

Technical Report No. 299
PROGRAM RANLIB: A DATA STORAGE AND
MANAGEMENT SYSTEM USER'S MANUAL

Jerry D. Peltz

Natural Resource Ecology Laboratory
Colorado State University
Fort Collins, Colorado 80523

GRASSLAND BIOME
U.S. International Biological Program

April 1976

TABLE OF CONTENTS

	Page
Title Page	i
Table of Contents	ii
Abstract	iv
Introduction	1
Data Trace Library Organization	1
Libraries Associated With Program RANLIB	2
RANLIB Instruction Set	3
Trace Identification Instructions	4
*NAME Instruction	4
*FIELD Instruction	5
*COMMENT Instruction	6
Library Creation/Modification Instructions	7
Adding Data Traces To A Library	8
Replacing Data Traces In A Library	10
Deleting Traces From A Library	12
Copying Data Traces From Library To Library	13
Merging Libraries	14
File Positioning Instructions	15
Library Retrieval/Output Instructions	18
Selecting Data Traces To Be Output	20
Formatting Output From A Selection	22
Separating Selected Data Traces With End Of File Marks	24
Using Program RANLIB	25
Example 1	27
Example 2	29
Example 3	32

TABLE OF CONTENTS (cont.)

	Page
Example 4	34
Example 5	36
Example 6	38
Example 7	39
Error Messages	41
*ADD	41
*COMMENT	42
*COPY	42
*DELETE	43
*FIELD	43
*FMTOUR	43
*LIST	44
*MERGE	44
*NAME	45
*ORDER	45
*READ	45
*REPLACE	45
*SELECT	46
*SKIPF	46
Acknowledgments	47
Literature Cited	48

ABSTRACT

Program RANLIB is a data storage and management system designed to assemble homogeneous data into time-trace format for storage in data libraries. The program instruction set allows a user to define input data to the system, create new data trace libraries and modify existing libraries. The organization of the data trace libraries is in a format that is acceptable to a plotting package which may receive time-trace information from SIMCOMP simulation runs, thereby making it possible to graphically compare computer simulated parameters and actual field collected data. This document presents the RANLIB instruction set, explains the use of the program, and lists error diagnostics that may be written during execution of the program.

INTRODUCTON

The Random Access Data Library program RANLIB was designed to build time-trace data libraries, to modify and maintain data trace libraries, and to select data traces from libraries for sequential output. A particular data trace library organization was selected for compatibility with a generalized plotting package (Stevens, in prep.) which could be interfaced with the SIMCOMP simulation system (Gustafson and Innis 1973). With this organization, actual field collected data and simulated output from SIMCOMP models could be compared graphically.

DATA TRACE LIBRARY ORGANIZATION

A data trace library consists of a collection of as many as 500 uniquely defined data traces. Each data trace in a library is referenced by its "trace name." A data trace consists of a header record, which defines the trace, and a number of data points. Data trace header records are organized into the following subfields:

1. Trace name subfield which contains the unique "trace name" of the data trace. A trace name may contain from 1 to 20 characters. The name must begin with a letter (A-Z) and may contain the characters A-Z, 0-9, blank, (, comma and). A comma is legal only when enclosed in parentheses as in subscripted names. For example:

DATATRACE(1)

X(1,3)

PAWAGH1975TRT5REP1

are legal data trace names.

2. Data points subfield which contains the number of data points in the trace. A data trace must contain at least 1 data point and may contain as many as 366.

3. Coordinates subfield which contains the number of coordinate values in a data point. If a data point contains 3 coordinate values, the coordinate subfield would contain the number 3. Each data point may contain from 2 to 6 coordinate values.

4. Coordinate name subfield which contains the names associated with each coordinate value. Initially these names are assumed to be "TIME," "VALUE," "FIELD3," "FIELD4," "FIELD5," and "FIELD6," but may be changed by the user when creating a trace. Coordinate names may contain the characters A-Z and 0-9 but must begin with a letter. Only those names associated with coordinate values defined by the user may be referenced. For example, if the user has defined his coordinates to be JDAY, MEAN, and STDERR; he cannot reference FIELD4...FIELD6.

5. Comment subfield which may contain commentary information about the data trace. The characters A-Z, 0-9, and blank are allowed in the comment subfield. As many as 80 characters may be included in a comment.

LIBRARIES ASSOCIATED WITH PROGRAM RANLIB

As stated in the previous section, data traces are referenced by their "trace name." Analogous to a trace name is a "library name" which is used to reference a particular library associated with the program. Three libraries are associated with program RANLIB during execution; they are MASTER, ALTERNATE, and RANOUT. Existing libraries may be associated with the library names by attaching them prior to execution

of the program.[†] Libraries created during execution are associated with the library name referenced when data traces were entered in the library.

RANLIB INSTRUCTION SET

Data trace libraries are created, modified, or retrieved by informing the program of specific actions to perform with instructions that are recognized by RANLIB. Using the appropriate instructions, the user may define data traces, create data trace libraries or add data traces to existing libraries, copy data traces from one library to another, and retrieve data traces from a library for sequential output. The general form of a RANLIB instruction is:

*<keyword>,<parameterlist>.

The asterisk must be in column 1 of the instruction followed by a keyword corresponding to the action requested. The parameter list further delimits the action requested by defining libraries to be used, trace names, and logical files on which to read or write data. The instruction set may be subdivided according to the general function of the instruction.

1. Trace identification instructions.
2. Library creation/modification instructions.
3. File positioning instructions.
4. Data selection/output instructions.

The instruction set will be explained by functional subset.

[†] See "Using Program RANLIB," page 25.

TRACE IDENTIFICATION INSTRUCTIONS

The trace identification instructions *NAME, *FIELD, and *COMMENT are used by program RANLIB to define the header record of a data trace and to organize data for input. These instructions are required before data traces may be added to a library.

*NAME Instruction

The *NAME instruction is used to define the data trace name. The instruction has two forms:

```
*NAME,<tracename>.
```

```
*NAME,<col1>-<col2>.
```

The first form, *NAME,<tracename>. explicitly defines the 1 to 20 character name of the next data trace to be added to a library. If this form is used, the data for each trace must be preceded by an explicit *NAME instruction. An alternate form, *NAME,<col1>-<col2>., informs the program that a trace name may be read from the first record of the input data used to assemble a trace. This instruction is in effect throughout execution of RANLIB unless another *NAME instruction is encountered.

If a homogeneous set of data records for several traces were to be used to create a library, the following instruction would have to be used only once to define the name of each trace.

```
*NAME,1-18.
```

If the trace name were not located on an input record, each trace would have to be named explicitly as in the following example.

```
*NAME,TRACENUMBERONE.
```

If the column range or the explicit name exceeds 20 characters, an error message will be written.

*FIELD Instruction

The number of coordinate values in each data point in a trace is defined by *FIELD instructions. The *NAME instruction must be followed by as many *FIELD instructions as the user needs to define a data point with the restriction that there must be at least 2 and may not be more than 6. The forms of the *FIELD instruction are:

*FIELD(<n>),<col1>-<col2>,<name>.

*FIELD(<n>),<col1>-<col2>.

The <n> defines the position in a data point a coordinate value will assume. For example, if n=1, the instruction is describing the first coordinate value in a data point. <col1>-<col2> informs the program the inclusive columns on an input record that the coordinate value can be read. The <name> parameter is optional; when present it defines the 1 to 10 character name of a coordinate field. If <name> is absent, the coordinate field name is defaulted to a predetermined name that depends on the coordinate position being defined. Default names are TIME, VALUE, FIELD3, FIELD4, FIELD5, and FIELD6. Examples and an explanation of *FIELD instructions follow.

*FIELD(1),36-42,JULIAN.

*FIELD(2),45-50,MEAN.

*FIELD(3),53-57,STDERR.

The above example defines 3 coordinate values for each data point in a trace. The coordinate names are "JULIAN," which occupies the first position in a data point, "MEAN," which is the second coordinate and "STDERR," which is the third. The values for JULIAN are to be read from columns 36 to 42 on an input record, 45 to 50 for MEAN, and 53 to 57 for STDERR. Default fields FIELD4 through FIELD6 are undefined.

*INDEP,20-30,DATE.

*DEP,40-50,BIOMASS.

*FIELD(3),60-70, STDEV.

*FIELD(4),10-15,NOBS.

In the preceding example, two alternate forms of the *FIELD instruction are used. The instruction *INDEP may be used to replace the *FIELD(1) instruction. As the keyword implies, *FIELD(1) and *INDEP are used to define the independent coordinate in a data point. *FIELD(2) may be replaced by *DEP. The second coordinate is assumed to be the major dependent value in a data point. As with the implicit form of *NAME, the *FIELD instructions will remain in effect throughout execution of the program or until other *FIELD instructions are encountered.

*COMMENT Instruction

The last trace identification instruction is the *COMMENT instruction which has the form:

*COMMENT,<commentarytext>.

or

*COMMENT,<col1>-<col2>.

The first form of the instruction is an explicit comment where <commentarytext> is a user written comment of 1 to 80 characters (characters A-Z, 0-9, blank) which will be inserted into the comment field of a data trace. The implicit form, *COMMENT,<col1>-<col2>., will inform the program to read a 1 to 30 character comment from the first record of a data set. If the explicit *COMMENT instruction is used, an 80 column comment would extend to another card. In this case, if the comment extends to the next card, the characters */ must be in columns 1 and 2 of the continuing card.

Some examples of data trace definitions follow:

```
*NAME,1-18.  
*FIELD(1),20-27,SAMPLENO.  
*FIELD(2),30-40,AVEBIOMASS.  
*FIELD(3),45-53,STDEV.  
*FIELD(4),19-20,NUMOBS.  
*COMMENT,AGHB PAWNEE 1970 G/M2.  
  
*NAME,AGMACARTHOTT72.  
*INDEP,20-30.  
*DEP,35-42.  
*COMMENT,58-80.
```

In the second example, the independent coordinate name is defaulted to TIME, the dependent coordinate name is defaulted to VALUE. A comment for the trace can be read from columns 58-80 on the first data card in the deck for the trace.

