		
	000014	LIST		
	IQ8SCH	000017		
	000013	SNBYT		
	LNBYT	EXIT		
	LISTCL			
BSRCH				
BSRCH				
BUFIN				
BUFIN	GETDUM	CONVI	DUFIN	QNAME789
GETDUM	DEVICE	BLOCKCOPY		
SKIPREC	FILL	000013		
FILL	000017	FINISH		
FINISH				
SKIPFILE				
REREAD				
DATE				
DATE				
FITCW				
FITCW			COM00	
FITES				
FITES			COM00	
FITFM				
FITFM	GCBFR	RVAL	COM00	
FITLD				
FITLD			COM00	
FITPN				
FITPN	XTOI		COM00	
FITRT				
FITRT	SQRTF		COM00	
FITRX				
FITRX	GCBFR	RVAL	COM00	
FITSD				
FITSD			COM00	
FITTC				
FITTC	GCBFR	RVAL	COM00	
	TEMPK			
FITVH				
FITVH	GCBFR	RVAL	COM00	
FITWT				
FITWT			COM00	
FITWX				
FITWX	GCBFR	RVAL	COM00	
FLT				
FLT	QLCOPY	XTOI		
GCBFR				
GCBFR	BSRCH	TYPE	1$$	2$$
	SHELF		3$$	6$$
			10$$	

Figure 2a Subroutines in SFTLIB, their interaction and use of COMMON blocks

ROUTINE	EXTERNAL SYMBOLS		COMMON BLOCKS	
GETCOEF				
GETCOEF	000005	00000D	1$$	2$$
LJUST	000005	GOB	3$$	4$$
	000002	000002	5$$	6$$
	ASSIGN	DEVICE	7$$	711$$
	000013	000017		
	LIST	LISTR		
	BUFIN	QLCOPY		
	KONVERT	TIMVERT		
	CONVI	EXIT		
	000015	*TYPE*		
	PUSH	QLBYT		
	QSBYT	TOJUL		
	IQ8SCH	WRITELCM		
	000014	SORTI		
	SORTD	GET		
	BSRCH	ALTERS		
	ALTERL	ALTERT		
	LJUST			
GETRCD				
GETRCD	QLCOPY	KFIX	G	
	FLT			
KFIX				
KFIX				
KONVERT				
KONVERT	QLCOPY	XTOI		
LIST				
LIST	000005	00000D	1$$	3$$
LISTR	000005	TYPE	5$$	6$$
	BLOCKCOPY	IQ8SCH	711$$	
	SORTI	SORTD		
NBCD				
NBCD	SNBYT			
NUMBR				
NUMBR	LNBYT	ITOJ		
POLYN				
POLYN	000005	000005		
PROCS				
PROCS	ASSIGN	000005	COM00	PD
	000005	*SHELFI*	1$$	2$$
	BSRCH	000013	3$$	4$$
	000017	READLCM	6$$	10$$
	TYPE	FITES		
	FITLD	FITPN		
	FITRT	FITSD		
	FITWT	000007		
	SHELF0	FITTC		
	WRITELCM	000014		
	FITRX	FITVW		
	FITWX	EXIT		
PUSH				
PUSH	LNBYT	SNBYT	PD	
POP				
GET				
RVAL				
RVAL				
SHELF				
SHELF	000013	000017	711$$	10$$
SHELFI	BLOCKCOPY	000014	2$$	4$$
SHELF0	000005	00000D		
	000005	EXIT		

Figure 2b Subroutines in SFTLIB, their interaction and use of COMMON blocks

ROUTINE	EXTERNAL SYMBOLS		COMMON BLOCKS
SORTI			
SORTI	BLOCKCOPY		
SORTD			
TABLMAK			
TABLMAK	000002	00000D	1$$
	000002	000015	
	000010	000007	
	000007	000008	
	000008	CONVI	
	000005	000005	
	EXIT		
TEMPK			
TEMPK			
TIMVERT			
TIMVERT			
TOJUL			
TOJUL	DATE		
TYPE			
TYPE			PD
VERIFY			
VERIFY			1$$

