

Report Generation

The Bibliographic/MARC Processing System provides a range of facilities for generating printed reports, catalog cards, and machine-readable files. Most types of output are produced by general purpose selection, sort, and print programs, according to instructions coded into option statements by the user. Certain specialized forms of output, such as a key-word-in-context (KWIC) listing, or a heirarchical and alphabetic thesaurus, are produced using programs designed specifically for those tasks. These special forms of output are described in separate chapters.

Retrieval and display capabilities

System capabilities for retrieving and displaying information from the data base include:

- * Selection of records for printing based on the presence, absence, or content of one or more variable or fixed fields, according to a selection rule containing logical operators AND and Ok.
- * Sorting of records by one or more variable and fixed fields;
- * Special handling of missing data fields, including provision for placing such records at the beginning or end, for specifying an alternate field for sorting, and for excluding such records from further processing.
- * Execution-time specification of fields to be printed, including modification of terminal punctuation and insertion of user-supplied captions.
- * Generation of report titles and page headings with (optional) page numbers.
- * Choice of single or multi-column format, with provision for keeping data for records or groups of records from being split across columns or pages.
- * Specification of column width, page size, indentions, and paragraph spacing, if default values are not desired.
- * Underscoring, dark printing (overstriking), and/or converting to upper case of data from selected fields or subfields.
- * Tabulation of records within categories and sub-categories for the preparation of summary reports.
- * Production of indexes to printed reports.

Conceptually, production of reports or other output involves four basic functions.

Selection:

Unless all records are to be included in the output product, the records desired must be selected from the master file, either by identifying the individual "did" (document identification) numbers or by specifying criteria for selection or rejection of records. Selection is generally accomplished by means of the Free Text Retrieval program (BPSRET) or the Indexed Retrieval program (THESRET). Both the Sort Key Edit (BPSSKED) and Print (BPSPT) also possess the ability to select or reject records based on the presence or absence of particular variable fields, but not on the content of field data. The Update program (BPSUPDT) when run in Select mode can select records whose "did" numbers are known.

Generation of postings:

If data from any one record may appear in more than one place in the output product (for example, a bibliography subject index which will list each citation under each of its subject designators), a copy of that record must be generated for each time it will appear (so that a bibliographic record having six subject designators will result in six copies of that record, one for each of the six subject designators). Generation of postings is accomplished through the Sort Key Edit program (BPSSKED).

Ordering:

Unless master file records are to be displayed or otherwise processed in the order of their "did" numbers, they must be sorted into the desired order. Ordering is accomplished through the IBM Sort/Merge program product (or an equivalent program) following modification of the MARC-format records by the Sort Key Edit program (BPSSKED).

Display:

If master file data is to be printed or passed in print line records to other programs or microfiche production facilities, the desired data must be displayed according to format specifications. "Display" as used in this context includes any extraction of information from the MARC-format master records. The display function is carried out by the Print program (BPSPT).

Preparation of option statements

These processing capabilities are invoked through option statements. Preparation of option statements is simplified by the many default assumptions which are made by the programs unless the user directs otherwise. Of course, the more sophisticated the report format and design, the more elaborate the option statements required.

Detailed consideration of the various program options and how they are coded into option statements is given in the chapters for those programs. Here, we will only illustrate some of the features and how they are expressed.

Consider a simple example: production of a list of titles from a MARC-format file of bibliographic citations. Assume, initially, that the titles are to be listed in their original order, that is, in the order of their "did" numbers. Such a simple report could be produced using only the Print program with these option statements:

R 050
L10
T20V245

In these statements, the "R", "L" and "T" are statement types, the "10" and "20" are (optional) statement sequence numbers, the "050" sets the column width at 50 characters, and "V245" refers to the variable field whose numeric tag is 245. If the master file had been created using the Update programs of the Bibliographic/MARC Processing System, they would contain alphabetic as well as numeric tags, and the T statement could have been written:

T20VTIL

Each MARC-format record will be processed according to these option statements, producing a report that looks like this:

Fertility and attitudes toward family planning in
Algeria, by Johan W. Eliot, et al.
Contemporary China: papers. Edited by Ruth
Adams.
The relation of performance to social background
factors among Army inductees, by Francis John
Ryan.
An Introduction to social research, by Hubert
M. Blalock.
Survey of the Alliance for Progress: problems of
agriculture, by William C. Thiesenhusen and Marion
Brown.