LIBRARY CREATION/MODIFICATION INSTRUCTIONS

In the previous section, trace definition instructions were explained. Once a trace or group of traces have been structurally defined, a library may be created or modified by adding traces to it. Other library creation/modification instructions allow the user to replace existing traces with new ones, create new libraries from existing libraries or modify existing libraries by transferring traces from other libraries. The three libraries available to the user are MASTER, ALTERNATE, and RANOUT which were mentioned in the library organization section of this manual. The library or libraries which will be involved in an action by

the program in response to a creation/modification instruction may be named in the parameter list of an instruction. However, each instruction implicitly assumes the libraries that will be operated on unless explicitly named by the user. Furthermore, the program assumes the source of the records that will be used to build a data trace unless specified by the user. The input sources recognized by the program are INPUT and ALTINPUT.

ADDING DATA TRACES TO A LIBRARY

Assume the user has defined the structure of a data trace he wants to add to a library with the appropriate trace identification instructions, the *ADD instruction will assemble the data trace and enter it in a library. The instruction has the form:

```
*ADD,<library>,<source>.
*ADD, , <source>.
*ADD,<library>.
*ADD.
```

For example:

```
*ADD,MASTER.
```

will add a data trace to the library called MASTER. The input cards containing the values for the trace must immediately follow the *ADD instruction.

```
*ADD.
```

has the same effect as the previous example.

```
*ADD,ALTERNATE,ALTINPUT.
```

will assemble the data cards residing on ALTINPUT into a data trace and enter the trace in library ALTERNATE.

```
*ADD, , ALTINPUT.
```

will assemble data from ALTINPUT and add the trace to library MASTER (default since a library is not explicitly named). Notice the double comma which must be included when defaulting parameters that are not explicitly defined in the instruction.

Assume the user has generally defined the data traces he wants to add to a library (*NAME,<col1>-<col2>. along with appropriate *FIELD instructions and *COMMENT) and more than one data trace is to be added to the library. Each set of data cards comprising a data trace could be preceded with a *ADD instruction or alternatively, the form:

*ADD,<library>,<source>,<n>.

could be used to assemble and add all the traces to a library. For example:

*ADD,MASTER,ALTINPUT,9.

would add the 9 data sets residing on ALTINPUT to library MASTER.

*ADD,9.

would add 9 data sets following the *ADD instruction to library MASTER. However the data sets must be separated by an end-of-record card (7-8-9 punched in column 1 of a data card) or a *EOR instruction which is recognized by the program as an end-of-record when used in conjunction with a library creation/modification instruction.

*ADD,3.

data cards for 1st trace

*EOR. or 7-8-9 punch in column 1

data cards for 2nd trace

*EOR. or 7-8-9 punch in column 1

data cards for 3rd trace

or,

```
*ADD,ALTERNATE,3.
```

where the alternate input file contains,

```
data cards for 1st trace
```

```
*EOR. or 7-8-9 punch in column 1
```

```
data cards for 2nd trace
```

```
*EOR. or 7-8-9 punch in column 1
```

```
data cards for 3rd trace
```

are correct deck set-ups for multiply adding data traces to a library.

Notice the first instruction where the repeat parameter immediately follows the *ADD. Since a file name or library name must be alphabetic the repeat parameter is recognized without intervening commas because it is numeric. However it must follow any explicit names in an instruction.

REPLACING DATA TRACES IN A LIBRARY

The *REPLACE instruction allows a user to eliminate erroneous data traces and replace them with correct ones. The form of the instruction is:

```
*REPLACE,<library>,<source>,<n>.
```

This instruction is identical to the *ADD instruction except that a data trace with the same name on a library will be replaced by the newly assembled trace. Examples:

```
*REPLACE,MASTER.
```

```
data cards for trace.
```

```
*REPLACE.
```

```
data cards for trace.
```

```
*REPLACE,,ALTINPUT.
```

The above sequence of cards will replace 2 data traces on library MASTER with traces assembled from cards immediately following the *REPLACE instructions. One other trace on library MASTER will be replaced using cards on file ALTINPUT.

*NAME,1-18.

*FIELD(1),20-23,JDAY.

*FIELD(2),25-35,MEAN.

*FIELD(3),40-45,STDERR.

*ADD,MASTER.

cards for data trace to be added to MASTER library

*ADD,ALTERNATE,3.

cards for data trace to be added to ALTERNATE library

*EOR.

cards for data trace to be added to ALTERNATE library

*EOR.

cards for data trace to be added to ALTERNATE library

*REPLACE.

cards to replace data trace on MASTER library

*NAME, PAWAGH1972TRT5.

*REPLACE,ALTERNATE.

cards to replace trace named "PAWAGH1972TRT5" on ALTERNATE library

The above example defines the data traces as stated with the trace identification instructions. One trace will be added to library MASTER from cards following the *ADD instruction. Three traces will be added to library ALTERNATE. One trace will be replaced on library MASTER; the name of the trace will be read from columns 1-18 of the first card following the *REPLACE instruction. The next trace to replace on library

ALTERNATE is explicitly named "PAWAGH1972TRT5". The data cards immediately follow the *REPLACE instruction. Other traces could not be added or replaced unless another *NAME instruction redefines the explicit name or defines the columns from which a name could be read.

DELETING TRACES FROM A LIBRARY

Data traces may be deleted from a library with the *DELETE instruction. Trace identification instructions need not be in effect because the trace is eliminated from the library and not replaced. The form of the instruction is:

```
*DELETE,<library>,<tracename>.
```

or

```
*DELETE,,<tracename>.
```

<library> in the second instruction is assumed to be MASTER.

```
*DELETE,ALTERNATE,PAWMACARTH72.
```

would delete the trace named "PAWMACARTH72" from library ALTERNATE.

```
*DELETE,MASTER,GWOO10PAWESATRTD.
```

```
*NAME,GWOO10PAWESATRTD.
```

```
*FIELD,20-23,DATE.
```

```
*FIELD,25-35,AVECMWTR.
```

```
*FIELD,36-45,STDEV.
```

```
*ADD,MASTER.
```

cards for data trace

In the above example the *DELETE and *ADD could be omitted and replaced by a *REPLACE instruction following the trace identification instructions.

The three instructions, *ADD, *REPLACE, and *DELETE, are primarily library creation instructions because they cause sequential input data

to be assembled into data trace format and entered in a library. After a library or many libraries have been created, the instructions *COPY and *MERGE may be used to create other libraries which are subsets or combinations of "parent" libraries. The resulting library is assumed to be RANOUT. However explicit use of library names by the user can cause any library to be the output library.

COPYING DATA TRACES FROM LIBRARY TO LIBRARY

Specific data traces may be copied from one library to another by instructing RANLIB to

```
*COPY,<sourcelibrary>,<targetlibrary>,<tracename>.
```

For example:

```
*COPY,MASTER,RANOUT,PAWBGH1974.
```

will cause the trace "PAWBGH1974" to be selected from library MASTER and entered on library RANOUT. Either library field may be defaulted by enclosing an empty field in the instruction with commas.

```
*COPY,,,AGHTRT172SJ.
```

will copy the trace "AGHTRT172SJ" from MASTER (default for the source library) to RANOUT (default for the target library).

By omitting the trace name parameter in a *COPY instruction, the user can copy the entire source library to the target library. Care must be taken, however, not to exceed the limit of 500 traces the target library may contain.

```
*COPY,ALTERNATE,RANOUT.
```

will copy the entire library ALTERNATE to the library RANOUT.

```
*COPY.
```

will copy the entire library MASTER to RANOUT.

*COPY,ALTERNATE.

*COPY.

will merge the libraries ALTERNATE and MASTER by copying each one to library RANOUT. However if ALTERNATE or MASTER contain identical trace names, an error will occur since trace names must be unique.

MERGING LIBRARIES

In the previous example, libraries MASTER and ALTERNATE were copied successively to RANOUT, effectively merging the two and placing the result on a target library. The *MERGE instruction has the form:

*MERGE.<sourcelibrary1>,<sourcelibrary2>,<targetlibrary>.

Any or all library name parameters may be defaulted by enclosing an empty field with commas or with:

*MERGE.

which will merge libraries MASTER and ALTERNATE and enter the result in RANOUT. Examples:

*MERGE,,,RANOUT.

*MERGE,ALTERNATE,MASTER.

*MERGE,ALTERNATE,RANOUT,MASTER.

In the last example above, each library must be explicitly named to avoid errors. If a library name were omitted, an attempt to enter duplicate traces on the target library would be detected. For example,

*MERGE.

defaults to,

*MERGE,MASTER,ALTERNATE,RANOUT.

Defaulted library names are position dependent in the instruction.

FILE POSITIONING INSTRUCTIONS

The file positioning instructions *READ, *SKIPF, *SKIPR, and *REWIND allow the user to position the data trace input files and change the input file from which program instructions will be read.

The *READ instruction informs the program the input device to read instructions and data from

*READ,<inputfilename>.

The input files are named INPUT and ALTINPUT. This instruction provides the user with a convenient way of creating large libraries where it might otherwise be a very tedious job. For example, a user who wants to summarize a large amount of data from several sites for several years, treatments, and replicates and later store the summary information in a time-series trace library, may produce an output file in the summarization program which looks like:

```
*ADD,MASTER.  
    data trace input cards  
  
*ADD,MASTER.  
    data trace input cards  
.  
.  
.  
  
*ADD,MASTER.  
    data trace input cards  
  
*READ,INPUT.
```

The output file above could then be saved on tape or another storage device and used as input to RANLIB later by associating the tape or storage device with ALTINPUT (University Computer Center 1975a). The input instructions which would cause the data to be assembled as data traces and entered in library MASTER are:

```
*NAME,1-18.  
*INDEP,21-30,JDAY.  
*DEP,31-40, MEAN.  
*FIELD(3),41-50,STDERR.  
*FIELD(4),51-55,NOBS.  
*READ,ALTINPUT.
```

The above example would define the data traces to be assembled and instruct RANLIB to read from ALTINPUT. ALTINPUT contains *ADD instructions and data cards used to assemble and enter data traces. The final instruction on ALTINPUT is "*READ,INPUT.," to instruct the program to resume reading from the standard input device. This instruction would not be necessary if further processing of instructions/data was not necessary. Execution would stop after running out of input. Care must be taken, when generating data sets and instructions with summarization programs, not to exceed the capacity of a library. A good rule to follow is to count data trace sets and limit them to 500 or change the library name on the *ADD instruction after 500 data sets have been generated. For example, ALTINPUT could be structured like the following for a large data set.

```
*ADD,MASTER.  
    data cards for 1st trace.  
    .  
    .  
    .  
*ADD,MASTER.  
    data cards for 500th trace.  
*ADD,ALTERNATE.  
    data cards for 501st trace  
etc.
```

The *SKIPF instruction is used to skip logical files on ALTINPUT. For example, the input to RANLIB which created a particular data trace library could be saved as a backup on magnetic tape in case the library is destroyed. If many library backups are saved on the same magnetic tape, the *SKIPF instruction could be used to position the tape to the desired point to recreate the library.