Figure 2c Subroutines in SFTLIB, their interaction and use of COMMON blocks

ENTRY PT:	CALLED BY:			
PROCS				
GCBFR	FITTC FITRX FITWX	FITVW FITFM		
TYPE	PROCS GETCOEF	GCBFR LIST		
TIMVERT	GETCOEF			
KONVERT	GETCOEF			
BUFIN	GETCOEF			
GETDUM	BUFIN			
SKIPREC FILL	BUFIN			
FINISH	BUFIN			
SKIPFILE REREAD				
FITWT	PROCS			
FITTC	PROCS			
TEMPK	FITTC			
FITES	PROCS			
FITRT	PROCS			
FITPK	PROCS			
FITLD	PROCS			
FITSD	PROCS			
FITVW	PROCS			
FITRX	PROCS			
FITFM				
POLYN				
RVAL	FITTC FITRX FITWX	FITVW FITFM		

ENTRY PT:	CALLED BY:	
FITCW		
GETCOEF		
LJUST	GETCOEF	
ALTER		
ALTERS	GETCOEF	
ALTERL	GETCOEF	
ALTERT	GETCOEF	
LISTCL	ALTER	
LIST	GETCOEF	ALTER
LISTR	GETCOEF	
DATE	TOJUL	
TOJUL	GETCOEF	
FITWX	PROCS	
SHELF	GCBFR	
SHELFI	PROCS	
SHELFO	PROCS	
SORTI	GETCOEF	LIST
SORTD	GETCOEF	LIST
BSRCH	PROCS GETCOEF	GCBFR
PUSH	GETCOEF	
POP GET	GETCOEF	
NBCD		
NUMBR		
GETRCD		
KFIX	GETRCD	
FLT	GETRCD	
TABLMAK		
VERIFY		

Figure 3 Subroutine use in SFTLIB

COMMON BLOCK:	USED BY:				
COM00	PROCS FITVW FITPN	FITHT FITRX FITLD	FITTC FITFM FITSD	FITES FITCW	FITRT FITWX
PD	PROCS	TYPE	PUSH		
1$$	PROCS TABLMAK	GCBFR VERIFY	GETCOEF	ALTER	LIST
2$$	PROCS	GCBFR	GETCOEF	SHELF	
3$$	PROCS	GCBFR	GETCOEF	ALTER	LIST
4$$	PROCS	GETCOEF	SHELF		
6$$	PROCS	GCBFR	GETCOEF	LIST	
10$$	PROCS	GCBFR	SHELF		
DUFIN	BUFIN				
QNAME789	BUFIN				
5$$	GETCOEF	ALTER	LIST		
7$$	GETCOEF				
711$$	GETCOEF	LIST	SHELF		
G	GETRCD				

Figure 4 COMMON block use by subroutines in SFTLIB

APPENDIX D - CONVERSION COEFFICIENT DATABASE FILES - "HPDB" FILES

REVERT needs one of these files to supply its ten reconversion subroutines with coefficients for their algorithms. Each "FIT" subroutine operates on raw data of its "device type." A list of device types may be found in Table 1, and a description of the process is given in the section titled MAJOR ROUTINES under PROCS.

The files are of constant length, 153,608 words, arranged as 1536 blocks of 100 words, plus an 8-word information block at the end. Individually the raw database dump files have no more than 1024 different licenses (or channels), the maximum permitted during any one data-collection period. However, the effect of merging several such files and the creation of duplicate licenses with ALTER, in order to have databases usable over a longer time-span, demands that they have a larger capacity. Data files usually have about 750 different licenses.

INFORMATION BLOCK - 8 words

word 1	The number of coefficient blocks in the body of the file.
words 2-3	Unused.
word 4	The name of the file at the time it was first created. It consists of "HPDB" followed by a 6-digit date (YY MM DD) that was translated from the date found in the raw dump file from which it was made, e.g., HPDB830428.
word 5	The name of the last raw dump file to be merged with this file, e.g., HPTP831024.
word 6	The time of day when REVERT last modified this file, e.g. 14:22:20.
word 7	The date when REVERT last modified this file, e.g., R 02/10/84.
word 8	The 6-digit date that was merged with the filename e.g., 831024.

