The Pennsylvania State University
Computation Center

Contributed Program
Contributed by John B. Smith
Department of English

BAG/2: A Bibliographic and Grouping System for Natural Language Data

November, 1981

PREREQUISITES:

A familiarity with the structure and form of bibliographies is assumed. In addition, knowledge of a conversational data entry system such as INTERACT [2] will be helpful. If data are to be kept on magnetic tape, familiarity with tape handling as described in [4] is required.

A form for readers' comments is provided at the back of this documentation. We urge the use of this form, to assist us in improving Computation Center services.

TABLE OF CONTENTS

Chapter		Page
I	OVERVIEW	1
II	CONFIGURATIONS	3
III	USE	4
	Building and Updating a Data Base	4
	SCAN	4
	SORT	5
	TRANSFER	6
	Using the Data Base	7
	SEARCH	7
	INDEX	8
	CONVERT	9
IV	DATA PREPARATION	10
	Planning	10
	Subject Taxonomy	10
	Form Taxonomy	11
	First Field	12
	Conventions	13
	Item Format	13
V	OPTIONS	14
	SEARCH Options	14
	SEARCH Example	20
	INDEX Options	21
	INDEX Example	23
CONTACT		25
REFERENCES AND ACKNOWLEDGMENTS		25

Chapter I
OVERVIEW

BAG/2 is a system of programs intended primarily to handle bibliographic citations, but is suitable for any data base that structurally resembles a bibliography -- that is, a collection of individual records consisting of identifiable components (author, title, etc.) ordered alphabetically under a subject taxonomic scheme (e.g. algebra, analysis, ... , topology). The system consists of five separate programs: three are used, in sequence, to <u>SCAN</u> the data, <u>SORT</u> them, and to <u>TRANSFER</u> the sorted data to the MASTER file; two are used to <u>SEARCH</u> and extract from the data base the information requested, and to <u>INDEX</u> the data base.

Data are presented to the system in 80-character lines in free format, with subject taxonomic class indicated at the beginning of each new record and form taxonomy markers included within the data; records are processed internally as variable-length records with a maximum length of 4,000 characters. You inform the system of your requirements through a control language using a consistent syntactic form. SEARCH program options include output format control for printed results, full Boolean search capabilities on subject and form taxonomies as well as content, suppression and rearrangement of record components, and resorting of retrieved records on any field(s). The INDEX program can produce up to nine separate, sorted indices to the master file on the basis of designated record component fields specified at output. Figure 1 shows a typical data flow through the system.

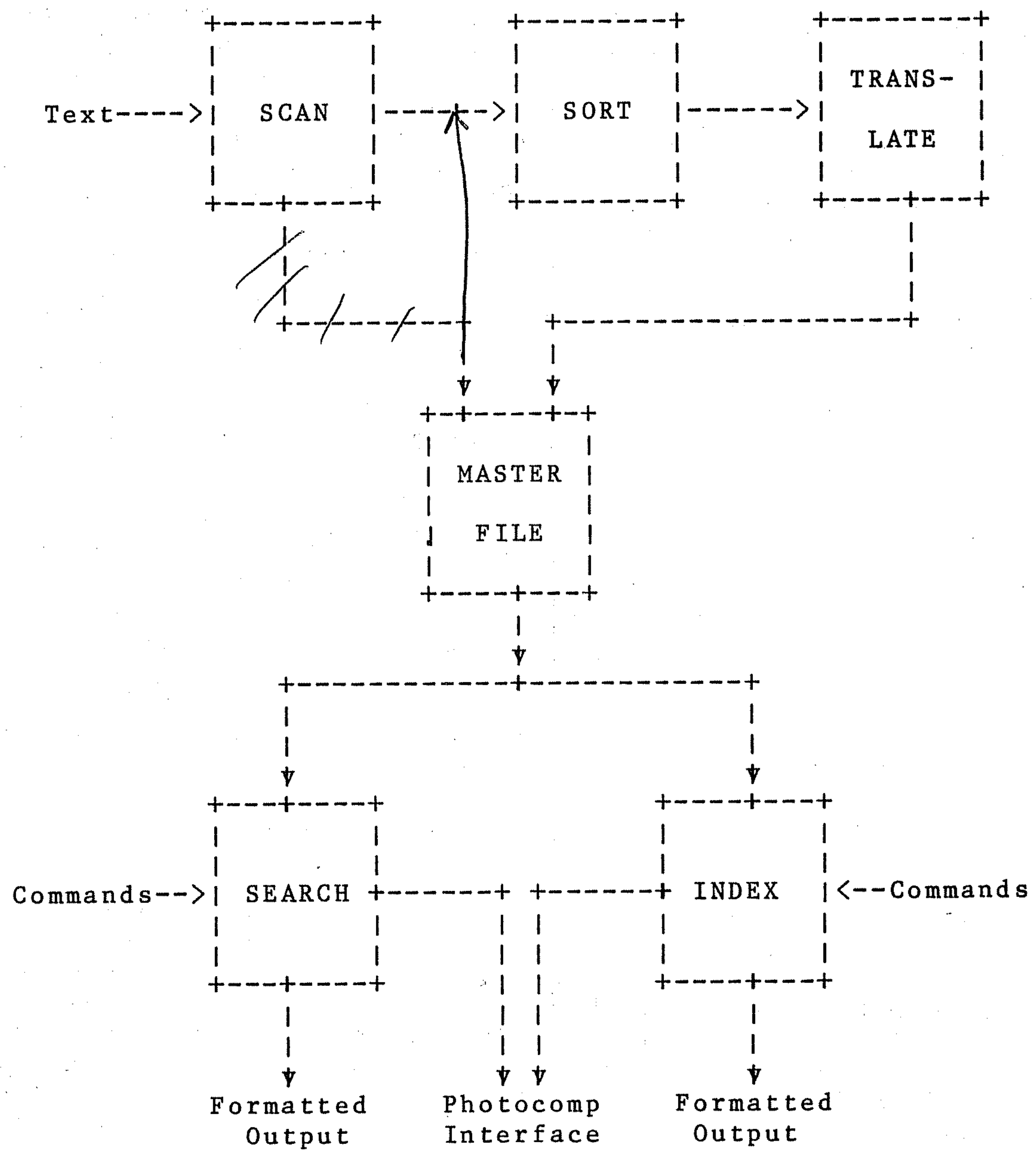
The system appears quite efficient: a recent (Fall, 1981) benchmark on a file of some 1,283 records, each an average of 300 characters, run on an IBM 3033, resulted in the following costs:

Construction of the data base master file:	$1-2
Retrieval (Boolean expression with five keys):	$.50

Although BAG/2 is a successor to the earlier BAG system [1], data files are not compatible between the two. The program CONVERT can be used to restructure records that have been processed by BAGSCAN to make them acceptable to BAG/2 programs. Many users, however, will wish to edit records further to take advantage of BAG/2's hierarchical subject taxonomy capability. The COPYREC option of SEARCH, described in Chapter V, can be used to deposit restructured records into an INTERACT file suitable for editing.

Figure 1

BAG/2 Organization



Chapter II
CONFIGURATIONS

The system can be used in several configurations. While the programs <u>SCAN</u>, <u>SORT</u>, and <u>TRANSFER</u> can be run in a single job with three steps, it is recommended that they be configured separately to ensure their verified sequential execution, and thus reduce the vulnerability of the master file. Though not described here, some form of backup for the master file as well as the keyed data is highly desirable.

The job control statements necessary to execute each program are provided below. The Master File as well as intermediate files for the SCAN and SORT steps are shown as members within a partitioned data set named "bagdata", located on an INTERACT disk; many users, however, will wish to configure the system using tape data sets. The writeups "JCL Techniques" and "PSU Magnetic Tape User's Guide" will be helpful in this case; if you need further help with these details, seek assistance from a consultant at the Computation Center to set up the various jobs to meet your particular requirements.

Chapter III
USE

In the Job Control Language (JCL) shown in this chapter, many input and output data sets are described as INTERACT files. In these statements, as in all of the JCL, information in upper case must be entered as shown, but information in lower case is to be supplied by you. In particular, the letters "gggggg" indicate your INTERACT group (account number), and "iii" indicates your INTERACT initials. The file names shown in lower case may be anything of your choosing; where a file is shown as a member of a partitioned data set (its name is in parentheses), it may be a regular sequential data set instead.

If you are using magnetic tape to store your files, the JCL statements for many input and output data sets will be entirely different; consult the writeup "JCL Techniques" for details.

Building and Updating a Data Base

SCAN

SCAN has the primary function of taking as input, keyed items in 80-character, free format lines and producing as output a variable-length record, one for each item, with multiple blanks eliminated at the ends of lines. One control option is available: by default, SCAN does not produce an accompanying printed version of the data; if a printed version is desired, specify PRINT through the CONTROL file. Output is likely to be directed to a disk or tape file. With either option, you should verify successful completion of the SCAN step before proceeding to the SORT step.

Input:

```
//     job statement (not Category W)
// EXEC BAG,PROG=BAG2SCAN
//CONTROL DD *
PRINT           {Optional; default is not to print}
//INPUT   DD *
          .
input data (typed lines, 80-character records)
          .
//OUT      DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//            DSN=MEN.gggggg.iii.bag2data(scandata),
//            DCB=(RECFM=VB,BLKSIZE=6400)
```

Output:

OUT will contain variable-length records, to be passed to SORT. If member "scandata" existed before, it will be replaced.

NOTE: The DISPosition parameter OLD must be replaced by NEW if the data set "bag2data" did not exist before this run.

SORT

SORT receives as input the variable-length records created by SCAN, sorts them, and merges them with the master file. Sorting is keyed to the first 100 characters of each record. Exact duplicate records are reduced to a single item. Again, output may be directed to tape or disk files. With either option, you should verify successful completion of the SORT step before proceeding to the TRANSFER step. Note that the input file IN consists of two concatenated files: the newly scanned records and the old master file. For the initial run, the reference to the nonexistent master file should be deleted.