Assume a magnetic tape has the following structure:

File 1 contains instructions and data necessary to create a trace library of aboveground herbage data
logical end-of-file mark

File 2 contains instructions and data necessary to create a trace library of belowground herbage data
logical end-of-file mark

File 3 contains instructions and data necessary to create a trace library of gravimetric soil water data.
logical end-of-information mark

If the library containing the gravimetric soil water data was inadvertently destroyed by a user or a computer system malfunction, the following instructions would enable RANLIB to recreate the library. The magnetic tape must be associated with ALTINPUT.[†]

*REWIND,ALTINPUT.

*SKIPF,ALTINPUT,2.

*READ,ALTINPUT.

The *REWIND instruction has the form:

*REWIND,ALTINPUT.

[†]See "Using Program RANLIB," page 25.

and is used to position ALTINPUT to the beginning of the associated tape.

The *SKIPF instruction has the form:

*SKIPF,ALTINPUT,<n>.

where n refers to the number of files to be skipped. In the example, the third file on ALTINPUT is the one to be used to recreate a library so two files will have to be skipped to position it to the proper place. If the repetition factor <n> is omitted from the *SKIPF instruction, its value is defaulted to 1.

LIBRARY RETRIEVAL/OUTPUT INSTRUCTIONS

The contents of existing data trace libraries may be printed to the standard output device by instructing program RANLIB to:

*LIST,<library>,DETAIL.

where <library> may be MASTER, ALTERNATE, or RANOUT. The result of this instruction is a list of the contents of the library, including each trace name, the number of points in the trace, the number of coordinate fields and the first 30 characters of the comment field. If the keyword "DETAIL" is omitted, a listing of only the trace names in a library is printed. For example:

*LIST,MASTER,DETAIL.

will print all the information pertinent to each trace in library MASTER.

*LIST,MASTER.

will print only the names of data traces which reside in MASTER. A listing of all libraries associated with the program during execution may be obtained by the following instructions:

*LIST,,DETAIL.

*LIST.

The first example above will print a detailed list of all traces for each library which was created or attached to the program during a run. The second example will have the same effect for each library except only trace names will be printed.

A listing of trace information obtained by a *LIST instruction alone is not sorted because of the technique used to enter data traces into the library. A sort feature is important since it allows the user to organize a list of trace names contained in a library by "keyfields" or subdivisions of the trace name. More importantly, the *ORDER instruction enables the user to organize trace names in a library and select a range of traces to be output to a sequential output file. The form of the *ORDER instruction is:

*ORDER,<keyfield>,<keyfield>,...,<keyfield>.

where <keyfield> is defined as a column number or a range of columns.

<keyfield>=<col>,or<col1>-<col2>.

The maximum number of keyfields which may be used for ordering trace names in a library is five. If the instruction

*ORDER.

is given, a default keyfield of 1-19 will be used for sorting the trace names. For example, assume the data trace library attached to MASTER contains aboveground herbage data from several different collection sites for different years, treatments and replicates. The trace name for each trace is organized according to the following:

Columns 1-2 contain data type code "AH"

Columns 3-12 contain genus/species code of plant collected

Columns 13-14 contain collection site code

Columns 15-16 contain year of collection

Column 17 contains treatment code

Column 18 contains replicate code

The user may instruct RANLIB to print a listing of library MASTER sorted by site, year, treatment, replicate, and genus/species with the following instructions.

*ORDER,13-14,15-16,17,18,3-12.

*LIST, MASTER, DETAIL.

Once encountered, a *ORDER instruction remains in effect for all libraries for the computer run unless changed by another *ORDER instruction.

SELECTING DATA TRACES TO BE OUTPUT

Data traces may be written in sequential format to an auxiliary output device by using the *SELECT instruction. This instruction allows the user to select individual data traces, a range of traces, or a combination of individual traces and ranges of traces. The instruction has the form:

```
*SELECT,<library>,<selectiondelimiter>,  
<selectiondelimiter>,...,<selectiondelimiter>.
```

where <library> may be MASTER, ALTERNATE, or RANOUT.

```
<selectiondelimiter>=<tracename> or
```

```
<tracename>-<tracename>
```

For example, assume the data traces on library MASTER have been sorted into the following order:

AGINVB	ACAR	8701
AGINVB	ACAR	8705
AGINVB	ACAR	8711
AGINVB	ACAR	8715
AGINVB	ACAR	8721
AGINVB	ACAR	8725
AGINVB	ARAN	8701
AGINVB	ARAN	8705
AGINVB	ARAN	8711
AGINVB	ARAN	8715
AGINVB	ARAN	8721
AGINVB	ARAN	8725

The data trace names are organized according to the following codes:

Columns 1-6	Data type code "AGINVB" aboveground invertebrates
Columns 8-11	Genus/species code
Column 14	Site code
Columns 15-16	Year
Column 17	Treatment code

The following instruction will select all data traces with genus/species code ACAR.

```
*SELECT,MASTER,AGINVB ACAR 8701-AGINVB ACAR 8725.
```

All treatment 1 genus/species ARAN data traces may be selected with the instruction:

```
*SELECT,,AGINVB ARAN 8701,AGINVB ARAN 8711,AGINVB ARAN 8721.
```

Genus/species ACAR and ARAN for 1970 and 1971 may be selected with the following:

```
*SELECT,,AGINVB ACAR 8701-AGINVB ACAR 8715, AGINVB ARAN  
8701-AGINVB ARAN 8715.
```

One example of selecting all the data traces for output would be:

```
*SELECT,MASTER,AGINVB ACAR 8701,AGINVB ACAR 8705-  
AGINVB ARAN 8715,AGINVB ARAN 8721,AGINVB ARAN 8725.
```

A *SELECT instruction may contain any number of individual trace names or trace name range delimiters. If the *SELECT instruction must be continued to another card, the characters */ must occupy columns 1 and 2. However, a trace name that must be continued to another card must be exactly as it was defined. The portion of the name that is on the first card must extend to column 80 before the instruction is continued to another card.

FORMATTING OUTPUT FROM A SELECTION

Data traces selected for output are written as a sequential data file with one record written for each data point in the trace. The default format for data trace output follows:

Columns 1-20	Trace name in 2A10 format
Columns 21-30	Field 1 value of data point F10.4 format
Columns 31-40	Field 2 value F10.4 format
Columns 41-50	Field 3 value F10.4 format
Columns 51-60	Field 4 value F10.4 format
Columns 61-70	Field 5 value F10.4 format
Columns 71-80	Field 6 value F10.4 format

As an example of the default output assume a selected trace has three coordinate fields, TIME, MEAN, and STDERR. An output record would have the following format.

Columns 1-20	Trace name
Columns 21-30	Value for TIME
Columns 31-40	Value for MEAN
Columns 41-50	Value for STDERR

Since there were no other fields defined, the remaining columns of each record would be blank.

The user may explicitly format output by preceding a *SELECT instruction with a *FMTOUR instruction.

*FMTOUR,<outputdescription>,<outputdescription>,...,<outputdescription>.

<outputdescription> may have any of the following forms:

```
NAME/<extractionlist>/  
COMMENT/<extractionlist>/  
<fieldidentifier>/<formatspecification>/
```

where:

<extractionlist> may be a column number, and/or a range of columns and/or a skip specification or a combination of column numbers, ranges, and skips. For example consider the following trace name and partial *FMTOUR instruction.

```
THIS IS A TRACE NAME  
*FMTOUR,NAME/6-7,1X,1-4,1X,9-20/,...
```

The above *FMTOUR instruction would specify the sequence of characters which would be extracted from the name and inserted into the first 20 columns of an output record. The first 20 characters would be:

```
IS THIS A TRACE NAME
```

A trace name output description may describe as many as 20 columns to be printed. Comment output descriptions may describe as many as 30 columns to be printed.

```
*FMTOUT,NAME/1-20/,...,COMMENT/1-30/.
```

will inform RANLIB to print the entire trace name in columns 1-20 of a sequential output record and to print 30 columns of the comment field at the end of the record.

The output description <fieldidentifier>/<formatspecification>/ informs RANLIB the particular coordinate value to print on an output record. <fieldidentifier> may be the field name or the field position within a data trace point. <formatspecification> is a FORTRAN (Control Data Corporation 1973) format phrase. For example:

```
*FMTOUT,NAME/1-20/,JDAY/I3,1X/,MEAN/F10.4/.
```

will format the output of a *SELECT instruction as follows:

Columns 1-20 will contain the trace name

Columns 21-23 will contain the value of the field JDAY

Columns 25-34 will contain the value of the field MEAN

The rest of the output record will be blank since no other fields were specified in the *FMTOUT instruction. An alternate way of specifying the coordinate value to be printed is:

```
*FMTOUT,NAME/1-20/,JDAY/I3,1X/,2/F10.4/,3/F10.4/.
```

In the above example the <fieldidentifier> for the second and third coordinates in a data point is their respective position in the data point. Either the field name or the field position may be used in a *FMTOUT instruction to reference a coordinate field.

SEPARATING SELECTED DATA TRACES WITH END OF FILE MARKS

When selecting data traces for output to a sequential file, it may be desirable to separate data traces with an end-of-file mark. The instruction:

*EOR.

will cause RANLIB to write an end-of-file on the sequential output device. For example:

```
*FMTOUT,NAME/1-20/,1/I4/,2/F7.2/,3/F7.2/,COMMENT/1-30/.
```

```
*SELECT,AGINVB LEPI 8711.
```

```
*EOR.
```

```
*SELECT,AGINVB LEPI 8715.
```

```
*EOR.
```

```
*SELECT,AGINVB ISOP 8701.
```

The *EOR instructions will cause an end-of-file mark to separate each data trace on the sequential output device.

