UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

REFS-DTR

A References Database for VAX DATATRIEVE

By

Carol N. Gerlitz[1]

Open-File Report 87-0637

1987

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Any use of trade names is for descriptive purposes only and does not imply endorsement by the USGS.

[1] Denver, Colorado

CONTENTS

ILLUSTRATIONS

APPENDICES

INTRODUCTION

REFS-DTR is a database management system for bibliographic references that utilizes VAX DATATRIEVE[1] and an ASCII file of references in block format. The system was written with VAX DATATRIEVE version 3.4 and VAX CDD[1] (Common Data Dictionary) version 3.3. A PASCAL program and the Convert utility of VAX RMS[1] (Record Management Services), version 4.5, facilitate conversion of the ASCII file into an indexed DATATRIEVE file.

DATATRIEVE procedures were written to search the database, modify references already in the database, and store new references. The procedures also retrieve information from REFS-DTR for output to the terminal screen and/or printer files at the user's request. Output files can range in size from one reference to a complete listing of the bibliography alphabetized by author.

REFS-DTR was originally written for the Delta, Utah, CUSMAP (Conterminous United States Mineral Assessment Program) project. This project maintained an ASCII file of over 900 bibliographic entries on a VAX computer; users of the file wanted to put it into a database system where keywords could be added for each reference. The REFS-DTR system written for the Delta project can be easily adapted to accommodate other users who have an ASCII file of references in block format.

This report publicly documents the REFS-DTR database management system used by project personnel of the Delta CUSMAP project; it is also meant to serve as a "hands-on" guide for use by anyone interested in creating and maintaining a bibliographic reference database.

GETTING STARTED WITH REFS-DTR

REFS-DTR Data File

An ASCII file that is to be converted into an indexed DATATRIEVE file must contain references in block format, left-justified, with no more than 80 characters per line. The ASCII file the Delta project started with contained a retrieval of references from GeoRef, the online equivalent of the Bibliography and Index of Geology published by the American Geological Institute. The file was received by modem, copied onto a floppy disk, and uploaded from a PC to the VAX by means of a file transfer program. Before the file was entered into the REFS-DTR database, the format of the references was substantially modified to agree with U.S. Geological Survey editorial style; all non-Delta references were weeded out, and new references were added to the ASCII file. Figure 1 is an example of the ASCII file format after editing, prior to conversion to a DATATRIEVE file.

REFS-DTR Dictionary Objects

To set up REFS-DTR, copy two DATATRIEVE domain (ORIG_REFS and REFS) and two DATATRIEVE record definitions (ORIG_REFS_RECORD and REFS_RECORD) into the common data dictionary (Appendices A through C). Modify domain definitions to

[1] VAX, DATATRIEVE, CDD, and RMS are trademarks of Digital Equipment Corporation. DATATRIEVE is an interactive language and report-writing tool for retrieving data from existing VAX RMS files; it can be used as a relational database management system.

Abbott, W.O. 1951, Cambrian diabase flow in central Utah: Compass, v. 29, no. 1, p. 51-10.

Afrasiabi, H., 1981, A study of mineralization in the Dugway Range, Tooele County, Utah: Salt Lake City, Utah, University of Utah, M.S. thesis, 75 p.

Ahlborn, R.C., 1977, Mesozoic-Cenozoic structural development of the Kern Mountains, eastern Nevada-western Utah: Brigham Young University Geology Studies, no. 24, pt. 2, p. 117-131.

Alexander, R.R., 1978, Growth, morphologic variability, and asymmetry among Chesterian (Upper Mississippian) brachiopods from the Confusion Range, Utah: Geological Society of America, Abstracts with Programs, v. 10, no. 5, p. 209.

Alexander, R.R., 1981, Predation scars preserved in Chesterian brachiopods; probable culprits and evolutionary consequences for the articulates: Journal of Paleontology, v. 55, no. 1, p. 192-203.

Alling, A.N., 1887, On the topaz from the Thomas Range, Utah: American Journal of Science, ser. 3, v. 33, p. 146-147.

Allmendinger, R.W., Hauge, T.A., Hauser, E.C., Potter, C.J., Klemperer, Nelson, K.D., Knuepfer, P., and Oliver, J., 1987, Overview of the COCORP 40 degrees N Transect, western United States: The fabric of an orogenic belt: Geological Society of America Bulletin, v. 98, 308-319.

Figure 1. Format of the ASCII references file prior to reformatting by Program REFS2DTR.

specify the correct complete path names to the dictionary in which your record definitions will reside and to the VMS[1] directory in which your data files will be located (see Appendix A). If necessary, also modify field lengths for the fields REFNUM, AUTHOR, and CITATION2 in the record definitions (see Program REFS2DTR section below).

The initial domain and record definitions, ORIG_REFS (Appendix A) and ORIG_REFS_RECORD (Appendix B), fit the format of the original ASCII data file when it is converted to an indexed DATATRIEVE file (see Procedure for Creating a DATATRIEVE File below). In ORIG_REFS_RECORD, the field YEAR is defined as a four-digit number that is only "VALID IF" it contains either "18" or "19". The value of YEAR is therefore restricted to "18xx" or "19xx", and year designations such as "1987a" are not allowed. Note that in the record definition the citation is broken into two fields, CITATION1 and CITATION2. This is because DATATRIEVE allows entry of a maximum of 255 characters into a variable during addition or modification of data. CITATION1 is therefore 255 characters long, and CITATION 2 contains the rest of the citation (245 characters).