(Figure 1)

We would probably want to improve the appearance and readability of this report by inserting a blank line before each new title, and perhaps by indenting the second and subsequent lines of each title by six spaces. Our modified option statements are now:

R 050
L10 1 06
T20V245

The option statements for the Print program are fixed format. That is, the locations of the various numbers and codes determines their meaning.

Now our report looks like this:

Fertility and attitudes toward family planning in
Algeria, by Johan W. Eliot, et al.

Contemporary China: papers. Edited by Ruth
Adams.

The relation of performance to social background
factors among Army inductees, by Francis
John Ryan.

An Introduction to social research, by Hubert
M. Blalock.

Survey of the Alliance for Progress: problems of
agriculture, by William C. Thiesenhusen and
Marion Brown.

(Figure 1)

If we now wish to have the citations numbered
sequentially and also to print the call number and "did"
number after each title, we will have to add additional
option statements, so that the full set will be:

```
R      050          X
L10    1          06
T20V245
L30    0606
T40VCAL
T50FDID                      V30
```

The Report (R) statement gives options that apply to
the whole report. Here, the "X" in column 17 requests
automatic numbering on each record, or citation. We have
added a second L statement to cause the call number and
"did" number to be placed on a new line, with the same
indentation.

Statement T40 requests printing of the call number, for
which the TIS-CPC alphabetic tag is "cal". Statement T50
prints the "did" number ("F" instead of "V" because the
"did" is not a variable field).

The "V30" in columns 26-28 has the effect of a tab to
column 31. Its precise meaning is "insert a constant, or
caption, of maximum length of 30 blanks, before the "did"
number; reduce the length of this constant for any
characters that have already been placed on the line."

Our latest version of option statements results in this
report:

1. Fertility and attitudes toward family planning in Algeria, by Johan W. Eliot, et al.
AE-1
2. Contemporary China: papers. Edited by Ruth Adams.
DS777.55.C44635 1966a B-125
3. The relation of performance to social background factors among Army inductees, by Francis John Ryan.
UH629.3.R92 B-2546
4. An Introduction to social research, by Hubert M. Blalock.
H62.B576 B-3008
5. Survey of the Alliance for Progress: problems of agriculture, by William C. Thiesenhusen and Marion Brown.
LA-39

(Figure 3)

The first and last records had no call numbers in them, so the maximum length blank constant was inserted before the "did" number. Note that the six character positions taken by the citation number (including period, blank, and space for larger values) match our delayed indent of six.

Now, let us add a page heading, by including page heading option statements:

```
R      050          X
P10    N00
P20X   L  $LIST OF $SMARC $TITLES
P30    D00R
P40    01
L10    1    06
T20V245
L30    0606
T40VCAL
T50FDID          V30
```

The "P" statements print the report title and page heading. Statement P10 prints the page number ("N" in column 6). The carriage spacing value "00" causes the heading data to be printed without any movement of the printer carriage. Since no horizontal print location is specified in columns 9-11, the page number will be centered.

Statement P20 will print the heading "List of MARC Titles," dark printed ("X" in column 4), beginning at the left margin ("L"), on the very next line (the default when no carriage spacing instruction appears).

The '\$'s in the heading text are shift characters. Unless the "asis" code appears in column 6, heading data is translated to lower case except where the letter follows a "\$" or is in a word preceded by "\$\$". This facilitates printing upper-lower case headings using option statements punched into cards.

Statement P30 prints the system date ("D") on the same line as the previous data (carriage spacing instruction "00"), against the right margin ("R" in print location). Statement P40 prints a blank line after skipping one blank line, for a total of two blank lines between the heading and the body of the report.

The report will look like this:

List of MARC Titles -2- 2/23/77

1. Fertility and attitudes toward family planning in Algeria, by Johan W. Eliot, et al. AE-1
2. Contemporary China: papers. Edited by Ruth Adams. B-125
DS777.55.C44635 1966a
3. The relation of performance to social background factors among Army inductees, by Francis John Ryan. B-2546
UH629.3.R92
4. An Introduction to social research, by Hubert M. Blalock. B-3008
H62.B576
5. Survey of the Alliance for Progress: problems of agriculture, by William C. Thiesenhusen and Marion Brown. LA-39

(Figure 4)

Job Control Language

Now that we have prepared our option statements, we need to accompany them with appropriate Job Control Language (JCL) statements to instruct the computer operating system to execute the Print program. Since JCL is subject to variation according to the installation, we will illustrate that used by the TIS-CPC at the N.C. Triangle Universities Computation Center. Suitable modifications to TIS-CPC JCL can be made by referring to the contents of the JCL procedure library and the JCL sections of the chapters for each program.