DATA - COEFFICIENTS SECTION - 100 word blocks

words 1-12	These words have definitions common to all blocks.

word 1	LOGICAL CHANNEL NUMBER. An integer between 1 and 1024. These numbers are not unique here or in data files as presented to REVERT. They are unique at the time of data collection.
word 2	LICENSE PLATE. A six-character identifier, usually three-letters, three numbers, left justified ASCII. These are unique at data collection time and for HPDB file merging, but by using ALTER, duplicates may be inserted in the file. (See words 7-9.)
word 3	PHYSICAL DEVICE TYPE. A two-letter designation, left justified ASCII.
words 4-6	CHANNEL DESCRIPTOR. Up to 30 characters of text, in ASCII.
words 7-9	DATE DEVICE INSTALLED (7), REMOVED (8), MODIFIED (9). On the database dump (HPTP) tapes and files, these are ten-character ASCII values of the form YYMMDDHHMM (year, month, day, hour, minute). Here, they are Julian days minus the leading 24* in real, floating point format for fast comparison with times on data files. Duplicate licenses (word 2) each have unique times over which the associated coefficients are applicable.
words 10-12	DATA AGE LIMITS. Originally scan table data space in the HP files, these values are entered via the ALTER file for the FIT sub-programs, which must determine if reference data bracketing the datum being reconverted is within a useful time span. The values are in minutes, real, floating point.
words 13-68	Each device type FIT subprogram uses these spaces uniquely. Word three, above, determines the arrangement and definitions in this section, which are described for each of the ten physical device types below. The types of data will be labeled R for real, floating point, A for reference license ASCII, right justified and I for integer.

* Julian days in this project range from 2444240 to 2446055. The leading "24" is stripped from all Julian day numbers in this project.

	ES TYPE DEVICES
word 13	Excitation supply R
	LD TYPE DEVICES
word 13	Slope. R
word 14	Intercept. R
	PN TYPE DEVICES
words 13-17	Polynomial coefficients, constant term (13) through fourth power term (17). R
	RT TYPE DEVICES
word 13	Reference resistance R
word 14	Coefficient A R
word 15	Coefficient B R
	SD TYPE DEVICES
word 13	Offset constant R
	WT TYPE DEVICES
word 13	Slope coefficient R
word 14	Intercept R
	TC TYPE DEVICES
word 13	Resistance R (not used)
word 14	Quartic coefficient set I (always zero)
word 15	Reference license A
word 16	Offset constant R
	RX TYPE DEVICES
word 13	Excitation license A
word 14	Rod length, metres R (not used-see words 40-46)
words 15-23	Calibration positions 1-9 R
words 24-32	Calibration readings 1-9 R
words 33-39	Temperature reference licenses 0-6 A

words 40-46	Distances to temperature readings 0-6	R
words 47-53	Calibration temperatures 0-6	R
words 54-60	Physical component lengths 1-7	R
words 61-63	Thermal expansion values 1-3	R
word 64	CTE temperature adder	R
word 65	CTE temperature multiplier	R
word 66	Head temperature multiplier	R
word 67	Head temperature adder	R
word 68	Offset constant	R

VW TYPE DEVICES

word 13	IRAD gauge sensitivity	R
word 14	Zero load reading	R
word 15	Gauge-rock thermal coefficient	R
word 16	Initial temperature	R
word 17	Initial set gauge reading	R
word 18	Gauge temperature reference license	A

WX TYPE DEVICES

word 13	Excitation license	A
word 14	Wire length	R
word 15	Sensitivity	R
words 16-22	Calibration temperature	R
words 23-29	Temperature reference license	A
words 30-36	Temperature multiplier	R
words 37-43	Thermal expansion coefficient	R
word 44	Offset constant	R

APPENDIX E - ITEMNAME FILE

The ITEMNAME file is necessary because the data base dump files (HPTP files) cannot be relied on to deliver the correct combination of item names and coefficient block locations. The GETCOEF subroutine section describes that process.

Figure 5 is a listing of the latest version. The top and right-hand side of the listing contains comments that lie outside the 4(A6,1X,I3,4X) format area and are thus ignored by the program. Blank lines are also acceptable anywhere. Use them to make the file more readable.

Some of the locations are negative. The minus signs are simply markers which are needed on reference channel numbers being read from the database dump tapes. These reference channel numbers must be converted to reference licenses. All integer data need not be channel numbers. The conversion from channel numbers to licenses is done in a second pass over the HPDB database after data for the necessary table has been completely read.