Use the domain definition REFS (Appendix A) and record definition REFS_RECORD (Appendix C) to restructure the database file (see Procedure for Creating a DATATRIEVE File). You can add new fields by modifying REFS_RECORD to include whatever you need space for. In the example here, REFS adds a KEYWORD field to the database. If you modify REFS_RECORD to add any other fields to the database, also modify the DATATRIEVE procedures (see Data Retrieval section below) to include the new fields in the output. The field YEAR is also restricted to "18xx" or "19xx" in REFS_RECORD, and the citation field is divided into CITATION1 and CITATION2.

Program REFS2DTR

Program REFS2DTR (Appendix D) reads the ASCII file of block-formatted references, rewrites each reference into a single line, and then reads each single-line reference and breaks it into three fields: author, year, and citation. "Author" includes the entire list of author names, and "citation" includes all information that follows "year" in the bibliographic entry. The program also generates a reference number that is added to the data file when it is written in its final tabular format for DATATRIEVE.

You must modify the PASCAL code, DATATRIEVE record definitions, and file definition language (FDL) file (Appendix E) if your initial file of references contains more than 1000 references or if any of your references have an author string longer than 140 characters or a citation string longer than 500 characters:

1) If you have more than 1000 references, modify the PASCAL code to change the size of the constant "maxrefs".
2) "Maxrefs" less than 10,000 implies a four-digit number. If you have more than 9999 but less than 100,000 references, in addition to changing "maxrefs", make the following changes:
   --in the PASCAL code, modify all "write" statements in Procedure WriteFile to include another leading zero for a total of five digits, modify the last "if" statement to read
   "if ((recnum > 999) and (recnum < 10000)) then", and add another statement, "if recnum > 9999 then write (fileout, recnum:5, author);".

[1] VMS is a trademark of Digital Equipment Corporation.

--in the PASCAL code, in order to accommodate the additional digit in the variable "recnum", increase the size of the variable "line" in Procedure ReadData from "varying [648] of char", and change the record_length in "open (tempfile..." and "open (fileout..." in the main program.

--in the DATATRIEVE record definitions ORIG_REFS_RECORD and REFS_RECORD, increase the field length for REFNUM to five (use "PIC 9(5)").

--in the FDL file, change "Key 0 length" to 5, and increase "Mean record size" and "Maximum record size" by one.

3) If you have a string of author names longer than 140 characters, modify the following items:

--in the PASCAL code given in Appendix D, the variable "author" is data type "varying [140] of char"; change "140" to the required length for the author string.

--in the PASCAL code in Appendix D, the variable "line" in Procedure ReadData is data type "varying [648] of char"; change this line length to accommodate the new author string length.

--in the PASCAL code, in Procedure ReadData under "pick out the author", change the parameter "140" in two loops to reflect the new string size: in the statements "for j := i-3 to 140" and "for j := i-2 to 140".

--in the PASCAL code, change the record_length in "open (tempfile..." and "open (fileout..." to accommodate the new author string length.

--in the DATATRIEVE record definitions ORIG_REFS_RECORD and REFS_RECORD, increase the field length for AUTHOR to the new author string length.

--in the FDL file, increase "Mean record size" and "Maximum record size" to accommodate the new author string length.

4) If you have a citation string longer than 500 characters, modify the following items:

--in the PASCAL code given in Appendix D, the variable "citation" is data type "varying [500] of char"; change "500" to the required length for the citation string.

--in the PASCAL code in Appendix D, the variable "line" in Procedure ReadData is data type "varying [648] of char"; change this line length to accommodate the new citation string length.

--in the PASCAL code, in Procedure ReadData under "pick out the citation", change the parameter "500" in two loops to reflect the new string size: in two statements "for j := (length (line) + 1) to 500".

--in the PASCAL code, change the record_length in "open (tempfile..." and "open (fileout..." to accommodate the new citation string length.

--in the DATATRIEVE record definitions ORIG_REFS_RECORD and REFS_RECORD, increase the field length for CITATION2 to accommodate the new citation string length. Note that in these record definitions the citation is broken into two fields, CITATION1 and CITATION2. If your citation string length is greater than 510, increase CITATION2 to 255 characters; add CITATION3 to accommodate the rest of the citation string, and modify all procedures to include CITATION3.

--in the FDL file, increase "Mean record size" and "Maximum record size" to accommodate the new citation string length.

PROCEDURE FOR CREATING A DATATRIEVE FILE

After domain and record definitions have been installed in the common data dictionary, reformat the original ASCII references file (by using those definitions) into single-line records, convert it to an indexed file, and restructure the file:

1) Load the ASCII references file onto the VAX. Check the file carefully and edit if necessary:
   --eliminate unusual year designations, such as "1987a"-- the format must be "18xx" or "19xx" to fit the database.
   --references with no year will become empty records in the DATATRIEVE file, so add "1999" to references that have no year; this will provide you with a flag to find references that need further checking.
   --be sure there are commas before and after the year.
2) Run Program REFS2DTR (Appendix D) to reformat the references into tabular format. Call the output file REFS.DTR. Program REFS2DTR generates a reference number that is added to the data file when it is rewritten.
3) Use the editor to create a file definition language (FDL) file, REFS.FDL (Appendix E). If you prefer, the FDL file can be created using the RMS Edit/FDL utility, which will ensure the correct format and syntax. To use this utility, type "edit/fdl REFS.FDL" at the system level ($ prompt). Select "invoke" from the FDL editor utility's main menu, and choose "indexed" from the script title selection. In response to prompts, specify the following:

In response to the prompt	Enter
Number of records initially loaded:	0
Number of records to be added:	1
Record format:	variable
Mean record size:	648[1]
Maximum record size:	648[1]
Key 0 length:	4[1]
Key 0 position:	0
Which File Parameter:	FD
Text for FDL Title Section:	FDL TO CONVERT REFERENCES TO DATATRIEVE FILE
Key 0 name:	REFNUM