To simplify the task of submitting a job, the JCL for the output programs of the Bibliographic/MARC Processing System has been organized into a library of JCL procedures. The user supplies the procedure name and the information that is specific to his request, such as the name and location of the master file to be used.

The procedure for executing the Print program is invoked by the following JCL statements:

```
//jobname JOB account,name,etc.
//*PROCLIB=procedure-library
//stepname EXEC PRINT
//QUERY.SYSIN DD *
(option statements)
//PRT.MARCIN DD DSN=marcfile,UNIT=unit,VOL=SER=volume
```

where the lower case words would be replaced with the appropriate parameters.

The procedures for executing the output programs first execute the Option Statement Preprocessor (QUERY) and then

one or more of the selection, sort, and print programs. The SYSIN ddname is used for option statements and the MARCIN ddname is used for the MARC-format input file to be processed.

In the statement "//QUERY.SYSIN", the stepname "QUERY" is not in fact necessary, since JCL DD statements without a stepname are assigned to the first step. So the statement could have been written:

```
//SYSIN DD *
```

The SYSIN statement may instead be written with the name CARDS, since the name CARDS is also known to the procedure (through appearing in a DDNAME parameter). The form "QUERY.CARDS" has been the usage until recently, and is still correct. The newer name, SYSIN, has the advantage of being a standard for various other systems, as well as being the default for the TUCC JCL scanner when non-JCL statements are read in a job stream. This default enables the TUCC user to omit the SYSIN statement entirely, if he desires, and write only:

```
//jobname JOB account,name,etc.
//*PROCLIB=UNC.PC.F282C.MCCLURE.PROCLIB
//stepname EXEC PRINT
(option statements)
//PRT.MARCIN DD DSN=marcfile,UNIT=unit,VOL=SER=volume
```

letting the TUCC JCL scanner generate a SYSIN statement before the option statements.

Option statements for re-ordering

Thus far, we have had recourse only to the Print program. We might prefer to have our list of titles printed in title order, rather than "did" order. For this, we will need the Sort Key Edit program and a sort program to re-order the records before they are processed by the Print program.

The Sort Key Edit program will read option statements to describe the sorting requirements. In this case, one statement will suffice:

```
SKED      VTIL AB N
```

Like the option statements for the Print program, SKED option statements are also fixed format. Their unique identification ("SKED") permits them to be interfiled with the Print option statements without risk of confusion.

"VTIL" refers to the variable field whose alphabetic tag is "til". The "AB" requests that only subfields "a" and "b" of the field are to be used in building the sort key (if the codes were omitted, the entire field would be used). The "N" in column 20 specifies that records that do not contain a "til" field are not desired for further processing. This feature could be used to eliminate the two records with no call number, if desired, by coding the SKED statement:

```
SKED      VTIL AB NVCAL      N
```

After subfields "a" and "b" are placed into the sort key, the SKED program will attempt to follow them with the call number. If it is unable to do so because there was no field with alphabetic tag "cal", the SKED program will cease processing of the record and immediately read the next, without writing out the record it had been processing. This is an example of the record selection capabilities of the

SKED program.

The job deck for executing the SKED program, the IBM Sort/Merge program, and the Print program would be:

```
//jobname JOB account,name,etc.
//*PROCLIB=procedure-library
//stepname EXEC SORTPRT
//QUERY.CARDS DD *
SKED      VTIL AB  NVCAL      N
(Print option statements)
//SKED.MARCIN DD DSN=marcfile,UNIT=unit,VOLUME=volume
```

The resulting report would look like that in Figure 4, except that the records would now appear in alphabetic order (ascending) by title, and the two records which lacked a call number would not be present. Note that in this JCL, the step name on the last statement was changed to correspond to the name of the procedure step that first reads the MARC-format master file. It may also be useful to note that both the Print program and the Sort Key Edit program have the ability to process MARC-format files whose records have had sort keys added to them by the SKED program.

Option statements for selection

The report we have prepared will list all records that contained a call number field. In many cases, we shall want to restrict processing to items that deal with a particular subject or subjects, associated with a particular geographic area, or otherwise identifiable by the information contained in their variable fields. For this type of selection, we will need the facilities of the Free Text Retrieval program (if the subjects or other fields had previously been used to construct an index to the master file, using the file inversion programs, a more efficient search procedure would be the Indexed Retrieval program).