An ALTER file may be appended to this file after first ending ITEMNAME use by an "END" starting in column one. If altering an HPDB database file (which does not need the use of an ITEMNAME file) the ALTER subroutine skips over all lines until it has passed the END.

```
                                                                IN13 4/26/83
                                                                MINUS SIGNS MARK LOCATIONS
                                                                OF REFERENCE CHANNELS
                                                                ---------
                                                                NAMES IN H-P FILE HAVE
                                                                6 CHARACTER MAXIMUM
LCNMBR   1     LCDESC   2     PDTYPE   3     LCDSTR   4
DATEIN   7     DATEOU   8     DATEMO   9     DVM      10          2
NRATE    11    TLAST    12

WTSLOP   13    WTINTR   14

TCOHMS   13    TCCOEF   14    TCREFC -15     TCOC     16          13

ESWLOC   13

RXESLC -13     RXRODL   14                                       4,13
RXKCP1   15    RXKCP2   16    RXKCP3   17    RXKCP4   18          4
RXKCP5   19    RXKCP6   20    RXKCP7   21    RXKCP8   22          4
RXKCP9   23                                                      4
RXKCC1   24    RXKCC2   25    RXKCC3   26    RXKCC4   27          4
RXKCC5   28    RXKCC6   29    RXKCC7   30    RXKCC8   31          4
RXKCC9   32                                                      4
RXTLC0 -33     RXTLC1 -34    RXTLC2 -35     RXTLC3 -36          4,13
RXTLC4 -37     RXTLC5 -38    RXTLC6 -39                         4,13
RXDTT0   40                                                      4
RXPCL4   41    RXPCL5   42    RXPCL6   43    RXPCL7   44          4
RXPCL8   45    RXPCL9   46                                       4
RXTOC0   47    RXTOC1   48    RXTOC2   49    RXTOC3   50          4
RXTOC4   51    RXTOC5   52    RXTOC6   53                        4
RXPCL1   54    RXPCL2   55    RXPCL3   56                        4
RXMTC1   61    RXMTC2   62    RXMTC3   63    RXMTC4   64          4
RXOC     68    RXCASE   69                                       4

RTR0     13    RTCA     14    RTCB     15

SDOC     13

LDSLOP   13    LDINTR   14

MDPRON   13    MDPAR1   14    MDPAR2   15    MDPAR3   16
MDPAR4   17    MDPAR5   18

VWSF     13    VWZLR    14    VWMM     15    VWTO     16          10  FOR DB > 4/12/82
VWR0     17    VWTCLC -18                                       10,13   DB > 4/12/82

WXESLC -13     WXNIRL   14    WXSEN    15                        9,13
WXTEMP1  16    WXTEMP2  17    WXTEMP3  18    WXTEMP4  19          9
WXTEMP5  20    WXTEMP6  21    WXTEMP7  22                        9        .......
WXTLC1 -23     WXTLC2 -24    WXTLC3 -25     WXTLC4 -26          9        .......
WXTLC5 -27     WXTLC6 -28    WXTLC7 -29                        9,13
WXMTY1   30    WXMTY2   31    WXMTY3   32    WXMTY4   33        9,12
WXMTY5   34    WXMTY6   35    WXMTY7   36                      9,12
WXTEC1   37    WXTEC2   38    WXTEC3   39    WXTEC4   40        9,12
WXTEC5   41    WXTEC6   42    WXTEC7   43                      9,12
WXOC     44                                                     9

PNC0     13    PNC1     14    PNC2     15    PNC3     16
PNC4     17

SYPRON   13    SYPAR1   14    SYPAR2   15    SYPAR3   16
SYPAR4   17    SYPAR5   18
```

Figure 5 The ITEMNAME file

APPENDIX F - HPTP LOGICAL FILE FORMATS

Database dump tapes from the HP 1000 system at NTS[2] are copied to CDC 7600 disk storage by the Utility Routine TAPECOPY[12]. The data are virtually useless in the copied form and must be translated to CDC 7600 word-length, BCD text and floating point format and arranged for quick reference during data reconversion. That is the major function of the subroutine GETCOEF. What follows is a collection of details about that subroutine and the files that it translates.