Input:

```
//      job statement (not Category W)
// EXEC BAG,PROG=BAG2SORT
//IN   DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.bag2data(scandata)
//     DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.bag2data(master)
//OUT  DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.bag2data(sortdata),
//          DCB=(RECFM=VB,BLKSIZE=6400)
```

where the input data set is the output from the SCAN step, concatenated with the old master file.

Output:

OUT will contain variable-length records, sorted alphabetically, to be passed to TRANSFER. If member "sortdata" existed before, it will be replaced.

TRANSFER

TRANSFER receives the newly sorted records and rewrites the master file. Individual records may be removed during the rewrite process and, optionally, placed in a file for editing and eventual reinsertion in the data base.

A list of record numbers in ascending order for items to be removed from the master file is presented through the file INPUT. Each number should be typed on a separate line, right-justified in columns 1-5; if the record is also to be copied to a file for editing, an asterisk (*) should be placed in column 6. These records, broken into approximately 60 characters on 80-character lines, will be written to the file BFILE, assumed to be a file accessible from a terminal. After editing, these can be included with the next batch of records at the SCAN step.

Input:

```
//      job statement (not Category W)
// EXEC BAG,PROG=BAG2TRAN
//IN     DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.bag2data(sortdata)
//INPUT DD *
             .
   Numbers of records to be deleted from Master File or
   to be removed and put in "editfile" for editing.
             .
//OUT    DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.bag2data(master),
//          DCB=(RECFM=VB,BLKSIZE=6400)
//BFILE DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.LIB(editfile),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12960)
```

where IN is the variable-length records produced by SORT; the record numbers in INPUT must be in ascending order.

Output:

OUT will be the new master file consisting of variable-length records, and will overwrite the old master. We recommend that, before you run this TRANSFER job, you use the IBM utility program IEBCOPY to copy your old master file (see [5]); it will then still be available in case TRANSFER is unsuccessful for some reason.

BFILE will be the 80-character lines of designated records removed from the master file, and will replace any previous version of member "editfile".

Using the Data Base

SEARCH

SEARCH is the primary access program for the data base. It permits printing, under format control, of the entire data base or selected items matching a search expression. Output may be printed on a high-speed printer or returned to the terminal. So that you may get an unimpeded view of the whole system, detailed discussion of each option will be described later in Chapter V, "Options".

Input to SEARCH is the master file, produced by the successful completion of the three sequential steps: SCAN, SORT, and TRANSFER. Control information is provided through the file CFILE; headings for the subject taxonomy may be provided through the file HFILE. Output is directed to the file PFILE for terminal or printer, in visual format; selected records can be placed in BFILE.

Input:

```
//      job statement (not Category W)
/*JP FULLSKIPS,UCS=TN,FORMS=16
// EXEC BAG,PROG=BAG2SRCH
//IN    DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//         DSN=MEN.gggggg.iii.bag2data(master)
//CFILE DD *
            .
   control statements for search
            .
//HFILE DD *
            .
   subject headings
            .
//PFILE DD SYSOUT=A,DCB=BLKSIZE=132
//BFILE DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//         DSN=MEN.gggggg.iii.LIB(srchedit),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=12960)
```

where IN is the variable-length records of the master file, CFILE contains all control statements, and HFILE is the file for insertion of headings for subject taxonomy.

Output:

PFILE will be the printed output file. NOTE: If option ACCENTS has been specified (see Chapter V), replace "UCS=TN" on the /*JP card shown with "UCS=AL"; this will cause the printing to be done with the ALA print train, which contains all of the characters described in Chapter V.

BFILE will be the selected records; if member "srchedit" existed before, it will be replaced.

INDEX

INDEX produces from one to nine separate sorted indices for the master file, created from various form taxonomy categories. You may specify at run time the categories to be used to create each separate index. Indices are assumed to be of two kinds: names of people (in index form) and descriptive terms, usually from a controlled vocabulary thesaurus. Input is the master file, control statements, and a separate file of cross reference terms to be added. Output is the printed set of indices requested. Note that this is a three-step job: the program BAG2INX1 produces the indices and passes them to the IBM SORT program, which sorts them on three fields and passes the results to the program BAG2INX2 for formatting and printing.

Input:

```
//      job statement (not Category W)
/*JP FULLSKIPS,UCS=TN,FORMS=16
// EXEC BAG,PROG=BAG2INX1
//CFILE DD *
         .
   control statements
         .
//CREFF DD *
         .
   cross-references
         .
//IN     DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//          DSN=MEN.gggggg.iii.bag2data(master)
//TEMP   DD UNIT=SYSDA,DSN=&&TEMP,SPACE=(TRK,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//          DISP=(NEW,PASS)
//OUT    DD UNIT=SYSDA,DSN=&&INDX,SPACE=(CYL,(10,10)),
//          DCB=(RECFM=FB,LRECL=100,BLKSIZE=3000),
//          DISP=(NEW,PASS)
//*
// EXEC SORT       {cataloged procedure for IBM SORT/MERGE}
//SORTIN  DD UNIT=SYSDA,DSN=&&INDX,DISP=(OLD,DELETE)
//SORTOUT DD UNIT=SYSDA,DSN=&&INDXS,SPACE=(CYL,(10,10)),
//           DCB=(RECFM=FB,LRECL=100,BLKSIZE=3000),
//           DISP=(NEW,PASS)
//INPUT   DD *
  SORT FIELDS=(1,2,FI,A,7,94,CH,A,3,4,FI,A)
//*
// EXEC BAG,PROG=BAG2INX2
//TEMP   DD UNIT=SYSDA,DSN=&&TEMP,DISP=(OLD,DELETE)
//IN     DD UNIT=SYSDA,DSN=&&INDXS,DISP=(OLD,DELETE)
//PFILE DD SYSOUT=A,DCB=BLKSIZE=132
```

where IN is the variable-length records of the master file, CFILE contains the control statements, and CREFF is the file of 80-character typed cross references.