Using the Free Text Retrieval program (BPSRET), let us select for processing only records whose title contains the word "social" or the word root "attitud" (for "attitude", "attitudinal") and which have a call number. Our search statements would be:

```
#SOCIAL FACTORS
1 CONTAIN 'SOCIAL','ATTITUD' (245)
2 HAVE (050)
1 AND 2
*
SKED and Print option statements
END
```

Search statements are free form, except that the Title, Delimiter (*), and END statements must begin in column 1. The statement beginning with a number sign (#) is a (required) search title. Only the "#" is mandatory. The Option Statement Preprocessor will make the title from the search title statement available for printing by the Print program. This facility is described in the chapter "Option Statements and the Option Statement Library".

The statement beginning "1" specifies the criterion "the variable field whose numeric tag is 245 must contain the character string 'SOCIAL' or the character string 'ATTITUD'." Statement 2 specifies the criterion "the record must have a variable field having numeric tag 050" (TIS-CPC tag for call number--the Free Text Retrieval program accepts only numeric tag specifications).

The logic statement that follows joins the search criteria into a single condition: select records that satisfy criteria 1 and 2. The * statement denotes the end of the search statements. The END statement concludes the search and sets the Retrieval program to read another search if there is one.

Since we now need the Retrieval program as well as the SKED, Sort, and Print programs, our job deck will be:

```
//jobname JOB account,name,etc.
//*PROCLIB=procedure-library
//stepname EXEC RETSSP
//QUERY.SYSIN DD *
#search title
1 search criterion #1
2 search criterion #2
(additional search criteria statements)
logic statement
*
(SKED and Print option statements)
END
//RET.MARCIN DD DSN=marcfile,UNIT=unit,VOL=SER=volume
```

Note that the step name in the last statement has again changed to correspond to the step name in the procedure of the first program that reads the MARC-format file.

Categorical and summary reports

We have now illustrated facilities for selecting, sorting, and displaying MARC-format records on a "record by record" basis, that is, one where each record is processed independently from the other, with the same information being displayed from each. Another class of report groups records into categories according to some common information in them. For example, we might like to have a list of bibliographic citations in order by title, grouped within subject categories.

Production of such a report requires, first, that the a record be generated for each subject term in the file; second, that the resulting records be placed in order by the subject term and subordinately by title; and third, that when the records are printed, the subject term be printed only from the first record that contained that term, even though the remaining records in the category also contain that subject term.

The report might look as follows:

Population.

Fertility and attitudes toward family planning in Algeria. AE-1.

Survey of the Alliance for Progress: problems of agriculture. LA-39.

Social factors.

Fertility and attitudes toward family planning in Algeria. AE-1.

An Introduction to social research. B-3008.

The relation of performance to social background factors among Army inductees. B-2546.

(Figure 5)

Record AE-1 contained two subject terms, so it has been posted under both "Population" and "Social factors".

The option statements to produce this report might be:

```
SKED      VSUPE      NTIL AB
SKED      VSUME      NTIL AB
R      050
L10      1 2
T20H
L30      1 0505
T40VTIL   AB
T50FDID
E
```

02

The SKED statements generate one record for each primary subject term (TIS-CPC tag "sup") and for each minor subject term (TIS-CPC tag "sum"). The "&" specifies the generation of postings function. Records will be interfiled and sorted by subject (major) and by subfields "a" and "b" of the title (minor).

The Print option statements define two paragraphs. Statement L10 introduces the category heading, to be preceded by two blank lines. The content of the heading, specified by statement T20, will be the term that was used by the first field of the SKED statement that generated the record.

Statement L30 defines a second paragraph, preceded by one blank line and with a block indentation of five spaces. The paragraph will contain subfields "a" and "b" of the title, followed by two spaces and then the "did" number.

The "1" in column 7 of statement L10 identifies this paragraph as a category or citation group heading that is to be printed only when it changes rather than once for each record. Categories can be defined at as many as four different levels, e.g.:

Subject
 Area
 Year of publication
 Author
 First citation.
 Second citation.
 Etc.