Some definitions and conventions are necessary here to minimize repetition and confusion. All word, bits or byte counts are decimal numbers. The CDC 7600 words are 60 bits. The HP 1000 words are 16 bits. HP bytes are 8 bits. CDC 7600 words will be capitalized; HP 1000 words will be lower case. Bit numbers in WORDS (words) are 60 (15) on the left (high order) and 0 on the right. CDC WORDS contain ten 6-bit characters (ASCII) and HP words contain two 8-bit characters (also called ASCII). To convert from an HP character to a CDC one, remove the left most two bits and subtract 40 (octal). See Table 3 for a comparison of binary, floating point number formats.

The files described below are all within one file on the CDC 7600. They were separate on the HP 1000 and on tape. (See the TAPECOPY section.)

File 1 Tape generation record - 12 words, 4 WORDS
bits 0-143: discarded (18 bytes)
bits 144-191: generation time (3 words double precision floating point) This a Julian day number without the leading 24. Noon GMT on Jan. 1, 1980 is 44 240.0.

File 2 Alarm message records - 100 - 52 WORDS each
These are skipped-over.

File 3 Database map file - about 275 records of 21 WORDS
This is an index to File 5, containing "item names" (names for coefficients and other data) that are in File 5. Only those names that can be found in the itemname file have their item numbers stored.
bytes 3-8 (words 2-4): itemname (left justified if fewer than six characters)

word 5: item number, the identifier in file 5
byte 14: datatype (R for real, floating point,
I for integer and X for character, ASCII).

File 4 Scan tables - 1024 9 WORD records
This data is now discarded due to its lack of predictability.
word 4, bytes 7, 8: DVM setting - two characters, copied to location 10
words 5, 6: Normal scan interval - One floating point number usually between 60 and 1440 (minutes) copied to location 11
words 17-19: Time of last scan, Julian day number, double precision, copied to location 12.
words 30-32, bytes 59-64: Logical channel name (license, six characters.
A table made during the reading of this file was stored by license and later retrieved as each coefficient block was being made (while reading File 5).

File 5 Database - up to 1024 records usually 40 WORDS each. There is one record for each license that was active at the time of this dump.
Word 1: number of items, N, in each of the next three segments in this record
Words 2 through N+1: item numbers of data in this record
words N+2 through 2N+1: lengths (in words) of items in next segment
words 2N+2 to end: coefficients, names, text, or reference channels.

TABLE 3. A comparison of floating point, binary numbers on the HP 1000 and the CDC 7600[4,5]. Bit 0 is at the right end of the word in both computers.

H P 1000	CDC 7600
word 1 bit 15 sign of fraction	WORD 1 bit 59 sign
bits 14-0 fraction	bit 58 exponent bias
word 2 bits 15-0 fraction, continued	bits 57-48 exponent
(double precision only)	bits 47-0 fraction
word 2 bits 15-8 fraction, concluded	
bits 7-1 exponent	
bit 0 sign of exponent	
(word 3 of double precision)	
2's complement negative	1's complement negative
Approximate Range: $\pm 10^{-38}$ to 10^{38}	Approximate range: $\pm 10^{-293}$ to 10^{322}
Accuracy: 6 or 7 decimal digits	Accuracy: 14 decimal digits
over 11 in double precision	

APPENDIX G - PROGRAM LIBRARY AND ASSOCIATED ROUTINES

REVERT needs data files that are tailored for it in format and length. Further processing, such as data point listing, storing and plotting are outside of REVERT's domain. Consequently, we have developed several programs (not Utility Routines) that are essential for all tasks associated with SFT-C data. They are listed here with a short description and sources of further help with their use; i.e., documentation or authors names. Program authors are listed chronologically; the last name is most likely to offer help, but all are working in Livermore at the time of this writing. Documents listed are only the latest on each subject.

PROGRAM	BREAD
PURPOSE	To read a family of data files made with TAPECOPY from a data tape, translate the data to CDC 7600 BCD (ASCII) and sort by license and time, and write a single file.
AUTHORS	R. A. Nyholm, J. Richter, P. E. Harben, G. L. Hage
DOCUMENT	BREAD, A CDC 7600 Program that Processes Spent Fuel Test - Climax Data, UCID-19814, APRIL 1983, G. Hage.
AVAILABILITY*	LIB files in .SFT:LIBS

* See Access to Programs and data section.