Output:

The output from the final step, PFILE, is the printed output file. Note that, as with SEARCH, the output should be printed with the ALA print train if Option ACCENTS was specified.

CONVERT

CONVERT takes data processed by the earlier BAG system [1] and converts the records to a format suitable for BAG/2 programs. Input is any BAG data set that has passed through _at_ _least_ the BAGSCAN program (i.e. CONVERT will _not_ accept raw data as originally typed). Output is a data set suitable for use with any of the BAG/2 programs beyond SCAN. If you want to edit records further, perhaps to redefine the subject taxonomy codes, the COPYREC option of SEARCH can be used to deposit records in an INTERACT file (#srchrec in the JCL configuration for SEARCH provided above) in a form suitable for editing and eventual re-introduction to BAG/2 through SCAN.

Input:

```
//      job statement (not Category W)
// EXEC BAG,PROG=BAG2CNVT
//IN  DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(OLD,KEEP),
//         DSN=MEN.gggggg.iii.master.bag
//OUT DD VOL=REF=MEN.gggggg.iii.LIB,DISP=(NEW,CATLG),
//         DSN=MEN.gggggg.iii.master.bag2,
//         SPACE=(TRK,(10,10),RLSE),
//         DCB=(RECFM=VB,BLKSIZE=6400)
```

where IN is the variable-length records from BAG.

Output:

OUT is the variable-length records for BAG/2.

Chapter IV
DATA PREPARATION

Planning

As with any computer project or system, a project using BAG/2 should devote considerable attention to organization of information, retrieval requirements, representation and presentation formats, and encoding conventions. These decisions, in turn, should be considered in the context of the overall project goals and project structure. Not all relevant planning matters can be addressed here, but several of the more important and necessary decisions are discussed below.

Subject taxonomy

The data base will be ordered by BAG2SORT in ascending alphabetic order. The system assumes, however, that each record or item is preceeded by a subject taxonomic marker that is included in the sort field. Of course, all items may be placed in the same taxonomic class, resulting in a single, top-to-bottom alphabetic ordering of the items. More often, the subject taxonomy feature is used hierarchically to divide the data base into separately ordered and separately extractable sections. The subject taxonomy marker consists of the reserved character (%) followed by two-digit categories separated by periods.

Form: %XX.XX.XX.XX

Example: %30.14.05.05

In the above example, the subject taxonomy is defined for four levels: all items with the first field less than 30 precede this item; within the cluster of items with first field 30, this item occurs after those with a second field less than 14 and before those with a second field greater than 14; within the 30.14 cluster, this item will occur after those with a third field less than 05 and before those with a third field greater than 05; etc. In an example involving a large (50,000 item) bibliography of research pertaining to Shakespeare, the following scheme was used: the first field divided the bibliography into

10 General Shakespeareana
20 Play Groups
30 Individual Works

The second field indicated, within the 30 group:

02 All's Well That Ends Well
04 Antony and Cleopatra
•

```
          .
          .
          14 Hamlet
          .
          .
          .
          92 Winter's Tale
```

The third field indicated, within the 30 group:

```
          05 Scholarship and Criticism
          10 Production and Staging
```

The fourth field indicated, within the 30 group and within the third level 05 group:

```
          05 Bibliographies and check lists
          10 Editions and texts
          .
          .
          .
          30 Criticism
```

Thus, the subject taxonomy marker

     %30.14.05.05

would begin each item dealing with the Individual Work Hamlet, Criticism and Scholarship, and bibliography or check list; all such items would be grouped and ordered alphabetically by what follows in the record.

Note that records prepared for the earlier BAG system and CONVERTed to BAG/2 format have only one hierarchical level of subject categories. Additional levels can be added manually to records deposited in an INTERACT file (#srchrec in the JCL configuration provided above for SEARCH) through the COPYREC option of the BAG/2 program, SEARCH (see Chapter V, "Options").

Form taxonomy

Within each record, separately identifiable fields should be marked with a form taxonomy marker. Typically, these would designate: author, title, journal, date of publication, descriptor terms, etc. For a reference guide for such divisions, you should compare BAG/2 form taxonomy classes with the MARC format developed by the Library of Congress; it is not necessary that the two systems match, but only that the BAG/2 system could be transformed algorithmically to the MARC format. In general, BAG/2 form taxonomy capability can support a wide variety of internal structures, including those equivalent to MARC, but in a more flexible and convenient manner. Over-division of the record into identifiable fields is preferable to under-division of the record. Since fields can be suppressed (described below), separate identifiable fields for confidential data can be added, used for retrieval, and then suppressed for output. Since the contents of a field can collectively be designated for underscoring, separate identification for separately published works (books) and contained works (chapters or journal articles) is desirable.