USING PROGRAM RANLIB

RANLIB is available as a permanent file on the CDC 6400 computer at Colorado State University. The program may be attached to a user's job with the following SCOPE (University Computer Center 1975a,b) system control card:

```
ATTACH,lfn,RANLIBLGOPRG,ID=JDP,MR=1.
```

where *lfn* is a local file name defined by the user, typically LGO.

The data trace libraries MASTER, ALTERNATE, and RANOUT are initialized by the program unless attached to the user's job prior to execution of RANLIB. Each library is associated with a particular logical file within the program. MASTER is associated with TAPE1, ALTERNATE with TAPE2, and RANOUT with TAPE3. The logical files TAPE1, TAPE2, and TAPE3 cannot be requested as magnetic tapes. A data trace library that is to be saved for future use must be requested as a permanent disk file prior to execution of RANLIB by the following SCOPE system control card:

REQUEST,*lfn*,*PF.

where *lfn* is TAPE1, TAPE2, or TAPE3 depending on which library is to be saved. Each library that is to be a permanent disk file must have a REQUEST control card. After execution, each library that is to be saved will have to be cataloged using the following control card:

CATALOG,*lfn*,*pfname*,ID=*id*,RP=*rp*.

where *lfn* is TAPE1, TAPE2, or TAPE3 as explained before. A library that was created and saved as a permanent file during an earlier run may be attached to a logical file pertaining to MASTER, ALTERNATE, or RANOUT by:

ATTACH,TAPE2,*pfname*,ID=*id*.

TAPE2 in the above SCOPE control card refers to library ALTERNATE. TAPE1 or TAPE3 may be used to refer to libraries MASTER or RANOUT, respectively.

The alternate input file ALTINPUT is associated with the logical file TAPE4. TAPE4 may be either a magnetic tape or a SCOPE system file. For example:

REQUEST,TAPE4,VSN=*vsn*,READ.*owner*.

would associate the magnetic tape *vsn* with ALTINPUT.

ATTACH,TAPE4,*pfname*,ID=*id*.

would associate the permanent file *pfname* with ALTINPUT.

The output file to which retrieved data traces are written is TAPE7. This file may be a magnetic tape or a SCOPE system file. If the selected output is to be saved, the magnetic tape or permanent file will have to be requested before execution of RANLIB.

The following pages contain examples of data trace library creation, modification, and retrieval using program RANLIB. The SCOPE system day-file is included to show the control cards necessary for each job.

Example 1

This example creates a master library from *ADD instructions and data cards which reside on ALTINPUT (TAPE4). The trace definition instructions are read from INPUT. When the *READ, ALTINPUT. instruction is encountered, RANLIB begins reading *ADD instructions and data cards from ALTINPUT (logical file TAPE4). The *READ, INPUT. instruction causes RANLIB to continue reading from INPUT where it encounters and executes the instructions *ORDER and *LIST,,DETAIL. Since library MASTER is the only library that exists when the *ORDER instruction is executed, only a listing of the MASTER library is produced.

```
MASTER      LIBRARY IS EMPTY, CREATION ASSUMED.
ALTERNATE   LIBRARY IS EMPTY, CREATION ASSUMED.
RANOUT      LIBRARY IS EMPTY, CREATION ASSUMED.
*NAME,1-18.
*FIELD(1),26-28,JDAY.
*FIELD(2),34-43,TRTMEAN.
*FIELD(3),44-53,STDEV.
*FIELD(4),29-33,NUMOBS.
*COMMENT,57-80.
*READ,ALTINPUT.
  INPUT STREAM WILL BE READ FROM UNIT 4
*ADD.
  TRACE AGINVB COLE 1723 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB DIPT 1721 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB DIPT 1723 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB HEMI 1721 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB HEMI 1723 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB HOMO 1721 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB HOMO 1723 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB HYME 1721 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB HYME 1723 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB ORTH 1721 ADDED TO LIBRARY MASTER
*ADD.
  TRACE AGINVB ORTH 1723 ADDED TO LIBRARY MASTER
*READ,INPUT.
  INPUT STREAM WILL BE READ FROM UNIT 5
*ORDER.
*LIST,,DETAIL.
```

DETAILED LISTING OF DATA TRACES ON LIBRARY MASTER

TRACE NAME	NUM.	PTS.	VALS./PT.		VARIABLE	NAMES	COMMENTS	
AGINVB COLE 1723	8		4	JDAY	TRTMEAN	NUMOBS	1G/MSQ	AG39
AGINVB DIPT 1721	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB DIPT 1723	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB HEMI 1721	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB HEMI 1723	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB HOMO 1721	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB HOMO 1723	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB Hyme 1721	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB Hyme 1723	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB ORTH 1721	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
AGINVB ORTH 1723	8		4	JDAY	TRTMEAN	STDEV	NUMOBS	1G/MSQ
NUMBER OF TRACES ON LIBRARY	11							

03/10/76 CSU SCOPE 3.3.14 B C012 C013 C140 C141 03/02/76
 08.30.43.TA1793J FROM AB 12A
 08.30.43.TA179.CM60000.T20.AFYR****.JP/RANLIB.
 08.30.43.REQUEST,TAPE1,*PF.
 08.30.44.COPYCR,INPUT,TAPE4.
 08.30.45.REWIND,TAPE4.
 08.30.45.ATTACH,LGO,RANLIBLGOPRG,ID=JDP,MR=1.
 08.30.50.CYCLE **, RANLIBLGOPRG
 08.30.50.PFN FOUND IN SD 026
 08.30.51.CYCLE 01, RANLIBLGOPRG
 08.30.51.FILE HAS BEEN ATTACHED
 08.30.51.LGO.
 08.31.01. NON-FATAL LOADER ERRORS - SEE MAP
 08.31.01.FL= 042700 CP 00001.861SEC. IO 00002.140SEC.
 08.31.13.STOP
 08.31.14.CATALOG,TAPE1,AGINVBLIB,ID=JDP,RP=999.
 08.31.22.PFD 4/5 FULL
 08.31.23.FILE CATALOGUED AS
 08.31.23.CYCLE 01, AGINVBLIB
 08.31.23.IN SD 001
 08.31.25.CP 3.365 SEC.
 08.31.25.PP 23.204 SEC.
 08.31.25.IO 4.320 SEC.

Example 2

This example creates a MASTER and an ALTERNATE library from a permanent file that was output from another program. The permanent file AGHTRACE contains data cards that contain the data values used to assemble and add traces to the libraries. The data sets on AGHTRACE are separated by SCOPE end-of-file marks.

All instructions are read from INPUT. The *ADD instructions define ALTINPUT (TAPE4) as the source from which to read data cards. Since both libraries MASTER and ALTERNATE exist when the *LIST,,DETAIL instruction is executed, a listing of the traces in each library is produced.

Both libraries produced by this example are to be used in later examples. Therefore the logical files associated with MASTER (TAPE1) and ALTERNATE (TAPE2) are requested as permanent files and subsequently cataloged (University Computer Center 1975a) after program RANLIB execution. The permanent file names of MASTER and ALTERNATE are AGHLIB1 and AGHLIB2, respectively.

```

MASTER LIBRARY IS EMPTY, CREATION ASSUMED.
ALTERNATE LIBRARY IS EMPTY, CREATION ASSUMED.
RANOUT LIBRARY IS EMPTY, CREATION ASSUMED.
*NAME,1-10.
*FIELD(1),26-28,JDAY.
*FIELD(2),34-43,TRTMEAN.
*FIELD(3),44-53,STDEV.
*FIELD(4),29-33,NUMOBS.
*COMMENT,57-80.
*ADD,MASTER,ALTINPUT,19.
TRACE VAC AGSM SC 4721 ADDED TO LIBRARY MASTER
TRACE VAC AGSM OD 4721 ADDED TO LIBRARY MASTER
TRACE VAC AGSM PL 4721 ADDED TO LIBRARY MASTER
TRACE VAC BUDA SC 4721 ADDED TO LIBRARY MASTER
TRACE VAC BUDA OD 4721 ADDED TO LIBRARY MASTER
TRACE VAC BUDA PL 4721 ADDED TO LIBRARY MASTER
TRACE VAC BOGR SC 4721 ADDED TO LIBRARY MASTER
TRACE VAC BOGR OD 4721 ADDED TO LIBRARY MASTER
TRACE VAC BOGR PL 4721 ADDED TO LIBRARY MASTER
TRACE VAC CAEL SC 4721 ADDED TO LIBRARY MASTER
TRACE VAC CAEL OD 4721 ADDED TO LIBRARY MASTER
TRACE VAC CAEL PL 4721 ADDED TO LIBRARY MASTER
TRACE VAC BRJA SC 4721 ADDED TO LIBRARY MASTER
TRACE VAC BRJA OD 4721 ADDED TO LIBRARY MASTER
TRACE VAC BRJA PL 4721 ADDED TO LIBRARY MASTER
TRACE VAC CANN SC 4721 ADDED TO LIBRARY MASTER
TRACE VAC CANN OD 4721 ADDED TO LIBRARY MASTER
TRACE VAC CANN PL 4721 ADDED TO LIBRARY MASTER
TRACE VAC WANN SC 4721 ADDED TO LIBRARY MASTER
*ADD,ALTERNATE,ALTINPUT,16.
TRACE VAC WANN OD 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC WANN PL 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC CPER SC 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC CPER OD 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC CPER PL 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC WPER SC 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC WPER OD 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC WPER PL 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC SUCCU SC 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC SUCCU OD 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC SUCCU PL 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC BOBU SC 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC BOBU OD 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC OTHER SC 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC OTHER OD 4721 ADDED TO LIBRARY ALTERNATE
TRACE VAC OTHER PL 4721 ADDED TO LIBRARY ALTERNATE
*ORDER.
*LIST,,DETAIL.