   Accept default values on all other items you are prompted for.
4) At the system level, use the RMS Convert utility to convert the .DTR-type file to an indexed file:

```
$ convert/fdl=REFS.FDL REFS.DTR ORIG_REFS.DAT
```

   Records in the newly-created "converted" file, ORIG_REFS.DAT, will be ordered by REFNUM.
5) In DATATRIEVE, restructure ORIG_REFS to REFS:

```
DTR> READY ORIG_REFS AS OLD READ
DTR> SHOW FIELDS FOR OLD
DTR> DEFINE FILE FOR REFS KEY=REFNUM (NO CHANGE, NO DUP);
DTR> READY REFS AS NEW WRITE
DTR> SHOW FIELDS FOR NEW
DTR> NEW = OLD
DTR> FINISH
```

[1] Remember to change mean/maximum record size and key 0 length if necessary (see Program REFS2DTR section above).

Restructuring provides the final format for the REFS-DTR data file. You can add new fields during restructuring by modifying REFS_RECORD to include whatever you need space for. In the examples here, the record definitions for ORIG_REFS and REFS differ by only one field--REFS adds a KEYWORD field to the database. Another field that might be useful is COMMENTS.

After restructuring you will have five RMS files: the original ASCII references data file, REFS.DTR, REFS.FDL, ORIG_REFS.DAT, and REFS.DAT. REFS.DAT is the final version of the data that will be used by DATATRIEVE as the database data file. REFS.DTR, REFS.FDL, and ORIG_REFS.DAT may be deleted after you have checked to be sure that the database system is functioning properly.

DATA RETRIEVAL

Data can be retrieved from REFS-DTR in many ways. You can use DATATRIEVE procedures to retrieve and format references (Appendices F through K). To do so, copy the procedures into the common data dictionary, inserting the correct path names to access the domain definition REFS and procedure LISTREFS. In addition, you can interactively retrieve references, using any desired selection criteria, and print all or part of the data in any required format. Two procedures designed to print data to the terminal screen and to a printer file are LIST_REFS (Appendix J) and FILE_REFS (Appendix K), respectively.

REPORT_REFS

Procedure REPORT_REFS (Appendix F) creates a listing of the complete up-to-date bibliography arranged alphabetically by author. REFNUM and KEYWORD fields are not included in this listing.

REFERENCES

Procedure REFERENCES (Appendix G) provides read/write access to the database. The procedure allows you to search through the references using any data field, to modify existing references, and to add new references. You will be prompted by the program for all the necessary information.

SEARCH_REFS

Procedure SEARCH_REFS (Appendix H) allows read-only access to the database. The procedure will prompt you for the necessary information to search through the references using any data field.

LISTREFS

Procedure LISTREFS (Appendix I) is called by Procedures REFERENCES and SEARCH_REFS to format output to the terminal screen or to a printer file. The fields REFNUM and KEYWORD are included in this listing.

LIST_REFS

Use Procedure LIST_REFS (Appendix J) to interactively retrieve references from REFS-DTR by any selection criteria. At the DTR> prompt enter "READY REFS", specify the references needed by using the FOR command, and enter

"EXECUTE LIST_REFS". The procedure lists the retrieved references on the terminal screen, including REFNUM and KEYWORD in the output.

FILE_REFS

Use Procedure FILE_REFS (Appendix K) to interactively retrieve references from REFS-DTR by any selection criteria. At the DTR> prompt enter "READY REFS", specify the references needed by using the FIND command, and enter "EXECUTE FILE_REFS". The procedure will prompt you for a file name and create a printer file in which the output includes REFNUM and KEYWORD.

APPENDIX A. DOMAIN DEFINITIONS

The two domain definitions listed below must be installed in the common data dictionary to set up REFS-DTR. Pathnames to record definitions and data files in the definitions shown here are for the system set up for Delta CUSMAP. Modify the pathnames to reflect the correct DATATRIEVE dictionary and VMS file in which your record definitions and data files will be located.

For example, the complete pathname to the record definition is "CDD$TOP.DTR$USERS.dictionary.record-definition", where "dictionary" is the name of the DATATRIEVE dictionary (or list of dictionary and subdictionary names, separated by periods) in which the record definition is located, and "record-definition" is ORIG_REFS_RECORD or REFS_RECORD. In the domain definitions shown here, the "dictionary" is user dictionary name "SYSCNG", subdictionary name "DELTA", and sub-subdictionary name "REFS".

The complete file specification is "device:[directory]filename", where "device" is the name of the disk and "directory" is the name of the directory (or list of directory and subdirectory names) in which the file, ORIG_REFS.DAT or REFS.DAT, is located. In the domain definitions below "SYS$USER" is the disk name, and "SYSCNG.DELTA.REFS" are the names of directory and subdirectories where the data files are located.

Domain Definition for ORIG_REFS:

```
DOMAIN ORIG_REFS USING
       CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.ORIG_REFS_RECORD ON
       SYS$USER:[SYSCNG.DELTA.REFS]ORIG_REFS.DAT;
```

Domain Definition for REFS:

```
DOMAIN REFS USING
       CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.REFS_RECORD ON
       SYS$USER:[SYSCNG.DELTA.REFS]REFS.DAT;
```

APPENDIX B. RECORD DEFINITION: ORIG_REFS_RECORD

The record definition listed below for ORIG_REFS_RECORD includes a "VALID IF" statement for the information field YEAR, which restricts the value of YEAR to "18xx" or "19xx". A year must therefore be entered for every bibliographic entry (YEAR cannot be left blank). If you enter "1999" for references that have no year, YEAR will provide a flag to find references that need further checking.