Form: %.XX

Example:	%.10	Author, index form
	%.15	Author, substitute print form
	%.20	Title, separately published
	%.25	Title, contained
	%.26	Cross reference
	%.30	Imprint statement
	%.35	Date (suppressed)
	%.40	Annotation
	%.50	Reviews
	%.61	Names for index
	%.64	Descriptive terms for index
	%.80	Accession number

These form categories are taken from the same Shakespeare bibliography project mentioned above; while not complete, the scheme illustrates the general principles of field identification. Most should be self-explanatory; the field %.15 is perhaps not. Names in field %.10 are always in index form (Last, First Middle) and, as such, determine the sorted order of items within subject taxonomy groupings. One option, described below, allows the second and subsequent names of multiple authorship items to be unscrambled by the system to appear in regular bibliographic order (First Middle Last). For names that cannot be unscrambled, or for inflected names, the exact form for printing may be given in %.15 and then substituted for the name in field %.10.

First Field

As the above example indicates, items within subject groups will be ordered on the basis of the first form taxonomy category and its contents. Thus, care should be exercised to insure that order of words, capitalization, and consistency of arrangement of categories will result in the ordering desired. For this reason, a complete and detailed specification of subject taxonomy, form taxonomy, and encoding conventions down to the level of explicit punctuation should be developed. It is advisable for those with little computer experience to discuss these specifications with someone familiar with sort protocols or someone familiar with BAG/2. As always, a sample or pilot run that fully tests subject and form taxonomies is recommended before full data entry.

Conventions

Reserved symbols:

1. % global, marks subject or form category.
2. {} contextual, marks shift in font (underlining)
3. <> contextual, marks accent specifications

Item Format

Subject category   data item   %end

Begin the item with the subject category: %XX.XX.....XX; type the data item as it should appear, including all punctuation, with form categories inserted: %.XX; end the item with the following marker: %end.

Example:

```
%10.05
%.10 Andrews, John,
%.11 ed.
%.20 Shakespeare Quarterly
%.30 26 (1975), 27 (1976).  Wash., D.C.:  Folger Shak. Lib.
%.35 1975-76
%.80 U328
%end
```

Comments:

As the above example indicates, items may be broken anywhere there would be a space in the record; for proofreading purposes, each new form category may begin on a new line.

Chapter V
OPTIONS

After the data base has been built and resides in the master file, the SEARCH and INDEX programs can be used to access it. Most applications rely more heavily on the SEARCH program for day-to-day retrieval; the INDEX program is then used to prepare an index to the master file when it is at some stable point of development, usually for the purpose of publication or some other "set" use. Options for both programs are specified through the file CFILE, and use a uniform syntax:

```
control_word:  p1,p2,......,pn;
or
control_word = p1,p2,......,pn;
```

That is, each option is associated with a specific control_word; commands are separated from one another by a semicolon; commands may have associated with them a list of parameters separated from the control_word by either an equal sign or a colon, and separated from one another by commas.

<u>Example:</u> SUPPRESS: %.23,%.29,%.34;

Each form category indicated would be suppressed on output.

<u>SEARCH Options:</u>

1.	PRINTBIB	Printed, standard bibliographic output; no parameters required.
2.	SELECT_CATS	Specifies a list of the categories, and only the categories, that will be printed. They will be printed in the order listed in the parameter list, thus permitting reorganization of the components of each data record for printing.
3.	PRINTREC	Prints the total record, including all taxonomic category markers, as a long character string.
4.	COPYBIB	Copies the items selected in the file BFILE, in bibliographic format.
5.	COPYREC	Copies the records in the file BFILE with all taxonomic markers included. This option can be used to deposit records in an INTERACT file suitable for editing and later re-introduction through SCAN.

6. SEARCH Indicates that a search is to be performed for the search expression that follows it. A search expression can be any valid Boolean expression with up to 40 operators and content terms. Permissable operators are: (& | ¬). Content terms may include subject taxonomy category, form taxonomy category, and content strings with or without associated form taxonomy.

<u>Examples:</u>

```
SEARCH:  %30.14 | Hamlet;
```

This search would retrieve all records in the primary subject taxonomy category for the play <u>Hamlet</u>, as well as any record in any other category containing the word "Hamlet".

```
SEARCH:  %.20Hamlet;
```

This search would seek all items which contain the word "Hamlet" in the title of a book (occurrence anywhere else will be ignored).

7. SORTCAT Indicates that the category or categories that follow are to be appended to the front of the item and the selected group resorted; before printing, the attached fields are removed. Thus, one may search for a group of items, resort them on, say, date, and then print the selected group in normal bibliographic form.