```

DETAILED LISTING OF DATA TRACES ON LIBRARY MASTER

TRACE NAME	NUM. PTS.	VALS./PT.		VARIABLE NAMES	COMMENTS
VAC AGSM OD 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
VAC AGSM PL 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 4 SPSV
VAC AGSM SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 0 SPSV
VAC BOGR OD 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
VAC BOGR PL 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 4 SPSV
VAC BOGR SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 0 SPSV
VAC BRJA OD 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
VAC BRJA PL 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 4 SPSV
VAC BRJA SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 0 SPSV
VAC BUDA OD 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
VAC BUDA PL 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 4 SPSV
VAC BUDA SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 0 SPSV
VAC CAEL OD 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
VAC CAEL PL 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 4 SPSV
VAC CAEL SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 0 SPSV
VAC CANN OD 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
VAC CANN PL 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 4 SPSV
VAC CANN SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 0 SPSV
VAC WANN SC 4721	10	4	JDAY	TRTMEAN STDEV NUMOBS	1G/MSQ VA 2 SPSV
NUMBER OF TRACES ON LIBRARY			19		

DETAILED LISTING OF DATA TRACES ON LIBRARY ALTERNATE

TRACE NAME	NUM. PTS.	VALS./PT.		VARIABLE NAMES	COMMENTS
VAC BOBU OD 4721	10	4	JDAY	TRTMEAN	
VAC BOBU SC 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 2 SPSV
VAC CPER OD 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 0 SPSV
VAC CPER PL 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 2 SPSV
VAC CPER SC 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 4 SPSV
VAC OTHER OD 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 0 SPSV
VAC OTHER PL 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 2 SPSV
VAC OTHER SC 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 4 SPSV
VAC SUCCU OD 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 0 SPSV
VAC SUCCU PL 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 2 SPSV
VAC SUCCU SC 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 4 SPSV
VAC WANN OD 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 0 SPSV
VAC WANN PL 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 2 SPSV
VAC WPER OD 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 4 SPSV
VAC WPER PL 4721	10	4	JDAY	TRTMEAN	NUMOBS 1G/MSQ VA 2 SPSV
VAC WPER SC 4721	10	4	JDAY	STDEV	NUMOBS 1G/MSQ VA 4 SPSV
NUMBER OF TRACES ON LIBRARY			16	TRTMEAN	NUMOBS 1G/MSQ VA 0 SPSV

03/10/76 CSU SCOPE 3.3.14 B C012 C013 C140 C141 03/02/76
 08.30.35.TA1793I FROM AB 11A
 08.30.35.TA179,CM60000,T20,AFYR****.JP/RANLIB.
 08.30.36.ATTACH,TAPE4,AGHTRACE,ID=JDP.
 08.30.38.CYCLE **, AGHTRACE
 08.30.38.PFN FOUND IN SD 012
 08.30.38.CYCLE 01, AGHTRACE
 08.30.38.FILE HAS BEEN ATTACHED
 08.30.38.REQUEST,TAPE1,*PF.
 08.30.39.REQUEST,TAPE2,*PF.
 08.30.39.ATTACH,LGO,RANLIBLGOPRG,ID=JDP,MR=1.
 08.30.44.CYCLE **, RANLIBLGOPRG
 08.30.44.PFN FOUND IN SD 026
 08.30.44.CYCLE 01, RANLIBLGOPRG
 08.30.44.FILE HAS BEEN ATTACHED
 08.30.44.LGO.
 08.30.52. NON-FATAL LOADER ERRORS - SEE MAP
 08.30.53.FL= 042700 CP 00001.829SEC. IO 00002.010SEC.
 08.31.34.STOP
 08.31.34.CATALOG,TAPE1,AGHLIB1,ID=JDP,RP=14.
 08.31.39.PFD 4/5 FULL
 08.31.40.FILE CATALOGUED AS
 08.31.40.CYCLE 01, AGHLIB1
 08.31.40.IN SD 002
 08.31.40.CATALOG,TAPE2,AGHLIB2,ID=JDP,RP=14.
 08.31.48.PFD 4/5 FULL
 08.31.49.FILE CATALOGUED AS
 08.31.49.CYCLE 01, AGHLIB2
 08.31.49.IN SD 003
 08.31.51.CP 6.095 SEC.
 08.31.51.PP 32.591 SEC.
 08.31.51.IO 6.375 SEC.

Example 3

In this example the data traces in the libraries created by example 2 are merged and entered in library RANOUT. The permanent files AGHLIB1 and AGHLIB2 are attached to logical files TAPE1 (MASTER) and TAPE2 (ALTERNATE) prior to execution of program RANLIB.

RANLIB encounters the *MERGE, MASTER, ALTERNATE. instruction and copies each library to RANOUT (the default library). The *ORDER instruction informs RANLIB to sort the trace names by columns 1-18 before any output instructions are executed. Since a detailed listing of the libraries MASTER and ALTERNATE was produced in example 2, the *LIST, RANOUT, DETAIL. instruction specifically requests that only library RANOUT be listed.

The merged data trace library created in this example will be used in example 4. Therefore logical file TAPE3 (RANOUT) was requested as a SCOPE (University Computer Center 1975a) system permanent file prior to execution of RANLIB and cataloged with the permanent file name AGHLIB12 after execution of program RANLIB.

MASTER LIBRARY HAS BEEN ATTACHED.
 ALTERNATE LIBRARY HAS BEEN ATTACHED.
 RANOUT LIBRARY IS EMPTY, CREATION ASSUMED.
 *MERGE,MASTER,ALTERNATE.
 LIBRARIES MASTER AND ALTERNATE MERGED AND WRITTEN TO RANOUT
 *ORDER,1-18.
 *LIST,RANOUT,DETAIL.

DETAILED LISTING OF DATA TRACES ON LIBRARY RANOUT

TRACE NAME	NUM. PTS.	VALS./PT.		VARIABLE	NAMES	COMMENTS
VAC AGSM OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC AGSM PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC AGSM SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC BOBU OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC BOBU SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC BOGR OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC BOGR PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC BOGR SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC BRJA OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC BRJA PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC BRJA SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC BUDA OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC BUDA PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC BUDA SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC CAEL OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC CAEL PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC CAEL SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC CANN OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC CANN PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC CANN SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC CPER OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC CPER PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC CPER SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC OTHER OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC OTHER PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC OTHER SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC SUCCU OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC SUCCU PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC SUCCU SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC WANN OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC WANN PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC WANN SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
VAC WPER OD 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 2 SPSV
VAC WPER PL 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 4 SPSV
VAC WPER SC 4721	10	4	JDAY	TRTMEAN	STDEV NUMOBS 1G/MSQ	VA 0 SPSV
NUMBER OF TRACES ON LIBRARY			35			

03/10/76 CSU SCOPE 3.3.14 B C012 C013 C140 C141 03/02/76
 08.35.44.TA17932 FROM AB 7A
 08.35.45.TA179.CM60000.T20.AFYR****.JP/RANLIB.
 08.35.45.REQUEST.TAPE3.*PF.
 08.35.51.ATTACH,TAPE1,AGHLIB1,ID=JDP.
 08.35.51.CYCLE **, AGHLIB1
 08.35.51.PFN FOUND IN SD 002
 08.35.52.CYCLE 01, AGHLIB1
 08.35.52.FILE HAS BEEN ATTACHED
 08.35.52.ATTACH,TAPE2,AGHLIB2,ID=JDP.
 08.35.52.CYCLE **, AGHLIB2
 08.35.52.PFN FOUND IN SD 003
 08.35.52.CYCLE 01, AGHLIB2
 08.35.52.FILE HAS BEEN ATTACHED
 08.35.53.ATTACH,LGO,RANLIBLGOPRG,ID=JDP,MR=1.
 08.35.54.CYCLE **, RANLIBLGOPRG
 08.35.54.PFN FOUND IN SD 026
 08.35.55.CYCLE 01, RANLIBLGOPRG
 08.35.55.FILE HAS BEEN ATTACHED
 08.35.55.LGO.
 08.36.05. NON-FATAL LOADER ERRORS - SEE MAP
 08.36.05.FL= 042700 CP 00001.738SEC. IO 00002.054SEC.
 08.37.25.STOP
 08.37.25.CATALOG,TAPE3,AGHLIB12,ID=JDP,RP=999.
 08.37.27.PFD 4/5 FULL
 08.37.28.FILE CATALOGUED AS
 08.37.28.CYCLE 01, AGHLIB12
 08.37.28.IN SD 005
 08.37.28.CP 2.403 SEC.
 08.37.28.PP 105.328 SEC.
 08.37.28.IO 88.870 SEC.

Example 4

In this example, the data trace "VAC BUDA PL 4721" which resides on the library created by example 3 (AGHLIB12) is replaced by data obtained from the summarized data file (AGHTRACE) used to create libraries AGHLIB1 and AGHLIB2 in example 2. The trace library is attached to TAPE1 (MASTER). The summarized data is attached to TAPE4 (ALTINPUT) prior to execution of RANLIB.

When a data trace is replaced, the structure of the trace must be redefined. This is accomplished by the instructions *NAME, *FIELD, and *COMMENT. The data used to assemble the replacement trace is the 6th data set on the input file ALTINPUT (AGHTRACE, which was attached to TAPE4). The instruction *SKIPF, ALTINPUT, 5. positions ALTINPUT to the desired data set and *REPLACE, MASTER, ALTINPUT. accomplishes the task of replacing the data trace. Since there is no change in the contents of the library, the *ORDER and *LIST instructions are unnecessary. However, to insure that the library modification is permanent, the SCOPE control card EXTEND,TAPE1 (University Computer Center 1975a) follows the program execution card LGO.

MASTER LIBRARY HAS BEEN ATTACHED.
ALTERNATE LIBRARY IS EMPTY, CREATION ASSUMED.
RANOUT LIBRARY IS EMPTY, CREATION ASSUMED.

*NAME,1-18.
*FIELD(1),26-28,JDAY.
*FIELD(2),34-43,TRTMEAN.
*FIELD(3),44-53,STDEV.
*FIELD(4),29-33,NUMOBS.
*COMMENT,57-80.
*SKIPF,ALTINPUT,5.
*REPLACE,MASTER,ALTINPUT.
TRACE VAC BUDA PL 4721 REPLACED ON LIBRARY MASTER

03/10/76 CSU SCOPE 3.3.14 B: C012 C013 C140 C141 03/02/76
13.17.09.TA179UZ FROM AB 13A
13.17.09.TA179,CM60000,T20,AFYR****.JP/RANLIB.
13.17.10.ATTACH,LGO,RANLIBLGOPRG,ID=JDP,MR=1.
13.17.10.CYCLE **, RANLIBLGOPRG
13.17.10.PFN FOUND IN SD 019
13.17.10.CYCLE 01, RANLIBLGOPRG
13.17.10.FILE HAS BEEN ATTACHED
13.17.10.ATTACH,TAPE1,AGHLIB12,ID=JDP.
13.17.11.CYCLE **, AGHLIB12
13.17.11.PFN FOUND IN SD 005
13.17.11.CYCLE 01, AGHLIB12
13.17.11.FILE HAS BEEN ATTACHED
13.17.11.ATTACH,TAPE4,AGHTRACE,ID=JDP.
13.17.11.CYCLE **, AGHTRACE
13.17.11.PFN FOUND IN SD 012
13.17.11.CYCLE 01, AGHTRACE
13.17.11.FILE HAS BEEN ATTACHED
13.17.11.LGO.
13.17.14. NON-FATAL LOADER ERRORS - SEE MAP
13.17.14.FL= 042700 CP:00001.712SEC. IO 00002.045SEC.
13.17.15.STOP
13.17.15.EXTEND,TAPE1.
13.17.15.CYCLE 01, AGHLIB12
13.17.15.FILE EXTENDED
13.17.16.CP 1.970 SEC.
13.17.16.PP 5.636 SEC.
13.17.16.IO 2.530 SEC.