PROGRAM	BTOA
PURPOSE	To copy a binary data file to a BCD (ASCII) file, point for point, in one of three formats: 1) Identical with BREAD output (no reconverted data), 2) The same as above, but with reconverted data replacing H-P converted data, 3) All data in a more readable format (by eye). Other options, by last author, are in a new revision which is stored in the second location.
AUTHORS	P. E. Harben, G. L. Hage, R. C. Carlson
DOCUMENT	Binary File Listing with the BTOA Program, Version Three, SFT 83-69, December 1983, G. Hage.
AVAILABILITY	LIB files in .SFT:LIBS and .SFT:CODES

PROGRAM	DBCOMP
PURPOSE	To compare two HPDB databases and write a file of ALTER commands that would change the older into the newer.
AUTHOR	G. L. Hage
DOCUMENT	None - Self instructing use
AVAILABILITY	LIB files in .SFT:LIBS

PROGRAM	MFAB
PURPOSE	To convert BREAD output BCD file(s) to a binary data file with space for the reconverted data values (fifth word per data point). Also, to merge files of either type into a binary data file, with a time-range constraint. This program replaces ATOB.
AUTHORS	P. E. Harben (ATOB), G. L. Hage
DOCUMENT	File Merging and Conversion Program, MFAB, SFT 82-96, June 1982, G. Hage
AVAILABILITY	LIB files in . SFT:LIBS and .SFT:CODES

PROGRAM	PLOTCMP
PURPOSE	To plot a complete binary data file with data from a single license on each graph, by time. Raw, or either or both types of converted data may be plotted, using data or user controlled limits (windows).
AUTHORS	P. E. Harben, G. L. Hage

DOCUMENT	*PLOTCMP Revisions and New Operating Procedure*, SFT 82-169, November 1982, G. L. Hage
AVAILABILITY	LIB files in .SFT:LIBS
PROGRAM	PLOTR
PURPOSE	To plot data for a given license from several binary or BCD data files
AUTHOR	P. E. Harben
DOCUMENT	PLOTR Users' Guide, DM 80-71, December 1980, P. Harben
AVAILABILITY	LIB files in .SFT:LIBS
PROGRAM	SFIX
PURPOSE	To correct two classes of errors on early data tapes.
AUTHOR	R. C. Carlson
DOCUMENT	SFIX - Corrections to Early Data Tapes, AG 82-33, June 1982, R. C. Carlson
AVAILABILITY	.SFT:CODES
PROGRAM	UFO
PURPOSE	To process BREAD or MFAB BCD data files for final storage, using VFO, which needs Uniformly Formatted Output.
AUTHOR	D. N. Montan
DOCUMENT	Data Storage and Access, SFT-C, SFT 82-4, January 1982, L. B. Ballou
AVAILABILITY	L. B. Ballou or D. N. Montan
PROGRAM	VFO
PURPOSE	To invert a matrix of data files each containing data over a short time period, sorted by license, to a series of files containing data from each individual license, by adding the data to existing files or creating the files as necessary. The latter files are stored via XPORT.
AUTHOR	D. N. Montan
DOCUMENT	See UFO
AVAILABILITY	L. B. Ballou or D. N. Montan

PROGRAM	KURVSFT
PURPOSE	To plot data files created by VFO
AUTHOR	D. N. Montan
DOCUMENT	None - has reasonable help package. Type HELP for a short six-line message; type HELP again for 23 more lines. Type HELL first to get three pages of instructions.
AVAILABILITY	In Public Library file OJMLIB. To execute type OJMLIB KURVSFT.

APPENDIX H - ERROR MESSAGES

In the following text, lower case letters or words symbolize the contents of that part of the message to the user's terminal. User reactions are *suggestions, given the nature of the messages,* but not necessarily the user's wishes. The source of the message is in parenthesis. Other error messages may come from ORDERLIB^8 subroutines or the LTSS Operating System.

b-file NOT AVAILABLE (MAIN)

The binary data file has not been read by XPORT, or the name was *mistyped. Either get the file or change its name, then type in again.*

I/O ERROR ON B-FILE b-file AT location (SHELF)

No recovery is attempted after disk read/write errors. Restart REVERT. These errors are very rare, but the test must be made for quality assurance.