8. SKIPCAT Printing will skip to a new line for the categories specified.

9. PAGESIZE Page size will be the number of lines indicated.

10. LINESIZE Line size will be the number of characters indicated; maximum: 131.

11. SKIPPAGE Printing will skip to a new page when the contents of the specified form categories change.

12. SORTLENGTH If SORTCAT is specified, items will be, by default, resorted on the basis of the first 100 characters of the item; this value may be altered with this command.

13. SUPPRESS The form categories specified will not be printed.

14. HEADINGS

Headings will be printed whenever the subject taxonomic category of an item differs from that of the preceding item. Headings must be provided through the file HFILE, following a rather rigorous convention. Lines are of two kinds: the actual heading with its corresponding form code, and lines that indicate control information. Each line that contains control information begins with a percent (%). Each line that contains an actual heading begins with the code, followed by at least one space and then the heading as it is to be printed. Maximum length (including internal blanks) for a heading is 40 characters.

The first line contains global heading information:

a. the percent sign
b. the total number of headings
c. the maximum hierarchical depth
d. the position relative to line size where each hierarchical level of heading is to be printed. Positive values indicate actual columns; negative values indicate centering at the reciprocal of the value across line size.

<u>Example:</u>

```
% 178 4 -2 -2 9 1
```

Meaning:
178 headings
4 hierarchical levels
center the first two heads; begin the 3rd level head in column 9 and the 4th level head in column 1.

Before each heading, insert a control line that gives the number of headings in the next hierarchical level <u>below</u> the heading that is about to be given.

<u>Example:</u>

```
% 178 4 -2 -2 9 1
% 3
%11
10 General Shakespeareana
```

In this example, the first line is as explained. The second line indicates that the first hierarchical level will have 3 heads (the first of which can be seen to be <u>General Shakespeareana</u>); the first head of level 1 will have 11 second level heads. The first level head, itself, is marked by the category 10 and is to be printed, General Shakespeareana, centered. To continue the example:

```
10 General Shakespeareana
%5
20 Play Groups
%46
30 Individual Plays
```

The second first level head, Play Groups, will have 5 second level heads; the third first level head, Individual Plays, will have 46 second level heads. The end of a hierarchical level is marked, %0. To continue the example:

```
30 Individual Plays
%0
.05 Festschriften
.10 Bibliographies
.15 Biography
.20 Editors
%4
.25 Reference Works
```

Having completed all 3 of the first level heads, we begin with the second level heads, from the top, and work through them. For the first four, there are no third level heads, but the fifth, Reference Works, is followed by some 4 third level heads which will be given after all other second level heads are specified. The process continues, top-to-bottom, one level at a time, until all heads are given. One final convention is provided for convenience. A negative value, instead of a positive value, in a head control statement indicates that heads for that item are to be picked up from the nth preceding head. Thus, for a series of heads having the same next lower divisions interrupted by an anomalous head with different next lower divisions, the pattern can be re-established by indicating the number of heads back up the list where the pattern may be picked up. For example:

```
%2
.02 Alls Well
   .
   .
   .
.56 Othello
%1
```

```
.58 Passionate Pilgrim
%-2
.60 Pericles
```

In this example, individual plays are normally followed by two heads at the next lower level. Passionate Pilgrim, however, has only one head at the next lower level. Pericles returns to the normal conventions for plays, in this case the next lower head structure associated with Othello, two heads back up the list.

15. UNDERLINE

The contents of the categories specified will be underscored. Shifts to Roman or shifts to Italics may be indicated in the data itself for particular strings by placing the string inside of curved braces with an "r" or an "i" immediately after the first brace.

<u>Examples:</u>

UNDERLINE: %.20;

All contents of category %.20 will be underlined.

{i Shakespeare Quarterly}

Only the words <u>Shakespeare</u> <u>Quarterly</u> will be underlined.

16. ACCENTS

Indicates that accents are contained in the data; no parameter is called for. This command assumes that accents have been indicated in the following way: immediately after the word containing accents appear the less than symbol (<), pairs of characters indicating position and accent, followed after all such pairs by the greater than symbol (>). For each accent, the first character of the pair within the < > symbols indicates the number of characters preceding the less than symbol where the accent appears, counting 1-9, a-z for 1-35; the second character in each pair indicates the particular accent selected from the following list that appears in the position indicated.

1	acute
2	grave
3	circumflex
4	umlaut
5	cedilla
6	tilde
7	breve
8	crossed ell
9	over dot
a	macron

b	angstrom
c	double acute
d	left hook
e	hachek
f	superscript
g	under dot
h	right hook
j	thorn
k	ligature
l	crossed dee

The appropriate characters will be printed (or over-printed) in the specified positions if the ALA print train (UCS=AL) is specified on the /*JP card.

17. SUBSTITUTE — This command is followed by triples: %.XX for %.YY, ... The category %.XX will be substituted for %.YY if %.XX appears in the record; otherwise, %.YY will be printed. Among other uses, this replacement feature allows names always to be listed in index order in %.YY but replaced by %.XX in cases where for some reason an alternate form is desired or, for multiple authors, one of the subsequent names can not be inverted.

18. NODUP — For the categories specified, duplicate contents in succession will be replaced in the second record and all subsequent records by five underscores; for multiple adjacent titles by the same author, this convention permits second and subsequent author fields to be replaced by underscores.