If your original ASCII file contains more than 9999 references, or if you increase the length of the author string or the citation string, you must modify this record definition accordingly (see Program REFS2DTR section).

```
RECORD ORIG_REFS_RECORD
01     REF_REC.
       05      REFNUM          PIC 9(4)
                               EDIT_STRING Z(4).
       05      AUTHOR          PIC X(140).
       05      YEAR            PIC 9(4)
                               VALID IF YEAR CONT '19' OR YEAR CONT '18'.
       05      CITATION1       PIC X(255).
       05      CITATION2       PIC X(245).
;
```

APPENDIX C. RECORD DEFINITION: REFS_RECORD

The record definition listed below for REFS_RECORD includes a "VALID IF" statement for the information field YEAR, which restricts the value of YEAR to "18xx" or "19xx". A year must therefore be entered for every bibliographic entry (YEAR cannot be left blank). If you enter "1999" for references that have no year, YEAR will provide a flag to find references that need further checking.

If your original ASCII file contains more than 9999 references, or if you increase the length of the author string or the citation string, you must modify this record definition accordingly (see Program REFS2DTR section).

```
RECORD REFS_RECORD USING
01      REF_REC.
        05       REFNUM          PIC 9(4)
                                 EDIT_STRING Z(4).
        05       AUTHOR          PIC X(140).
        05       YEAR            PIC 9(4)
                                 VALID IF YEAR CONT ´19´ OR YEAR CONT ´18´.
        05       CITATION1       PIC X(255).
        05       CITATION2       PIC X(245).
        05       KEYWORD         PIC X(200).
;
```