(Figure 6)

In general, each level is represented by the corresponding field on the SKED statement and by an L statement having that level number in column 7. What is actually printed as the group heading is governed by the T statement(s) that follow. The "H" in column 4 of statement T20 causes the data that was used by the SKED statement in sorting the record to be printed in the heading. Note that in this case the use of the "H" code is not equivalent to specifying "VSUP" to request printing of the primary subject heading because:

1. the record being printed may have been generated by the second of the two SKED statements, which specified a minor subject term;
2. each record may contain multiple subject headings--the one that properly belongs as the category heading is the one that was used to sort the particular record being printed, which is precisely what is located in response to code "H".

A two-level version of the above report, primary geographical area by primary subject term, could be produced using these option statements:

```
SKED      VGAP&    HVSUP&    N
R
L10      1 2
T20H
L30      2 1 0202
T40H
L50      1 0404
T60VTIL
E
                                A'AREA NOT STATED.'
```

This report will have two blank lines before each area heading (statement L10), one blank line before each subject heading (statement L30), and one blank line before each title (statement L50). Records that do not contain subject terms will be excluded because of the "N" on the SKED statement after "VSUP".

Records that contain subject terms but do not contain geographical area terms will be assigned an arbitrarily high sort key (because of the "H" following "VGAP" on the SKED statement) so that they come after all legitimate area names on the printed report. These records will appear under the heading "AREA NOT STATED" specified on statement T20 as an alternate ("A" in column 26) caption, which is to print when the field referenced by the T statement is not present.

Hierarchical categories

In both of these examples, the records that were posted into categories could belong to more than one category. Also, any second-level category could be subordinate to any first-level category. Another variation would be where the first- and second-level categories were interdependent, so that the subordinate term could appear with only a specific higher-level term.

The following sample groups records by nationality within region:

AFRICA

ALGERIA

Bouisri, Djamal-Eddine
Population Council-70 1969-71

Mehani, Nafissa
Population Council-70 1969-71

GHANA

Annan, Geoffrey
AID-21100 1969-71

Quartey, Veronica
AID-2242 1972-73

(Figure 1)

The data base printed in this illustration contains manpower training records. Each record has a nationality field, with the country name as subfield "a" and the region name as subfield "b". Since in this case each record has only one such item of information, the region and nationality could occupy different fields without possible ambiguity about which nationality is associated with which region.

Option statements for this listing are:

SKED	VNAT B	HVNAT A	HVNME	H
R 050				
L10 X1E			A'NOT STATED'	
T20H			A'NOT STATED'	
L30 X2E110202			A'NAME MISSING'	
T40H	0505			
L50				
T60H	0505			
L70				
T80VSPN			!-!	
T85VSID			! !	
T90VTYR				
E				

The SKED statement causes records missing either a region (subfield "b" of "nat"), a nationality (subfield "a"), or a name ("VNME") to be sorted high. Such records will be listed as "NOT STATED" or "NAME MISSING". Statements T80, T85, and T90 print the sponsor, sponsor identification number, and training years, with intervening hyphen and blank.

The "X" in column 6 of the first two L statements causes the data to be printed in upper case, regardless of its representation in the record. The "E" in column 8 of the first two L statements causes the generation of

continuation headings. That is, when a category extends beyond the end of the page (or column, if this were a multiple column report), the heading will be repeated at the top of the next column, e.g.:

AFRICA (cont.)

Continuation headings may be specified to any level desired.

If the records contained multiple "nat" fields, then the SKED statement would have been:

```
SKED      VNAT&B   HVNAT=A   HVNME      H
```

The added codes "&" and "=" cause a record to be generated for each occurrence (code "&") of the "nat" field with the region and country taken from the same occurrence (code "=").

Tabulation

Yet another variant of a categorical listing is a summary report that rather than printing data from the individual records, merely tabulates them within categories. The summary report is simply an extension of the categorical report, with no data being printed except at group level.

Africa	5
Algeria	13
Botswana	29
Congo	125
Nigeria	-----
	172
	====
REPORT TOTAL	172

(Figure 4)

This two-way frequency distribution illustrates the tabulation capabilities of the Bibliographic/MARC Processing System. Option statements are:

```
SKED      VNAT B   HVNAT A   H
R 035
L10 1&21                A'NOT STATED'
T20H
L30 2 0202              A'NOT STATED'
T40H
L50 2T
T60N
L70 1T      4   R      R  -----
T75C
T80N
L90      R      4   R      R  =====
T93C
T97NR
E      'REPORT TOTAL'
```

Here, the level-one group is the region (defined by the first field in the SKED statement and the "1" in column 7 of statement L10). The level-two category is country of nationality (subfield "a" of field "nat"). If either is missing, "NOT STATED" will be printed as the group name.