19. INVERT_NAMES — For the categories specified, the first name of a multiple name category is left in index form (Last, First Middle); all subsequent names are inverted into conventional order (First Middle Last).

20. MASTER_NUMBERS — Accompanying each record will be its position in the masterfile.

21. RELATIVE_NUMBERS — Accompanying each record will be a running number, from 1 to the number of items printed.

22. SPACING — The number specified is the number of blank lines between items printed.

<u>SEARCH Example:</u>

The following example illustrates a number of the options described. Listed are three components: the actual commands supplied to the SEARCH program, the messages reported back by SEARCH, and the actual items retrieved:

<u>Commands</u>

```
HEADINGS;PRINTBIB;ACCENTS;MASTER_NUMBERS;LINESIZE=55;
SUBSTITUTE: %.15 FOR %.10; UNDERLINE: %.20;
SKIPCAT: %.50;INVERT_NAMES: %.10;
SUPPRESS: %.35, %.61, %.62, %.63, %.64, %.80;
SEARCH: %30.14 | Hamlet;
```

<u>Program Messages</u>

```
     ITEMS WILL BE PRINTED IN BIBLIOGRAPHIC FORMAT
     A SEARCH WILL BE PERFORMED FOR THE FOLLOWING VALUES:
          %30.14
          |
          Hamlet
          DUMMY LOGICAL EXPRESSION FOR SEARCH IS: 0|0
     PRINTING WILL SKIP TO NEW LINE ON THE FOLLOWING CATEGORIES:
          %.50
     LINESIZE WILL BE  55 SPACES
     THE FOLLOWING CATEGORIES WILL BE SUPPRESSED:
          %.35
          %.61
          %.62
          %.63
          %.64
          %.80
     SUBJECT HEADINGS WILL BE PRINTED
     THE FOLLOWING CATEGORIES WILL BE UNDERLINED:
          %.20
     ACCENTS WILL BE PRINTED
     THE FOLLOWING SUBSTITUTIONS WILL BE MADE:
          %.15 FOR %.10
     NAMES WILL BE INVERTED IN THE FOLLOWING CATEGORIES:
          %.10
     NUMBER OF ITEM IN MASTER FILE WILL BE PRINTED
```

<u>Retrieved Items</u>

GENERAL SHAKESPEAREANA

Festschriften and Analyzed Collections

25 *Erzgraber, Willi. Hamlet <u>Interpretationen</u>. (Wege der Forschung 214.) Darmstadt: Wissenschaftliche Buchgesellschaft, 1977. vi + 570 pp. [See <u>SQ</u> Bibliog. for 1977, Item 22.]
Rev.: Heuer, Hermann. SJH, 114 (1978-79), 349.

Language, Linguistics, Philology

Grammar and Syntax

246 Dent, R. W. "Hamlet: Scourge and Minister." s.v. Ham./Scholarship/Criticism

Stylistics

266 Brody, Jules. "Freud, Hamlet,. . . ." s.v. Ham./Scholarship/Criticism

Translation

279 "Hamlet on Sale in China." s.v. Ham./Scholarship/Translation

Productions, Stage History

Actors, Acting, Directing

363 "If Actors Had Their Choice of Roles. . . ." N.y. times, 18 June 1978, Sec. 2, pp. 1, 11. [John Cullum opts for Coriolanus and Lear; Geraldine Fitzgerald for Cordelia and Nora Melody in O'Neill's Touch of the Poet, "O'Neill's retelling" of Lr.; Joel Grey for Richard III; Barnard Hughes for Wolsey in H8 and Gloucester in Lr.; Raul Julia for Othello; Eartha Kitt for Cleopatra; Frank Langella for Hamlet; Jack Lemmon for Hamlet; Lawrence Luckinbill for Macbeth; Estelle Parsons for Cleopatra; other Shak. mentions.]

Stage and Theatre History

421 Bartoshevich, A. V. "The Political Decade in England and Shakespeare." Teatr, 8 (1978), 119-26. [British prods. of 1930s dominated by the idea of "political art." Olivier's Hamlet and "The Oxford poets." Prods. of the period in U.S. and Germany. In Rus.]

INDEX Options:

INDEX is a three-step process that can produce from 1 to 9 separate, alphabetized indices on each run. INDEX uses the same command structure as SEARCH:

```
command_word  : or =   p1,p2,.....pn;
```

Indices are currently assumed to be of two types: names (contained in an identifiable form category in index form [Last, First Middle]), and descriptive terms (probably from a controlled vocabulary thesaurus). The following options are available:

1. INDEX1 — This command, and by analogy INDEX2 through INDEX9, specifies the form categories from which the index is to be constructed. If the category is a name category, place an N immediately after the four-character category marker or the symbol that separates the name from the remainder of the form category. This last feature can be used, for example, to separate the name of a reviewer from the title and/or reference information. Multiple name components within the same form field are assumed to be separated by semicolons. The absence of an N or the name delimeter symbol implies that the index is to be formed from single words, separated by commas.