APPENDIX D. PASCAL CODE FOR PROGRAM REFS2DTR

You must modify this PASCAL code if your initial ASCII file contains more than 1000 references or if any of your references have an author string longer than 140 characters or a citation string longer than 500 characters (see Program REFS2DTR section for details).

```
PROGRAM REFS2DTR (input, output, filein, tempfile, fileout);
(***************************************************************************
*  Written in VAX-11 Pascal, version 3.4, by Carol N. Gerlitz, April 1987:
*  this program reads references from an ascii file and outputs a text
*  data file that can then be converted into a DTR file using RMS/CONVERT
***************************************************************************)

(* declare the external concatenation procedure *)
PROCEDURE STR$CONCAT (%DESCR DSTSTR: varying [a] of CHAR;
                     SRC1STR: varying [b] of CHAR;
                     SRC2STR: varying [c] of CHAR); EXTERN;

const
  maxrefs = 1000;         (* maximum number of references that can be
                                                  processed by this program *)

type
  refrecord = record (* one reference of a bibliographic entry in text file *)
        recnum   : integer;   (* record or reference number *)
       author   : varying [140] of char;
       year     : varying [4] of char;
       citation : varying [500] of char;
    end; (* refrecord *)
  refs = array [1..maxrefs] of refrecord;

var
  filein    : text;       (* input data file *)
  fileout   : text;       (* final output data file *)
  infile    : varying [14] of char;
                          (* user-input data file name assigned to filein *)
  number    : integer;    (* counts number of references in data file *)
  outfile   : varying [14] of char;
                          (* user-input data file name assigned to fileout *)
  reference : refs;       (* info on references written to DATATRIEVE file *)
  tempfile  : text;       (* temporary data file *)

PROCEDURE INITIALIZE (var reference : refs);
(***************************************************************************
* Initializes the array of records with reference data, temporary ascii file
***************************************************************************)

var  i : integer;  (* loop control variable and array subscript *)

begin (* initialize *)
  for i := 1 to maxrefs do
    with reference[i] do
```

```
      begin (* with *)
        recnum := i;
       author := '';
       year := '';
       citation := '';
      end; (* with *)
end; (* initialize *)

PROCEDURE LINEDATA (var filein, tempfile : text);
(***************************************************************************
* Reads lines from original ascii file and writes each ref to a single line
* in a temporary file
***************************************************************************)

var i,j      : integer;              (* faithful index friend *)
    line     : varying [80] of char; (* reads one line of file at a time *)

begin (* linedata *)
  while not (eof (filein)) do
    begin
      readln (filein, line);
      if line = '' then (* start new line for new record *)
        writeln (tempfile)
      else
        begin
          if line[length(line)] <> ' ' then
            (* if no space at end of line, add one *)
            write (tempfile, line, ' ')
          else
            write (tempfile, line);
        end (* else *)
    end; (* while not eof *)
end; (* linedata *)

PROCEDURE READDATA (var reference : refs; var number : integer;
                                                 var tempfile : text);
(***************************************************************************
* Reads references from temporary file
***************************************************************************)

var yearfound     : boolean; (* flag: end of author, start of year found *)
    begincitation : boolean; (* flag: end of year, start of citation found *)
    i,j            : integer; (* faithful index friend *)
    line           : varying [648] of char;
                     (* used to read author _citation 1 character at a time *)

begin (* readdata *)

  (* initialize reference counter *)
  number := 0;

  while (not eof (tempfile)) and (number < maxrefs) do
    begin (* while *)
```

```
(* increment record counter *)
number := number + 1;

(* read in one record of data at a time *)

with reference[number] do
  begin (* with *)

    (* initialize flags *)
    yearfound := false;
    begincitation := false;

    (* skip blank lines *)
    repeat
      readln (tempfile, line);
    until line <> '';

    (* read the reference and break it into parts *)
    for i := 1 to length (line) do
      begin (* for *)
        if i >= 7 then
          begin (* if *)

           (* pick out the author *)
            if ((line[i] = '9') or (line[i] = '8')) and
                (line[i-1]='1') then
              while not yearfound do
                begin (* while *)

                  (* find author if date _author separated by ', ' *)
                  if (line[i-2] = ' ') and (line[i-3] = ',') then
                    begin (* if *)
                      for j := 1 to i-4 do
                        str$concat(author, author, line[j]);
                      for j :=  i-3 to 140 do
                        str$concat(author, author, ' ');
                    end; (* if line *)

                  (* find author if date _author separated only by ' ' *)

                  if (line[i-2] = ' ') and (line[i-3] <> ',') then
                    begin (* if *)
                      for j := 1 to i-3 do
                        str$concat(author, author, line[j]);
                      for j :=  i-2 to 140 do
                        str$concat(author, author, ' ');
                    end; (* if line *)

                  yearfound := true;
                end; (* while not yearfound *)
```

```
                  (* pick out the year *)
                  if ((line[i-2] = '9') or (line[i-2] = '8')) and
                      (line[i-3] = '1') then
                    for j := i-3 to i do
                      str$concat(year, year, line[j]);

                  (* pick out the citation *)
                  if ((line[i-5] = '9') or (line[i-5] = '8')) and
                      (line[i-6] = '1') then
                    while not begincitation do
                      begin (* while *)

                        (* find citation if citation _year separated by ', ' *)

                        if (line[i-2] = ',') and (line[i-1] = ' ') then
                          begin (* if *)
                            for j := i to length (line) do
                              str$concat(citation, citation, line[j]);
                            for j :=  (length (line) + 1) to 500 do
                              str$concat(citation, citation, ' ');
                          end; (* if line *)

                        (* find citation if citation _year separated by ' ' *)
                        if (line[i-2] = ' ') then
                          begin (* if *)
                            for j := i-1 to length (line) do
                              str$concat(citation, citation, line[j]);
                            for j :=  (length (line) + 1) to 500 do
                              str$concat(citation, citation, ' ');
                          end; (* if line *)

                        begincitation := true;
                      end; (* while not begincitation *)

                end; (* if i > 7 *)
            end; (* for  i := 1 to length (line) *)
        end; (* with reference[number] *)
    end; (* while not eof *)
  writeln;

  if number = 0 then
    writeln (' DATA FILE WAS NOT READ...NUMBER IS STILL ', number:1);

  (* print warning if unread data remains in file *)
  if not eof (tempfile) then
    begin (* if *)
      writeln ('==> Maximum number (', maxrefs:1, ') of refs exceeded;',
                                               ' remaining data ignored.');
      writeln ('  Last ref read is ', reference[maxrefs].author,
                                                reference[maxrefs].year);
    end; (* if *)

end; (* readdata *)
```

```
PROCEDURE WRITEFILE (var reference : refs; var number : integer;
                                                     var fileout : text);
(***************************************************************************
* Writes out data to text file for DATATRIEVE.
***************************************************************************)

var  i : integer;  (* faithful index friend *)

begin (* writefile *)

  (* write one record of data at a time *)
  for i := 1 to number do
    with reference[i] do
      begin (* with *)
        if recnum < 10 then
          write (fileout, ´000´, recnum:1, author);
        if ((recnum > 9) and (recnum < 100)) then
          write (fileout, ´00´, recnum:2, author);
        if ((recnum > 99) and (recnum < 1000)) then
          write (fileout, ´0´, recnum:3, author);
        if recnum > 999 then
          write (fileout, recnum:4, author);
        write (fileout, year:4);
        writeln (fileout, citation);
      end; (* with *)

end; (* writefile *)

(**************************************************************************)

Begin  (* refs2dtr *)

  (* print information message *)
  writeln;
  writeln;
  writeln (´WELCOME TO Program Refs2dtr--´);
  writeln (´  this program will read bibliographic text from an ascii file´);
  writeln (´  and create a text file from it for use in DATATRIEVE.´);
  writeln;

  (* prompt user for and read in input and output file names *)
  writeln (´ENTER NAME OF INPUT DATA FILE (an ascii file)´);
  write (´   Input file?  ´);
  readln (infile);
  writeln (´ENTER NAME OF OUTPUT DATA FILE (text file for DATATRIEVE)´);
  write (´   Output file?  ´);
  readln (outfile);

  (* open input and temporary output data files *)
  open (filein, infile, history := old);
  reset (filein);
  open (tempfile, ´templ.dat´, history := new, recordlength := 648);
  rewrite (tempfile);
```

```
 Initialize (reference);
 LineData (filein,tempfile);

 (* reset temporary file *)
 reset (tempfile);

 ReadData (reference, number, tempfile);

 (* open output data file *)
 open (fileout, outfile, history := new, recordlength := 648);
 rewrite (fileout);

 WriteFile (reference, number, fileout);

 (* close data files *)
 close (filein);
 close (tempfile, disposition := delete);
 close (fileout);

End.  (* refs2dtr *)
```

APPENDIX E. FDL FILE FOR CONVERTING ORIGINAL DATA TO AN INDEXED FILE

If your original ASCII file contains more than 9999 references, or if you increase the length of the author string or the citation string, you must modify SIZE and KEY 0 LENGTH in this FDL file (see Program REFS2DTR section for details).

```
TITLE  "FDL TO CONVERT REFERENCES TO DATATRIEVE FILE"

IDENT  " 1-APR-1987 08:30:33   VAX-11 FDL Editor"

SYSTEM
       SOURCE                  VAX/VMS

FILE
       ORGANIZATION            indexed

RECORD
       CARRIAGE_CONTROL        carriage_return
       FORMAT                  variable
       SIZE                    648

AREA 0
       ALLOCATION              9
       BEST_TRY_CONTIGUOUS     yes
       BUCKET_SIZE             3
       EXTENSION               3

AREA 1
       ALLOCATION              6
       BEST_TRY_CONTIGUOUS     yes
       BUCKET_SIZE             3
       EXTENSION               3

KEY 0
       CHANGES                 no
       DATA_AREA               0
       DATA_FILL               100
       DATA_KEY_COMPRESSION    yes
       DATA_RECORD_COMPRESSION yes
       DUPLICATES              no
       INDEX_AREA              1
       INDEX_COMPRESSION       yes
       INDEX_FILL              100
       LEVEL1_INDEX_AREA       1
       NAME                    "REFNUM"
       PROLOG                  3
       SEG0_LENGTH             4
       SEG0_POSITION           0
       TYPE                    string
```