CANNOT CREATE file. CR TO RETRY, E TO END (GETCOEF)

Several reasons may account for not being able to create an HPDB database file. The operating system could not find enough space - wait, and try again. The name duplicates a file that already exists - *end, and fix your* filenames. Other obscure reasons are listed on page nine of reference eleven.

BUFIN ERROR READING ALARM MESSAGE n = ℓ (GETCOEF)

No recovery can be made. Try reading the file made by TAPECOPY again. This error has never occurred. The record number n is ℓ words long.

BUFFIN ERROR READING DATA BASE MAP RECORD n = ℓ (GETCOEF)

Same as above.

BUFIN ERROR READING SCAN TABLE n = ℓ (GETCOEF)

Same as above.

BUFFIN ERROR READING DATA BASE RECORD n = ℓ (GETCOEF)

Same as above.

REFERENCE CHANNEL LIST OVFLO FROM FILE file (GETCOEF, PUSH)

No recovery. During the loading of an ITEMNAME file, a table of reference channels had more than twelve entries per device. Device types are identified by the first two characters of itemnames.

ERROR WRITING file (GETCOEF)

No recovery. Restart the job. This error, writing an HPDB file, has never occurred.

ERROR READING file (GETCOEF)

No recovery. Restart the job. This error, reading from an HPDB file, has never occurred.

file CANNOT BE USED TO AUGMENT file (GETCOEF)

Two HPDB database files cannot be processed by REVERT, but an HPTP database dump file can be used to augment an HPDB database file. If a comparison of the two files is desired, use DBCOMP (in a SFTC library file). No recovery.

name NOT IN file (PROCS)

The license for the data point being reconverted is not in the HPDB database. This data point will be ignored with a -9999. inserted in the reconverted value. The HPDB database probably needs the new license. Compare its listing with the BREAD interface. This message is printed only once for each license.

dd NOT A DEVICE TYPE (PROCS, TYPE)

The table of device types in subroutine TYPE doesn't contain this data point's device type. The HPDB database contains the device type for this data point's license. This data point will be ignored with a -9999. inserted in the reconverted value. This message is printed only once for each license.

ERRORS RECONVERTING (PROCS)

n dd

If any of the FIT subroutines return an error flag to PROCS, it is counted in a total reserved for the appropriate device type. On completion, if any of the totals are non-zero, they are tabulated here.

I/O ERROR ON FILE name I = i (PROCS)

Disk read or write errors using a binary data file or read errors on an HPDB file cause this message. The i value is the index of the license in the sorted license table during passes 2 and 3; on pass one it counts 510-word steps through the binary data file.

APPENDIX I - TAPECOPY

TAPECOPY is an LTSS Utility routine (UR-403)[3] that is used for reading (or writing) magnetic tape of any format, density or parity. Our use reads five files written by the HP-1000 on 9-track units. No conversion of numbers or characters is done at this stage. (See GETCOEF subroutine and HPTP file formats.)

Dialog with TAPECOPY runs as follows:

```
User:   TAPECOPY 512 30720 / 5 5
        OK
User:   DBRPT T. (90) HPTP D. 5        (Read nine, oh.)
        OPTIONS OR CARRIAGE-RETURN
User:   Return key
        TYPE IN OCTAL LENGTH OF DISK FILE TO BE CREATED
User:   200,000
        WAITING ON TAPE ASSIGNMENT
        ALL TAPES ASSIGNED
        DOUBLE END-OF-FILES READ ON INPUT FILE.  CONTINUE?
User:   NO
        5 END-OF-FILES READ.  TYPE I, O, CARRIAGE-RETURN,
        END, OR OPTIONS
User:   END
         ALL DONE
```

A tape containing a database dump is given a name for identification at the LLNL Computer Center, Building 113. Any four or five-character name is adequate if the tape is to be retrieved within a week. We use DBRPT for this application, on line two above.

By default, the disk file is written in Monitor format. All records are separated by inter-record 60-bit words, which have 3 fields, right justified: 1) type - 6 bits, 01 for binary data, 00 for ASCII data; 2) length of previous record - 18 bits; 3) length of next record - 18 bits. The lengths count 60-bit words. For example, the first record in the HPTP file is 4 words long and the first word in the file is an inter-record word which, in octal, is 00000010 000000 000004. No marks in the disk file distinguish the different files that are on the tapes except for changing record lengths in the inter-record words.