   Example:
   ```
   INDEX1: %.10N,%.50|,%.61N;
   INDEX2: %.64;
   ```

   In the first line, you have consolidated names from three separate fields into a single index; %.10 and %.61 fields have names in standard index form, while field %.50 has names, also in index form, but contains additional information after each name, separated by the symbol |. In the second line, you have indicated that a separate index will be drawn from the contents of form category %.64; items in %.64 are asssumed to be separated from one another by commas.

2. LINESIZE — Line size will be the specified number of print positions per output line.

3. PAGESIZE — Page size will be the specified number of lines per page.

4. UNDERLINE — Indicates that some fields may contain words with manual underscoring indicated ({i word}); see description in SEARCH options for additional details. Underscores will be printed.

5. ACCENTS — Indicates that some fields may contain accented words; see description in SEARCH options above for details. Accents will be printed if the ALA print train (UCS=AL) is specified on the /*JP card.

6. CROSS_REF

Indicates that cross references are supplied for the index given.

<u>Example:</u>

```
CROSS_REF = 2;
```

Actual cross references are assumed to be supplied through the file CREFF; each cross reference should consist of three parts: the term as it would appear in the data, a comma, and the reference that should be printed with the primary term in the index.

<u>Example:</u>

```
counsel, see advice
allegory, see also type
```

If the first term appears in the data, the phrase after the comma is placed between the primary term and the numeric references to the master file; if the primary term does not appear in the data, the entire cross reference is printed.

7. HEADINGS

Headings can be supplied through the file HFILE. Since there are no hierarchical divisions within an index, simpler encoding conventions are used than those for SEARCH. Separate headings may be supplied for each of the 9 indices, and each may have up to 4 lines of text. Each line of each heading should appear on a line by itself; preceding each heading or heading group must be a number, on a line by itself, that indicates the number of lines in the heading.

<u>Example:</u>

```
1
Author Index
2
Names of Actors, Actresses, Directors,
Producers, Scene Designers, and Others
```

<u>INDEX Example:</u>

<u>Commands</u>

```
INDEX1: %.10N, %.50|, %.61N;
INDEX2: %.64;
ACCENTS;
CROSS_REF = 2;
```

Program Messsges

```
OPTIONS FOR THIS INDEX:
     INDEX 1 COMPOSED FROM THE FOLLOWING CATEGORIES:
          %.10N
          %.50|
          %.61N
     INDEX 2 COMPOSED FROM THE FOLLOWING CATEGORIES:
          %.64
     ACCENTS WILL BE PRINTED
     CROSS REFERENCES WILL BE INCLUDED FOR INDEX 2
```

INDEX1

```
  Aaron, Jules, 2459.
  Abartis, Caesarea, 1174.
  Abdel Sayed, Samiha Yassa, 264, 951, 1424.
  Abraham, Claude K., 70.
  Abrams, Richard, 2647.
  Adams, Barry B., 2777.
  Adams, Caesar Blair, 1141.
  Adams, Percy G., 251.
  Adams, Robert E., 546.
  Adamson, William DeLancey, 1407, 1460, 2252, 2287, 2853.
  Adelman, Janet, 1262, 1367, 1828.
  Aden, John M., 251.
  Ades, John I., 810.
```

INDEX2

```
  absurd, 508, 960, 1909, 1910.
  act and scene division, 412, 621, 710, 1197, 1278, 1538, 2381.
  aesthetics, 123, 242, 278, 299, 376, 400, 416, 643, 739, 791,
    805, 811, 834, 843, 970, 2219, 2336, 2570, 2595, 2746, 2840.
  agon, see also conflict.
  agriculture, 150.
  alchemy, 1479, 1503.
  alcohol, 755.
  alienation, 2484.
  allegory (see also trope), 131, 689, 839, 1209, 1860, 1879,
    1899, 2195, 2659, 2679, 2683, 2707, 2743, 2859, 2861, 2867.
  allusions, 161, 237, 265, 281, 580, 592, 619, 714, 835, 895,
    1276, 1479, 1742, 2219, 2843.
  ambiguity (see also ambivalence), 957, 1379, 1576, 1672, 1674,
    1742, 2105.
  ambition, 2029, 2423.
  ambivalence (see also ambiguity), 673, 1156, 1484, 1579, 1581.
```

CONTACT:

Technical questions and problems may be directed to any member of the Computation Center Applications Staff. Questions concerning distribution of the BAG/2 system should be directed to the author:

John B. Smith
Department of English
The Pennsylvania State University
University Park, PA 16802
(814) 865-9681 (office)
(814) 422-8486 (home)

REFERENCES AND ACKNOWLEDGMENTS:

BAG/2 was written by John B. Smith of the Department of English at Penn State University; Dr. Smith also prepared this documentation on the use of the system, with the editorial assistance of Barbara Kautz of the Computation Center faculty.

Except for [5], the following are all publications of the Computation Center:

[1] BAG: General Purpose Bibliographic and Grouping System.

[2] PSU Guide to INTERACT.

[3] JCL Techniques.

[4] PSU Magnetic Tape User's Guide.

[5] OS/VS2 MVS Utilities, IBM Manual GC26-3902.