APPENDIX F. PROCEDURE REPORT_REFS

Modify the READY command in Procedure REPORT_REFS, below, to include the complete pathname to the DATATRIEVE domain REFS. The complete pathname is "CDD$TOP.DTR$USERS.dictionary.REFS", where "dictionary" is the name of the DATATRIEVE dictionary (or list of dictionary and subdictionary names, separated by periods) in which the domain is located. In the example here, the "dictionary" is user-dictionary name "SYSCNG", subdictionary name "DELTA", and sub-subdictionary name "REFS".

```
PROCEDURE REPORT_REFS
!Written by Carol N. Gerlitz, April 1987
  DECLARE CALENDAR_DATE          USAGE IS DATE
                                 DEFAULT "TODAY"
                                 EDIT_STRING IS DDBMMMBYYYY.
  READY CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.REFS
  FN$WIDTH(80)
  PRINT SKIP 3,
    'This program will prepare a report which lists a complete', SKIP,
    'up-to-date bibliography alphabetically by author', SKIP 3,
    'Please be patient...this will take awhile.....', SKIP 5
  FIND REFS WITH REFNUM GT 0 AND (YEAR NE '19' OR YEAR NE '18') SORTED
       BY AUTHOR, YEAR, CITATION1
  ON REFERENCE.LIST
    REPORT
    AT TOP OF REPORT PRINT SKIP 15,
        COL 20, "* * * * * * * * * * * * * * * * * * *", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "*", COL 33, "BIBLIOGRAPHY", COL 56, "*", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "*", COL 33, CALENDAR_DATE(-), COL 56, "*", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "*", COL 56, "*", SKIP,
        COL 20, "* * * * * * * * * * * * * * * * * * *", SKIP,
        NEW_PAGE
    SET REPORT_NAME = 'BIBLIOGRAPHY'
    PRINT AUTHOR||', '|YEAR||', '|CITATION1|CITATION2 USING T(80), SKIP
    END_REPORT
  PRINT '==> Your bibliography is ready to print; the file is called',
    'REFERENCE.LIST.', SKIP 6
  FINISH REFS
END-PROCEDURE
```