Example 5

In this example, two libraries are created by copying a data trace to each of them from the library that was created by example 1. In addition, one data trace is deleted from the original library, reducing the number of traces in that library from 11 to 10. The original library is attached to TAPE2 (ALTERNATE). prior to execution of RANLIB. The *LIST instruction lists all three libraries since they all exist as a result of the instructions executed by RANLIB. Compare the library listings with that in example 1 to see that one trace has actually been eliminated from the original library.

LOAD MAP - RANLIB.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCSSR	VER	LEVEL	HARDWARE	COMMENTS
INITMSS	36715	121	SL-SYSMISC	12/05/7					
INPUTCS	37036	134	SL-SYSMISC	12/05/7					
INPUTSS	37172	55	SL-SYSMISC	04/13/7					
MODERS	37247	1424	SL-SYSMISC	12/09/7					
REWINDS	40673	52	SL-SYSMISC	12/05/7					
WRITMSS	40745	103	SL-SYSMISC	04/13/7					
REMARKS	41050	30	SL-SYSMISC	12/05/7					
GETRA	41100	17	SL-SYSMISC	03/19/7					
SIO\$	41117	1463	SL-SYSMISC	03/19/7					
//	42602	62							

LOADER 1.0

03/12/76 14.45.01.

1.701 CP SECONDS

55500B CM STORAGE USED

9 TABLE MOVES

MASTER LIBRARY IS EMPTY, CREATION ASSUMED.
 ALTERNATE LIBRARY HAS BEEN ATTACHED.
 RANOUT LIBRARY IS EMPTY, CREATION ASSUMED.
 *COPY,ALTERNATE,RANOUT,AGINVB HEMI 1721.
 COPY FROM ALTERNATE TO RANOUT COMPLETE.
 *COPY,ALTERNATE,MASTER,AGINVB HEMI 1723.
 COPY FROM ALTERNATE TO MASTER COMPLETE.
 *DELETE,ALTERNATE,AGINVB ORTH 1723.
 DATA TRACE AGINVB ORTH 1723 ELIMINATED FROM LIBRARY ALTERNATE
 *ORDER.
 *LIST,,DETAIL.

DETAILED LISTING OF DATA TRACES ON LIBRARY MASTER

TRACE NAME	NUM. PTS.	VALS./PT.	JDAY	TRTMEAN	STDEV	VARIABLE NAMES	NUMOBS	1G/MSQ	COMMENTS	AGIN
AGINVB HEMI 1723	8	4	JDAY						AG41	
NUMBER OF TRACES ON LIBRARY		1								

DETAILED LISTING OF DATA TRACES ON LIBRARY ALTERNATE

TRACE NAME	NUM. PTS.	VALS./PT.	JDAY	TRTMEAN	STDEV	VARIABLE NAMES	NUMOBS	1G/MSQ	COMMENTS	AGIN
AGINVB COLE 1723	8	4	JDAY						AG39	AGIN
AGINVB DIPT 1721	8	4	JDAY						AG40	AGIN
AGINVB DIPT 1723	8	4	JDAY						AG40	AGIN
AGINVB HEMI 1721	8	4	JDAY						AG41	AGIN
AGINVB HEMI 1723	8	4	JDAY						AG41	AGIN
AGINVB HOMO 1721	8	4	JDAY						AG42	AGIN
AGINVB HOMO 1723	8	4	JDAY						AG42	AGIN
AGINVB HYME 1721	8	4	JDAY						AG43	AGIN
AGINVB HYME 1723	8	4	JDAY						AG43	AGIN
AGINVB ORTH 1721	8	4	JDAY						AG43	AGIN
NUMBER OF TRACES ON LIBRARY		10							AG44	AGIN

DETAILED LISTING OF DATA TRACES ON LIBRARY RANOUT

TRACE NAME	NUM. PTS.	VALS./PT.	JDAY	TRTMEAN	VARIABLE NAMES	NUMOBS	IG/MSQ	COMMENTS	AG+1	AGIN
------------	-----------	-----------	------	---------	----------------	--------	--------	----------	------	------

03/12/76 CSU SCOPE 3.3.14 0 C012 C013 C140 C141 03/02/76
14.40.35.TA1795Z FROM AB 10A
14.40.35.TA179,CM60000,T20,AFYR****.JP/RANLIB.
14.40.35.ATTACH.LGO,RANLIBLGOPRG,ID=JDP,MR=1.
14.40.39.CYCLE **, RANLIBLGOPRG
14.40.39.PFN FOUND IN SD 019
14.40.39.CYCLE 01, RANLIBLGOPRG
14.40.39.FILE HAS BEEN ATTACHED
14.40.40.ATTACH.TAPE2,AGINVBLIB,ID=JDP.
14.40.41.CYCLE **, AGINVBLIB
14.40.41.PFN FOUND IN SD 001
14.40.42.CYCLE 01, AGINVBLIB
14.40.42.FILE HAS BEEN ATTACHED
14.40.47.LGO.
14.42.56.POLLOUT COMPLETED. (FL 60000)
14.44.54.ROLLIN COMPLETED.
14.45.01. NON-FATAL LOADER ERRORS - SEE MAP
14.45.01.FL= 042700 CP 00001.781SEC. IO 00002.005SEC.
14.45.06.STOP
14.45.07.CP 2.063 SEC.
14.45.07.PP 14.441 SEC.
14.45.07.IO 3.005 SEC.

Example 6

In this example, a data trace is selected for output from the library created by example 3. The output format is defaulted since no *FMTOU instruction is included. The library "AGHLIB12" is attached to TAPE1 (MASTER) prior to execution of RANLIB. The selected trace is written to TAPE7 by RANLIB. TAPE7 is rewound and copied to OUTPUT by SCOPE system control cards after the program execution card LGO. to obtain a printed copy of the selected output.

MASTER LIBRARY HAS BEEN ATTACHED.
ALTERNATE LIBRARY IS EMPTY, CREATION ASSUMED.
RANOUT LIBRARY IS EMPTY, CREATION ASSUMED.
*SELECT,MASTER,VAC AGSM OD 4721.
TRACE NAME VAC AGSM OD 4721 SELECTED AND WRITTEN TO AUXILLARY OUT

VAC AGSM OD 4721	77.0000	76.3000	8.9000	18.9000
VAC AGSM OD 4721	90.0000	124.4000	21.0000	13.8000
VAC AGSM OD 4721	112.0000	143.9000	11.1000	16.2000
VAC AGSM OD 4721	139.0000	80.1000	9.3000	18.9000
VAC AGSM OD 4721	153.0000	32.1000	4.9000	11.7000
VAC AGSM OD 4721	182.0000	36.1000	1.8000	18.6000
VAC AGSM OD 4721	202.0000	99.0000	6.6000	18.5000
VAC AGSM OD 4721	237.0000	102.3000	6.9000	18.1000
VAC AGSM OD 4721	266.0000	104.9000	6.0000	15.9000
VAC AGSM OD 4721	301.0000	145.4000	7.5000	14.4000

03/10/76 CSU SCOPE 3.3.14 B C012 C013 C140 C141 03/02/76
08.53.34.TA17942 FROM AB 3A
08.53.34.TA179,CM60000,T20,AFYR***.JP/RANLIB.
08.53.36.ATTACH,TAPE1,AGHLIB12,ID=JDP.
08.53.38.CYCLE **, AGHLIB12
08.53.38.PFN FOUND IN SD 005
08.53.39.CYCLE 01, AGHLIB12
08.53.39.FILE HAS BEEN ATTACHED
08.53.39.ATTACH,LGO,RANLIBLGOPRG,ID=JDP,MR=1.
08.53.44.CYCLE **, RANLIBLGOPRG
08.53.44.PFN FOUND IN SD 026
08.53.44.CYCLE 01, RANLIBLGOPRG
08.53.44.FILE HAS BEEN ATTACHED
08.53.44.LGO.
08.54.27. NON-FATAL LOADER ERRORS - SEE MAP
08.54.27.FL= 042700 CP 00001.737SEC. IO 00002.005SEC.
08.54.29.STOP
08.54.29.REWIND,TAPE7.
08.54.29.COPYSBF,TAPE7,OUTPUT.
08.54.31.CP 1.978 SEC.
08.54.31.PP 13.929 SEC.
08.54.31.IO 2.230 SEC.

Example 7

In this example, two sets of data traces are selected for sequential output from the library created by example 3. The *FMTOUT instruction formats the output into the following organization:

Columns 1-18	Trace name
Columns 19-21	"JDT"
Columns 22-24	Julian day
Columns 26-32	Treatment mean
Columns 34-40	Standard deviation
Columns 41-44	"NOBS"
Columns 45-48	Number of observations
Columns 49-78	Comment

The data sets selected for output are separated by a SCOPE end-of-file mark which is written as a result of the *EOR instruction.