Statements L50, L70, and L90 define group and report trailers. A group trailer, like a group or category heading, is printed only when there is a change in the data

that was used to sort the records. The trailer facility is invoked by the "T" in column 8 (for group trailers) or the "R" in column 8 (for a trailer to be printed at the end of the report).

Statement L50 defines a trailer for the second-level group, which consists of all the records for a given country. Statement L70 defines a trailer for the first-level group, which consists of all the records for a given region (and therefore includes all of the second-level groups for countries in that region).

The tabulation totals themselves are printed by the "N" in column 4 of the T statements within the group trailers. The total referred to is that of the group within which the T statement falls, so statement T60 prints the total for each level-two group (country) and T80 prints the region totals. Statement T97 prints the report total. Statements T75 and T93 print only captions, or constants. Because the group referred to has already ended by the time the trailer prints, it can contain only tabulation totals and fixed captions, not data from the last record of the group.

The group trailers in this report illustrate some additional formatting tools. The "R" in column 17 of statement T60 specifies that the data (in this case, the tabulation total) is to be justified against the right margin. The "R" in column 25 of statements L70 and L90 have the same effect as coding the right-justification option on their subordinate T statements. The "4" in column 15 of these L statements causes a new line to begin for each subordinate T statement.

Index reports

One special type of categorical report is an index to a previously printed report. The index report lists all the occurrences of one or more fields, along with a reference number or other data from each citation in the base report. A subject index to our sample report in Figure 4 might look like this:

Sample index report

3/8/77

Subject index

Agriculture: 2, 5.
 Attitudes: 1, 2, 3, 4.
 Alliance for Progress: 5.
 Family planning: 1, 2, 5.
 Fertility: 1, 5.
 Methodological issues: 2, 4.
 Military affairs: 2, 3, 5.

(Figure 9)

In this index report, each time a new level-one category is read, it is printed along with the reference number of the record. When the category read is the same as that already printed, only the reference number is printed.

The reference numbers are the citation (record) numbers that were generated by the Print program in producing the report in Figure 4. The subject terms printed in the index need not have been printed in the report itself, but must have been present in the records. Record number 4, for

example, must have contained at least two subject terms, "Attitudes" and "Methodological issues", in order to have been posted under those terms in the index report.

An index report is produced by taking the MARC-format records that were processed by the Print program and sorting them into the order required for the index. Generally, additional copies of the records (postings) will be generated, to provide one copy for each of its appearances in the index. The sequence of operations is:

- (1) Selection of records to be included in the report (unless all records are to be included. include).
- (2) Sorting of records, by the SKED and Sort programs (unless the base report is being printed in the master file order).
- (3) Processing of the output from (2) by the Print program to print the base report.
- (4) Processing of the output from (2) by the SKED program to generate postings and to prepare the records for re-sorting into the order of the indexes; sorting the records with the Sort program.
- (5) Processing the output from (4) by the Print program to print the indexes.

The job deck to accomplish these steps would typically involve two JCL procedures:

```
//INDEXES JOB account-code, etc.
//* Typical report with indexes
//REP EXEC RETSSP
//QUERY.CARDS DD *
(option statements for report and indexes)
//RET.MARCIN DD DSN=marcfile,UNIT=unit,VOL=SER=volume
//INDEXES EXEC SORTPRT,FARM.QUERY=INDEX
//QUERY.CARDS DD DSN=*.REP.QUERY.OUTPUT,DISP=(OLD,DELETE)
//SKED.MARCIN DD DSN=*.REP.SORT.SORTCUT,DISP=(OLD,DELETE)
```

The option statements for the indexes can be included along with those for the base reports, by a procedure that will be explained later. RETSSP will accomplish the steps of selection, sorting, and printing of the base report (steps 1-3 above). SORTPRT will carry out the generation of postings, resorting, and printing (steps 4-5 above). Note that the input to the SORTPRT procedure comes from the output of the RETSSP procedure, in this example through the use of referbacks in the dataset name (DSN) parameter.