APPENDIX G. PROCEDURE REFERENCES

In Procedure REFERENCES, below, modify the READY command to include the complete pathname to the DATATRIEVE domain REFS; modify the EXECUTE command to include the complete pathname to the DATATRIEVE procedure LISTREFS. The complete pathname to these DATATRIEVE dictionary objects is "CDD$TOP.DTR$USERS.dictionary.object", where "dictionary" is the name of the DATATRIEVE dictionary (or list of dictionary and subdictionary names, separated by periods) in which the domain or procedure to be executed is located, and "object" is the name of the domain or procedure. In the example here, the "dictionary" is user-dictionary name "SYSCNG", subdictionary name "DELTA", and sub-subdictionary name "REFS"; "object" is REFS or LISTREFS.

```
PROCEDURE REFERENCES
!Procedure to search, modify, and/or add new references to the bibliography
!Written by Carol N. Gerlitz, April 1987
DECLARE ANSWER         PIC X
                       VALID IF ANSWER CONT 'Y' OR ANSWER CONT 'N'.
DECLARE CONTINUE       PIC X
                       DEFAULT 'Y'
                       VALID IF CONTINUE CONT 'Y' OR CONTINUE CONT 'N'.
DECLARE CONTINUEMODIFY PIC X
                       VALID IF CONTINUEMODIFY CONT 'Y' OR
                               CONTINUEMODIFY CONT 'N'.
DECLARE CONTINUESEARCH PIC X
                       VALID IF CONTINUESEARCH CONT 'Y' OR
                               CONTINUESEARCH CONT 'N'.
DECLARE FILEIT         PIC X
                       VALID IF FILEIT CONT 'Y' OR FILEIT CONT 'N'.
DECLARE MAINCHOICE     PIC X
                       VALID IF MAINCHOICE EQ '1' OR MAINCHOICE EQ '2' OR
                               MAINCHOICE EQ '3' OR MAINCHOICE EQ '4'.
DECLARE OUTPUT         PIC X
                       VALID IF OUTPUT CONT 'S' OR OUTPUT CONT 'F' OR
                       OUTPUT CONT 'B'.
DECLARE PAUSE          PIC X.
DECLARE SEARCHCHOICE   PIC X
                       VALID IF SEARCHCHOICE EQ '1' OR SEARCHCHOICE EQ '2' OR
                               SEARCHCHOICE EQ '3' OR SEARCHCHOICE EQ '4' OR
                               SEARCHCHOICE EQ '5' OR SEARCHCHOICE EQ '6'.
READY CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.REFS SHARED WRITE
FN$WIDTH(80)    !clear the screen
PRINT SKIP 2,
  '             ** WELCOME TO THE REFERENCE DATA BASE **', SKIP,
  '             ----------------------------------------'
WHILE CONTINUE CONT 'Y'
  BEGIN
    PRINT SKIP 3, 'DO YOU WANT TO', SKIP, '   1. Search through the references',
      SKIP, '   2. Modify reference(s)', SKIP, '   3. Add new reference(s)',
      SKIP, '   4. EXIT THIS PROGRAM', SKIP
    MAINCHOICE = *.'number of your choice'
    IF MAINCHOICE = 1 THEN
      BEGIN !search through the references
```

```
CONTINUESEARCH = 'Y'
 WHILE CONTINUESEARCH CONT 'Y'
  BEGIN
   FN$WIDTH(80)
    PRINT SKIP 2,'WHAT DO YOU WANT TO SEARCH ON?', SKIP,
      '   1. Author', SKIP, '   2. Year', SKIP, '   3. Citation', SKIP,
      '   4. Keyword', SKIP, '   5. Reference number', SKIP 2,
      '   6. NO MORE SEARCHES', SKIP
    SEARCHCHOICE = *.'number of your choice'
    IF SEARCHCHOICE = 1 THEN
        BEGIN !author search
         PRINT SKIP
         OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
         IF OUTPUT CONT 'S' THEN !output to screen
            BEGIN !screen output
             ON TT
               FOR REFS WITH AUTHOR CONT *.'author' SORTED
                 BY AUTHOR, YEAR, CITATION1
                    EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
             PRINT SKIP
              FILEIT = *."Y to print this list to a file, N to continue"
               IF FILEIT CONT 'y' THEN OUTPUT = 'f'
            END !screen output
         IF OUTPUT CONT 'F' THEN !output to file
           ON *.'file name'
             FOR REFS WITH AUTHOR CONT *.'author' SORTED
               BY AUTHOR, YEAR, CITATION1
                  EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         IF OUTPUT CONT 'B' THEN !output both to screen and file
           BEGIN !screen _file output
             ON TT ON *.'file name'
               FOR REFS WITH AUTHOR CONT *.'author' SORTED
                       BY AUTHOR, YEAR, CITATION1
                  EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
             PRINT ' '
             PAUSE = *.'any key, then <carriage return> to continue'
           END !screen _file output
         PRINT SKIP
        END !author search
    IF SEARCHCHOICE = 2 THEN
        BEGIN !year search
         PRINT SKIP
         OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
         IF OUTPUT CONT 'S' THEN !output to screen
            BEGIN !screen output
             ON TT
               FOR REFS WITH YEAR CONT *.'year' SORTED BY
                 AUTHOR, YEAR, CITATION1
                    EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
             PRINT SKIP
              FILEIT = *."Y to print this list to a file, N to continue"
               IF FILEIT CONT 'y' THEN OUTPUT = 'f'
            END !screen output
         IF OUTPUT CONT 'F' THEN !output to file
```

```
       ON *.'file name'
         FOR REFS WITH YEAR CONT *.'year' SORTED BY
           AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
     IF OUTPUT CONT 'B' THEN !output both to screen and file
       BEGIN !screen _file output
         ON TT ON *.'file name'
           FOR REFS WITH YEAR CONT *.'year' SORTED BY
                   AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT ' '
         PAUSE = *.'any key, then <carriage return> to continue'
       END !screen _file output
     PRINT SKIP
    END !year search
IF SEARCHCHOICE = 3 THEN
    BEGIN !citation search
     PRINT SKIP
     OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
     IF OUTPUT CONT 'S' THEN !output to screen
        BEGIN !screen output
         ON TT
           FOR REFS WITH CITATION1 CONT
             *.'word or phrase from citation' OR CITATION2 CONT
             *.'--once again-- word or phrase from citation' SORTED
             BY AUTHOR, YEAR, CITATION1
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT SKIP
          FILEIT = *."Y to print this list to a file, N to continue"
           IF FILEIT CONT 'y' THEN OUTPUT = 'f'
        END !screen output
     IF OUTPUT CONT 'F' THEN !output to file
       ON *.'file name'
         FOR REFS WITH CITATION1 CONT
           *.'word or phrase from citation' OR CITATION2 CONT
           *.'--once again-- word or phrase from citation' SORTED BY
           AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