The library AGHLIB12 is attached to TAPE1 (MASTER) prior to execution of RANLIB. The SCOPE control card LGO. causes execution of the program. Since a printed output of the results of the example are needed, the sequential output file TAPE7 is rewound and copied to output. There are two COPYSBF,TAPE7,OUTPUT. cards since the two data sets created by the *SELECT instructions are separated by an end-of-file mark.

```
MASTER      LIBRARY HAS BEEN ATTACHED.
ALTERNATE   LIBRARY IS EMPTY, CREATION ASSUMED.
RANOUT      LIBRARY IS EMPTY, CREATION ASSUMED.
*FMTOUT,NAME/1-18/,JDAY/*JDT *,I3/,TRTMEAN/1X,F7.2/,STDEV/1X,F7.4/,NUMOBS/*NOBS*
*/,I4/,COMMENT/1-30/.
*SELECT,*,VAC AGSM  OD 4721-VAC AGSM  SC 4721.
TRACE NAME VAC AGSM  OD 4721   SELECTED AND WRITTEN TO AUXILLARY OUT
TRACE NAME VAC AGSM  PL 4721   SELECTED AND WRITTEN TO AUXILLARY OUT
TRACE NAME VAC AGSM  SC 4721   SELECTED AND WRITTEN TO AUXILLARY OUT
*EOR
*SELECT,MASTER,VAC BOGR  OD 4721-VAC BOGR  SC 4721.
TRACE NAME VAC BOGR  OD 4721   SELECTED AND WRITTEN TO AUXILLARY OUT
TRACE NAME VAC BOGR  PL 4721   SELECTED AND WRITTEN TO AUXILLARY OUT
TRACE NAME VAC BOGR  SC 4721   SELECTED AND WRITTEN TO AUXILLARY OUT
```

VAC AGSM	OD	4721	JDT	77	76.30	8.9000NOBS	181G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	90	124.40	21.0000NOBS	131G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	112	143.90	11.1000NOBS	161G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	139	80.10	9.3000NOBS	181G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	153	32.10	4.9000NOBS	111G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	182	36.10	1.8000NOBS	181G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	202	99.00	6.6000NOBS	181G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	237	102.30	6.9000NOBS	181G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	266	104.90	6.0000NOBS	151G/MSQ	VA	2	SPSV
VAC AGSM	OD	4721	JDT	301	145.40	7.5000NOBS	141G/MSQ	VA	2	SPSV
VAC AGSM	PL	4721	JDT	77	.60	.2000NOBS	111G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	90	3.10	.8000NOBS	121G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	112	11.80	.9000NOBS	111G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	139	30.60	3.5000NOBS	181G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	153	66.90	4.9000NOBS	171G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	182	104.30	7.5000NOBS	141G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	202	153.10	6.3000NOBS	111G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	237	88.80	9.2000NOBS	141G/MSQ	VA	4	SPSV
VAC AGSM	PL	4721	JDT	266	83.50	4.1000NOBS	171G/MSQ	VA	4	SPSV
VAC AGSM	SC	4721	JDT	77	8.30	1.6000NOBS	121G/MSQ	VA	4	SPSV
VAC AGSM	SC	4721	JDT	90	76.90	8.9000NOBS	181G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	112	127.50	21.5000NOBS	131G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	139	155.70	11.6000NOBS	161G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	153	110.70	11.7000NOBS	181G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	182	99.00	6.3000NOBS	191G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	202	140.40	9.0000NOBS	141G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	237	252.10	11.0000NOBS	121G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	266	191.00	15.0000NOBS	141G/MSQ	VA	0	SPSV
VAC AGSM	SC	4721	JDT	301	188.40	9.4000NOBS	141G/MSQ	VA	0	SPSV
VAC BOGR	OD	4721	JDT	77	153.70	7.7000NOBS	151G/MSQ	VA	0	SPSV
VAC BOGR	OD	4721	JDT	90	0.00	0.0000NOBS	181G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	112	0.00	0.0000NOBS	131G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	139	0.00	0.0000NOBS	161G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	153	0.00	0.0000NOBS	181G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	182	0.00	0.0000NOBS	111G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	202	.50	0.0000NOBS	181G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	237	2.50	.2000NOBS	181G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	266	4.60	.8000NOBS	181G/MSQ	VA	2	SPSV
VAC BOGR	OD	4721	JDT	301	1.60	.2000NOBS	151G/MSQ	VA	2	SPSV
VAC BOGR	PL	4721	JDT	77	3.90	.7000NOBS	141G/MSQ	VA	2	SPSV
VAC BOGR	PL	4721	JDT	90	0.00	0.0000NOBS	111G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	112	0.00	0.0000NOBS	121G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	139	.40	.1000NOBS	111G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	153	0.00	0.0000NOBS	181G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	182	14.00	1.5000NOBS	171G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	202	15.30	1.2000NOBS	141G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	237	7.70	.6000NOBS	111G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	266	8.10	1.5000NOBS	141G/MSQ	VA	4	SPSV
VAC BOGR	PL	4721	JDT	301	2.30	.4000NOBS	171G/MSQ	VA	4	SPSV
VAC BOGR	SC	4721	JDT	77	2.00	.5000NOBS	121G/MSQ	VA	4	SPSV
VAC BOGR	SC	4721	JDT	90	0.00	0.0000NOBS	181G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	112	0.00	0.0000NOBS	131G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	139	.40	.1000NOBS	161G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	153	0.00	0.0000NOBS	181G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	182	14.00	1.5000NOBS	191G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	202	15.80	1.2000NOBS	141G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	237	10.20	.7000NOBS	121G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	266	12.80	1.9000NOBS	141G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	301	3.90	.6000NOBS	141G/MSQ	VA	0	SPSV
VAC BOGR	SC	4721	JDT	301	5.90	1.1000NOBS	151G/MSQ	VA	0	SPSV

03/10/76 CSU SCOPE 3.3.14 B C012 C013 C140 C141 03/02/76
08.58.37.TA17949 FROM AB 13A
08.58.37.TA179,CM60000,T20,AFYR***.JP/RANLIB.
08.58.41.ATTACH.TAPE1,AGHLIB12,ID=JDP.
08.58.42.CYCLE **, AGHLIB12
08.58.43.PFN FOUND IN SD 005
08.58.42.CYCLE 01, AGHLIB12
08.58.42.FILE HAS BEEN ATTACHED
08.58.42.ATTACH.LGO,RANLIBLGOPRG,ID=JDP,MR=1.
08.58.47.CYCLE **, RANLIBLGOPRG
08.58.47.PFN FOUND IN SD 026
08.58.48.CYCLE 01, RANLIBLGOPRG
08.58.48.FILE HAS BEEN ATTACHED
08.58.48.LGO.
08.59.05. NCN-FATAL LOADER ERRORS - SEE MAP
08.59.06.FL= 042700 CP 00001.743SEC. IO 00002.010SEC.
08.59.09.STOP
08.59.09.REWIND.TAPE7.
08.59.09.COPYSBF,TAPE7,OUTPUT.
08.59.09.COPYSBF,TAPE7,OUTPUT.
08.59.10.CP 2.656 SEC.
08.59.10.PP 12.081 SEC.
08.59.10.IO 2.545 SEC.

ERROR MESSAGES

When program RANLIB interprets an instruction, it performs extensive error checking to insure that the instruction was properly constructed. Errors detected concerning the physical form of the instruction cause an error diagnostic to be written immediately following the instruction on the printed output. Logical errors that are detected while the program is performing the requested action will cause an error diagnostic to be written on the printed output at the point the error is detected. In either case, the program will continue scanning instructions for detectable errors and will then abort and print an error message in the system day file. Any SCOPE system control cards following the program execution card will not be processed.

This section contains error messages written by RANLIB for each instruction. They are grouped by the instruction for which they may be written. Most are self-explanatory.

*ADD

LIBRARY FILE OR SOURCE FILE NAME LENGTH EXCEEDED.

A trace library name or source file name may be 1-10 characters in length.

ILLEGAL CHARACTER <char> COLUMN <col>, LIBRARY NAME OR SOURCE NAME IS ALPHANUMERIC (LETTERS/NUMBERS).

<char> is the illegal character.

THE TRACE TO BE ADDED TO <libname> FROM <source> HAS NO NAME.

The user failed to define the trace name prior to the *ADD instruction.

ILLEGAL FIELD DEFINED GREATER THAN THE NUMBER OF FIELDS.

Coordinate fields must be consecutively defined; 4 fields must be 1, 2, 3, 4,; not 1, 2, 3, 5.

<libname> IS NOT A DEFINED LIBRARY NAME.

A library name in the instruction was not MASTER, ALTERNATE, or RANOUT.

DUPLICATE TRACE IDENTIFIER <tracename> ON LIBRARY FILE <libname>.
A trace with the same name as the one being added already exists on the library.

<libname> IS FULL.

An attempt to exceed the capacity of the library has been detected.

<source> SOURCE FILE UNDEFINED.

Source file name is not INPUT or ALTINPUT.

ILLEGAL CHARACTER IN NAME FIELD OF DATA RECORD.

The implicitly defined trace name on the data record contains an illegal character.

INPUT BUFFER LENGTH EXCEEDED.

The number of data points in a trace exceeded the input buffer capacity. RANLIB will allow 366 points with 6 coordinate fields defined.

NO DATA ON INPUT SOURCE FILE.

An end of field was encountered before any data was read.

*COMMENT

COMMENT FIELD IS TOO LONG, IT WILL BE TRUNCATED TO 80 CHARACTERS.

Explicitly defined comment contained more than 80 characters.

COMMENT INSTRUCTION COLUMN DEFINITION CONTAINS NONNUMERIC CHARACTERS.

COLUMN NUMBERS ON COMMENT INSTRUCTION OUT OF ORDER.

*COPY

LIBRARY NAME HAS TOO MANY CHARACTERS.

Library name may contain 1-10 characters.

ILLEGAL CHARACTER <char> COLUMN <col>.

<char> is the illegal character.

END OF RECORD REACHED BEFORE INSTRUCTION TERMINATION, COPY CANNOT CONTINUE TO ANOTHER RECORD.

User forgot to terminate the instruction with a period.

<libname> IS NOT A DATA LIBRARY.

Library name is not MASTER, ALTERNATE, or RANOUT.

COPY CANNOT OCCUR FROM <libname> TO <libname>.

<libname> HAS NOT BEEN CREATED YET.

User may have attached a library to the wrong logical file.

TRACE NAME <tracename> DOES NOT RESIDE ON <libname>.
DUPLICATE TRACE NAME <tracename> ON LIBRARY <libname>.

*DELETE

LIBRARY NAME HAS TOO MANY CHARACTERS.
TRACE NAME HAS TOO MANY CHARACTERS.
<libname> IS NOT A DEFINED LIBRARY NAME.
 <libname> is not MASTER, ALTERNATE, or RANOUT.
<libname> IS NOT A DATA LIBRARY.
<tracename> NAMED TRACE DOES NOT RESIDE ON <libname> LIBRARY.
 User may have attached library to the wrong logical file.
<char> IS ILLEGAL CHARACTER IN COLUMN <col>.
A TRACE NAME WAS NOT DEFINED ON THE DELETE INSTRUCTION.