The option statements for producing the indexes might be the following:

```
SEXT      VSUP&      NREF
SEXT      VSUM&      NREF
R 30360606 1 11
P10      L  $SAMPLE INDEX REPORT
P20      D00R
P30      01  $Subject $index
P40      01
L10      1&    02
T20H
T30FREF      1', '      02
               '.,'
```

The SEXT statement is a special form of the SKED statement. Its function is exactly the same except that the records that are written contain only those variable fields that were used to build the sort key are extracted from the MARC record and included in the output record. Since an index

report prints only the field that was used for sorting; no other fields are usually required. The use of the extraction facility increases the efficiency of the re-sorting process.

The "FREF" specification in the SEXT (SKED) statements causes the physical sequence number of the record to be used as the minor sort field ("S" -- sequence number -- may be written instead of "FREF"). The sequence number will be included in the sort key and also inserted into the fixed-field portion of the output records. It is this sequence number that is printed in the index by the "FREF" (reference number) request on statement T30.

Statement T30 also contains another new feature. The "1" in column 26 indicates that the constants that follow are to be appended after the field data depending upon whether the (level 1) category or citation group has been completed or not. If the group has not ended, the first of the two constants is appended (,); if the group has ended, the second constant (.) is appended. This is the mechanism for achieving the punctuation in the line:

Attitudes: 1, 2, 4.

This constant processing code has the same effect, though its capability is more restricted, as the group trailer facility that was illustrated in the tabulation example in Figure 8.

Ordinarily, each new MARC-format input record is printed beginning on a new line. In this index report, however, the only information printed from most of the MARC input records is the reference number, since we print the category name only when it changes. The "1" in column 20 of the Report statement specifies that this is a summary or index report where only level 1 category groups should automatically begin on a new line.

The "I" in column 19 of the Report statement specifies a special page numbering facility designed for index reports. Since we generally want the page numbers on an index report to continue from the last page number that was used on the report it refers to, the Print program provides a parameter file (the PARMFIL) for passing that information from the execution that prints the base report to the execution that prints the index report. The index report page numbering option causes the starting page number for the first index to begin after the last page of the base report, while additional indexes for that same report continue their page numbers from the previous index report. Index reports are identified with the base report through report number code conventions that are explained below in the section on "Processing multiple reports".

Placement of option statements for index reports

The option statements that are required for the generation of postings, re-sorting, and printing of index reports (steps 4-5 in our list above) may be submitted directly into the procedure that carries out those steps. It is generally more convenient, as in our example of JCL for a typical report with indexes, to have all our option statements in one place, letting the procedure that produces the index reports read the edited preprocessed option statements from the first procedure.

Option statements for index reports when they accompany option statements for base reports are introduced by the statement:

.INDEXES

e.g.:

```
#SSAMPLE REPORT WITH INDEXES
1 CONTAIN 'GHANA' (043)
1
*
SKED          VTIL AB  N
$REPORT
.INDEXES
SEXT          VSUP&    NS
SEXT          VSUM&    NS
$INDEX
END
```

When these statements are read by the output programs, the statements between the .INDEXES and END statements will be ignored, unless the programs are operating in Index mode. In Index mode, most of the statements that precede the .INDEXES statement will be ignored. So the option statements may be entered together as in the above example. The RETSSP step in our JCL example above operates in normal mode, ignoring the index report option statements. The SORTPRT step is set to Index mode, so in reading the output from the first procedure, only the option statements that are appropriate for the index reports are used. The details of this process are explained in the chapter "Option Statements and the Option Statement Library".

Option statement library

The statements "\$REPORT" and "\$INDEX" illustrate calls to the Option Statement Library. In order to increase the convenience and power of using the output programs, frequently used sequences of option statements may be placed in a library of option statements. Statements in the library are invoked through the use of library calls. This use of this facility are discussed in a later chapter.

Processing multiple reports

All of the general purpose output programs have the ability to process multiple reports during a single execution. So several bibliographic searches or other retrievals may be carried out, sorted, and printed simultaneously, yet independently. This facility reduces both computer costs and manual effort involved in multiple jobs.

Each report is assigned a report number code that is inserted in the MARC-format records for that report. A record to be included in two different reports will appear twice, with different report number codes. The sort function will place the records in order by their report number codes, enabling the Print program to print each report separately.

Report number code assignment may be made according to default conventions or controlled by the user. In most situations, the automatic conventions suffice and the user need not be concerned with the details of report number codes. For example, consider the following schematic representation of option statements for two searches with two indexes each:

```

#SSEARCH ONE
(search statements)
*
(SKED and Print option statements
for printing search one)
.INDEXES
(SKED and Print option statements
for printing first index for search one)
(SKED and Print option statements
for printing second index for search one)
END
#SSEARCH TWO
(search statements)
*
(SKED and Print option statements
for printing search two)
.INDEXES
(SKED and Print option statements
for printing first index for search two)
(SKED and Print option statements
for printing second index for search two)
END

```

No report number codes need be specified for this example, since the assignment conventions will associate the indexes of the first search with its records and the indexes of the second search with the records of the second search. The Retrieval program numbers each search consecutively, beginning with number 1. The Sort Key Edit program, in normal mode, assigns the first set of SKED statements (up to the first END statement) to search number 1 and writes the records for this search with report number code 10, reserving codes 11-19 for the indexes to that report. SKED statements after the first END statement are assigned to search number 2 and the records for the base report receive code 20, reserving codes 21-29 for the indexes to the second search. So in our example of two searches each with two indexes, the report number codes will be assigned:

Search 1, base report	10
Search 1, first index	11
Search 1, second index	12
Search 2, base report	20
Search 2, first index	21
Search 2, second index	22

This numbering convention enables the Print program to associate the first index of a particular search with the base report for that search, so that the index report page numbering facility can be accomplished.

In Index mode, the Sort Key Edit program assigns the first set of SKED statements (after the .INDEXES statement and before the first END statement) to base report 10 and writes out the records with report number code 11, 12, 13, etc., according to the number of EIDX (End-Index) statements that have preceded each SKED statement.

In either mode, the SKED program writes out dummy, place-holder records for reports where SKED statements were read but no output records were written (possibly because a search found no records). These dummy records enable the Print program to assign the option statements it reads to the appropriate MAKC records, since the Print program applies each set of option statements to the records with the next report number code found on its input file.

Transaction generation

The output from the Print program consists of a machine-

readable file that is ordinarily routed for printing by the Operating System. But it is not obligatory that the file be printed, and there are a number of situations in which the file produced by the Print program is read by other programs for further processing (e.g., for the production of microfiche via COM--computer output microfiche, for analysis by a statistical package or text processing program).

One such situation is the use of the Print program to generate transaction lines that are then read by the Update programs of the bibliographic/MARC Processing System modify the MARC-format master file. The need for systematic changes of this nature may arise as experience with a new application uncovers difficulties or opportunities that had not been foreseen.

Since the output file produced by the Print program contains fixed length 133-byte records, all that is necessary for transaction generation is to design option statements that produce the required transaction lines. The Update programs provide special options to remove the carriage control character that begins each print line record (PARM=GENTRANS in the Data Edit program) and to suppress or limit the information printed on the Proof Sheet (PARM=NOPRT or LIMPKT in the Update program).

Suppose a variable field "fab" with two subfields, "a" and "b", is to be transformed into separate fields, "fda" and "fdb". The transaction lines for such a change would look like this:

```
did(C      B-123
fda        This was subfield "a".
fdb        This was subfield "b".
fab(D
```

where the "fab" field in record B-123 was originally:

```
fab        This was subfield "a". #This was subfield "b".
```

To generate transactions in this form, we could employ option statements:

```
R 132      CON
L10        10
T20FDID    'did(C      '
T30VFAB    A        'fda        '
T40VFAB    B        'fdb        '
T50CFAB    'fab(D'
E
```

In this example, statement T20 will generate the "did" line for each record and statements T30 through T50 will generate the "fda", "fdb", and "fab" transaction lines, respectively (it should be noted that our use of lower case letters for tags is primarily for appearance, since it is equally acceptable to use upper case tags in transaction lines and it is generally easier to write upper case letters in option statements that must be punched into cards or entered into a TSO dataset).

The "CON" in the Report statement requests continuous printing of data, without the generation of skips to the top of a new page (except at the beginning of the output file). This specification avoids the possible insertion of carriage space lines between the transactions of a particular "did".

The tag "FAB" in statement T50 causes this constant to print only if the record contains an "fab" tag. This specification could have been omitted, since in generating a "did" line for every record without first testing for the

presence of an "fab" field, we have made the tacit assumption that all records contain one (and only one) "fab" field, each containing only subfields "a" and/or "b". It would be possible to modify or add option statements if these assumptions were not correct.

This concludes our overview of basic capabilities of the output programs of the Bibliographic/MARC Processing System. These capabilities are presented in greater detail in the following chapters. Full discussion of the option statements and Job Control Language for each program appears in the chapters for the individual programs.