*FIELD

NAME INSTRUCTION MUST PRECEED FIELD DEFINITIONS.
ILLEGAL FIELD DEFINITION, FIELD NUMBER MUST BE IN RANGE 1-6.
FIELD NAME HAS MORE THAN 10 CHARACTERS.
ILLEGAL CHARACTER <char> COLUMN <col>.

*FMTOUT

FIELD NAME GREATER THAN 10 CHARACTERS OR FORMAT SPECIFICATION GREATER
 THAN 9 CHARACTERS.
 The coordinate field name may be 1-10 characters in length.
 Format specification enclosed by / / cannot exceed 9 characters.
THE NUMBER OF FIELDS SPECIFIED FOR OUTPUT CANNOT EXCEED 8.
FIELD POSITION NUMBER MUST BE IN RANGE 1-6.
NAME OR COMMENT SPECIFICATION CANNOT HAVE MORE THAN 5 SUBFIELDS.
 Name or comment extraction list cannot extract more than 5 subfields.
 A skip specification is counted as an extraction.

SUBFIELD COLUMNS FOR NAME ARE LIMITED TO 1-20.

SUBFIELD COLUMNS FOR COMMENT ARE LIMITED TO 1-30.

COLUMN NUMBERS IN NAME/COMMENT SUBFIELD ARE OUT OF ORDER.

TOTAL NUMBER OF COLUMNS EXTRACTED FROM NAME FIELD CANNOT EXCEED 20.
The sum of columns extracted and any skip specifications may not exceed 20.

TOTAL NUMBER OF COLUMNS EXTRACTED FROM COMMENT FIELD CANNOT EXCEED 30.
The sum of columns extracted and any skip specifications may not exceed 30.

ILLEGAL CHARACTER <char> COLUMN <col>.

FMTOUT EXPECTED A CONTINUATION CARD.
RANLIB reached the end of the instruction without encountering a period.

*LIST

LIBRARY OR KEYWORD HAS TOO MANY CHARACTERS.
The user misspelled a library name or the keyword DETAIL.

<libname> IS NOT A DATA LIBRARY AND CANNOT BE LISTED.

ILLEGAL CHARACTER <char> COLUMN <col> IN LIST INSTRUCTION.

*MERGE

TOO MANY LIBRARIES NAMED OR IMPLIED ON MERGE INSTRUCTION.
More library names were defined or defaulted in the instruction than the three required.

<libname> IS NOT A DATA LIBRARY.

OUTPUT LIBRARY IS FULL, MERGE TERMINATED.
The libraries being merged contain more data traces than the output library may contain.

DUPLICATE TRACE NAME ON OUTPUT LIBRARY.
A data trace with the same name as a trace on a source library is already in the output library.

ILLEGAL CHARACTER <char> COLUMN <col>.

*NAME

TRACE NAME EXCEEDS 20 CHARACTERS.

Trace names may be 1-20 characters in length.

ILLEGAL CHARACTER <char> COLUMN <col>.

INSTRUCTION TERMINATED BEFORE TRACE NAME BUILT.

User may have mispunched a period for a comma.

COLUMN DEFINITION CONTAINS NONNUMERIC CHARACTERS.

Error in implicit name definition.

COLUMN DELIMITERS INDICATE TRACE NAME EXCEEDS 20 CHARACTERS.

Implicit column definition defines a trace name to be read which is greater than 20 characters in length.

COLUMN NUMBERS FOR NAME FIELD ARE OUT OF ORDER.

*ORDER

KEY FIELDS FOR TRACE ORDERING ARE RESTRICTED TO COLUMNS 1-20.

TOTAL NUMBER OF COLUMNS WHICH CAN BE ORDERED CANNOT EXCEED 20.

The sum of columns defined by key fields must be less than or equal to 20.

ILLEGAL CHARACTER <char> COLUMN <col>.

*READ

FILE NAME HAS MORE THAN 10 CHARACTERS.

<filename> IS NOT A LEGAL INPUT FILE.

Input file names are INPUT and ALTINPUT.

CHARACTER <char> COLUMN <col> IS ILLEGAL.

*REPLACE

TRACE <tracename> IS NOT ON LIBRARY <libname> AND CANNOT BE REPLACED.

*REPLACE is a variation of *ADD; therefore, error messages for *ADD may also be written.

*SELECT

LIBRARY NAME HAS MORE THAN 10 CHARACTERS OR TRACE NAME HAS MORE THAN 20 CHARACTERS.

<libname> IS NOT A DATA LIBRARY.

<libname> DATA LIBRARY IS EMPTY.

User may have attached library to wrong logical file.

<tracename> IS NOT ON LIBRARY <libname>.

TRACE <tracename> DOES NOT HAVE A FIELD NAMED <fieldname>.

Data trace does not contain a coordinate field named in the *FMTOUT instruction.

SELECT EXPECTED A CONTINUATION TO ANOTHER CARD.

User may have left an ending period off of SELECT card.

CHARACTER <char> COLUMN <col> IS ILLEGAL.

*SKIPF

FILE NAME ON SKIPF INSTRUCTION IS MORE THAN 10 CHARACTERS.

<filename> IS NOT A LEGAL INPUT FILE NAME.

ILLEGAL CHARACTER <char>.

ACKNOWLEDGMENTS

My thanks to Jon D. Gustafson and David M. Swift who were responsible for the inception of the system; Van Baker who, along with Jon Gustafson, was instrumental in helping define the RANLIB instruction set; and Kim Stevens who helped to coordinate the Data Trace System with the

SIMCOMP plotting package.

- Control Data Corporation. 1973. FORTRAN extended reference manual, 6000 version 3.0. Control Data Corporation, Sunnyvale, Calif. ca. 210 p.
- Gustafson, J. D., and G. S. Innis. 1973. SIMCOMP version 3.0 user's manual. US/IBP Grassland Biome Tech. Rep. No. 218, Colorado State Univ., Fort Collins. 149 p.
- Stevens, K., in prep. RALPLOT user's guide. Tech. Rep., Colorado State Univ., Fort Collins.
- University Computer Center. 1975^a. User's manual, Section III.2 and III.3. University Computer Center, Colorado State Univ., Fort Collins. p. 3-4 to 3-28.
- University Computer Center. 1975^b. User's manual, Index. University Computer Center, Colorado State Univ., Fort Collins. p. Index-1 to Index-9.

Addendum to Technical Report No. 299
Program RANLIB: A data storage and
management system user's manual

Jerry D. Peltz

Natural Resource Ecology Laboratory
Colorado State University
Fort Collins, Colorado 80523

GRASSLAND BIOME
U.S. International Biological Program

November 1977

UPDATE TO THE RANLIB INSTRUCTION SET

After using program RANLIB to create and maintain large time-trace data libraries, it became necessary to modify the *COPY instruction and to add *LOAD and *UNLOAD instructions.

The modification to the *COPY instruction incorporates adding a feature where a specific range of traces may be copied from one library to another. The general format of the instruction follows.

*COPY, <sourcelibrary>, <targetlibrary>, <traceparameters>.

where <traceparameters> is defined by the following:

```
<traceparameters>:=<tracename>|  
<tracename>,<traceparameters>|  
<tracename>-<traceparameters>
```

For example, the instruction

*COPY, MASTER, RANOUT, PAWBGHD174-PAWBGHG174. will cause traces "PAWBGHD174" *through* "PAWBGHG174" on the library associated with MASTER to be copied to the library RANOUT.

*COPY, TRACEAAA-TRACEZZZ, TRACE01, TRACE05. will cause traces "TRACE01 and TRACE05" to be copied to RANOUT from MASTER in addition to traces TRACEAAA through TRACEZZZ.

A selected group of traces may be copied from one library to another by using the *ORDER instruction in conjunction with the *COPY instruction. Suppose the MASTER library contains the following traces in sequence:

```
TRACEAAA  
TRACEBBB  
TRACECCC  
TRACE01  
TRACE15  
TRACEZZZ  
TRACEDDD
```

Traces TRACEAAA through TRACEDDD excluding TRACE01 and TRACE15 could be copied to another library with the instructions:

*ORDER.

*COPY,MASTER,ALTERNATE,TRACEAAA-TRACEDDD.

The *ORDER instruction should be used whenever a range of traces are to be copied to insure that only the traces desired are copied. *ORDER is necessary because traces are not ordered sequentially in *any* library when they are stored, including the "target" library in a *COPY instruction.

The *LOAD and *UNLOAD instructions which have been added to the instruction set help eliminate the number of computer runs required when data traces are to be selected from many libraries to be merged into one library or formatted for output. These instructions have the general format

*LOAD,<library>.

*UNLOAD,<library>.

where <library> may be either MASTER, ALTERNATE, or RANOUT.

The *UNLOAD instruction causes a sequential file of the appropriate <library> to be written to the sequential output file TAPE7. If data trace libraries are to be saved on magnetic tape, this instruction is necessary to format the library so that it may be accessed at a later time with the *LOAD instruction.

The *LOAD,<library>. instruction causes the library associated with <library> (if any) to be detached from the program and a new library read from TAPE4 (ALTINPUT).

The old <library>, if not saved with a *UNLOAD, is lost.

Examples.

*LOAD,MASTER.

*LOAD,ALTERNATE.

*MERGE.

*UNLOAD,RANOUT.

In the proceeding example, the libraries MASTER and ALTERNATE are converted from sequential format on TAPE4 (ALTINPUT) to the format used by RANLIB. The library RANOUT is created by merging MASTER and ALTERNATE and is converted to sequential format on TAPE7 by the *UNLOAD instruction where it will be saved on magnetic tape.

Suppose the user wants to list the traces contained in several libraries stored on magnetic tape. The tape has been associated with TAPE4 prior to executing RANLIB. The following sequence of instructions will list the contents of the first six libraries on the magnetic tape.

*LOAD,MASTER.

*LOAD,ALTERNATE.

*LOAD,RANOUT.

*ORDER.

*LIST,,DETAIL.

*LOAD,MASTER.

*LOAD,ALTERNATE.

*LOAD,RANOUT.

*LIST,,DETAIL.