     IF OUTPUT CONT 'B' THEN !output both to screen and file
       BEGIN !screen _file output
         ON TT ON *.'file name'
           FOR REFS WITH CITATION1 CONT
             *.'word or phrase from citation' OR CITATION2 CONT
             *.'--once again-- word or phrase from citation' SORTED
             BY AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT ' '
         PAUSE = *.'any key, then <carriage return> to continue'
       END !screen _file output
     PRINT SKIP
    END !citation search
IF SEARCHCHOICE = 4 THEN
    BEGIN !keyword search
     PRINT SKIP
     OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
     IF OUTPUT CONT 'S' THEN !output to screen
        BEGIN !screen output
         ON TT
           FOR REFS WITH KEYWORD CONT *.'keyword' SORTED BY
             AUTHOR, YEAR, CITATION1
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT SKIP
          FILEIT = *."Y to print this list to a file, N to continue"
          IF FILEIT CONT 'y' THEN OUTPUT = 'f'
        END !screen output
     IF OUTPUT CONT 'F' THEN !output to file
       ON *.'file name'
         FOR REFS WITH KEYWORD CONT *.'keyword' SORTED BY
           AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
     IF OUTPUT CONT 'B' THEN !output both to screen and file
       BEGIN !screen_file output
         ON TT ON *.'file name'
           FOR REFS WITH KEYWORD CONT *.'keyword' SORTED BY
                   AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT ' '
         PAUSE = *.'any key, then <carriage return> to continue'
       END !screen_file output
     PRINT SKIP
    END !keyword search
IF SEARCHCHOICE = 5 THEN
    BEGIN !reference number search
     PRINT SKIP
     OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
     IF OUTPUT CONT 'S' THEN !output to screen
        BEGIN !screen output
         ON TT
           FOR REFS WITH REFNUM = *.'reference number' SORTED
             BY AUTHOR, YEAR, CITATION1
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT SKIP
          FILEIT = *."Y to print this list to a file, N to continue"
          IF FILEIT CONT 'y' THEN OUTPUT = 'f'
        END !screen output
     IF OUTPUT CONT 'F' THEN !output to file
       ON *.'file name'
         FOR REFS WITH REFNUM = *.'reference number' SORTED BY
           AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
     IF OUTPUT CONT 'B' THEN !output both to screen and file
       BEGIN !screen_file output
         ON TT ON *.'file name'
           FOR REFS WITH REFNUM = *.'reference number' SORTED BY
                   AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
         PRINT ' '
         PAUSE = *.'any key, then <carriage return> to continue'
```

```
               END !screen_file output
             PRINT SKIP
            END !reference number search
        IF SEARCHCHOICE = 6 THEN
         CONTINUESEARCH = 'N'
      END !while continuesearch
   FN$WIDTH(80)
  END !search through references
IF MAINCHOICE = 2 THEN
  BEGIN  !modify references
   FN$WIDTH(80)
    CONTINUEMODIFY = 'Y'
    PRINT SKIP,'THE REFERENCE YOU WANT TO MODIFY WILL BE LOCATED BY',
     'REFERENCE NUMBER.', SKIP,
      '  If you don''t know the reference number, enter N below and go to',
      SKIP, '  SEARCH THROUGH THE REFERENCES to find the one you want.', SKIP
    ANSWER = *.'Y if you know the reference number, N if you don''t'
    IF ANSWER CONT 'N' THEN CONTINUEMODIFY = 'N'
    WHILE CONTINUEMODIFY CONT 'Y'
      BEGIN !while continuemodify
       FN$WIDTH(80)
        PRINT SKIP
        FOR REFS WITH REFNUM = *.'reference number'
          PRINT SKIP 2, '==> REFERENCE TO BE MODIFIED:  ' THEN
          PRINT SKIP, 'REFERENCE NUMBER:  '|REFNUM,
                SKIP, 'AUTHOR    :  '|AUTHOR USING T(80),
                SKIP, 'YEAR      :  '|YEAR,
                SKIP, 'CITATION  :  '|CITATION1|CITATION2 USING T(80),
                SKIP, 'KEYWORDS  :  '|KEYWORD USING T(80) THEN
          PRINT SKIP 2, '==> hit TAB, then RETURN if no change', SKIP THEN
         MODIFY AUTHOR THEN
         MODIFY YEAR THEN
         MODIFY CITATION1 THEN
         MODIFY CITATION2 THEN
         MODIFY KEYWORD THEN
          PRINT SKIP, '==> HERE IS YOUR MODIFICATION:' THEN
          PRINT SKIP, 'REFERENCE NUMBER:  '|REFNUM,
                SKIP, 'AUTHOR    :  '|AUTHOR USING T(80),
                SKIP, 'YEAR      :  '|YEAR,
                SKIP, 'CITATION  :  '|CITATION1|CITATION2 USING T(80),
                SKIP, 'KEYWORDS  :  '|KEYWORD USING T(80)
        PRINT SKIP 2, 'DO YOU WANT TO MODIFY ANOTHER REFERENCE?'
        CONTINUEMODIFY = *.'Y to modify another reference, N to quit'
      END !while continuemodify
   FN$WIDTH(80)
  END !modify references
IF MAINCHOICE = 3 THEN
  BEGIN  !add references
   FN$WIDTH(80)
    PRINT SKIP
    REPEAT *.'number of references you want to add'
      BEGIN !repeat
        PRINT SKIP, '==> new reference  '||RUNNING COUNT|
         '--hit TAB, then RETURN if no entry', SKIP 2,
             '<CTRL Z> will get you out of REFS--but current reference',
              'will not be stored.', SKIP
            STORE REFS
          END  !repeat
        PRINT SKIP, '==> YOUR NEW REFERENCES HAVE BEEN ADDED.', SKIP
       PAUSE = *.'any key, then <carriage return> to continue'
       FN$WIDTH(80)
      END  !add references
    IF MAINCHOICE = 4 THEN
      CONTINUE = 'N'
    FN$WIDTH(80)
  END !of the procedure
FINISH REFS
END-PROCEDURE
```

APPENDIX H. PROCEDURE SEARCH_REFS

In Procedure SEARCH_REFS, below, modify the READY command to include the complete pathname to the DATATRIEVE domain REFS; modify the EXECUTE command to include the complete pathname to the DATATRIEVE procedure LISTREFS. The complete pathname to these DATATRIEVE dictionary objects is "CDD$TOP.DTR$USERS.dictionary.object", where "dictionary" is the name of the DATATRIEVE dictionary (or list of dictionary and subdictionary names, separated by periods) in which the domain or procedure to be executed is located, and "object" is the name of the domain or procedure. In the example here, the "dictionary" is user-dictionary name "SYSCNG", subdictionary name "DELTA", and sub-subdictionary name "REFS"; "object" is REFS or LISTREFS.

```
PROCEDURE SEARCHREFS
!Procedure to search through references in the bibliography
!Written by Carol N. Gerlitz, April 1987
DECLARE CONTINUESEARCH PIC X
                       VALID IF CONTINUESEARCH CONT 'Y' OR
                               CONTINUESEARCH CONT 'N'.
DECLARE FILEIT         PIC X
                       VALID IF FILEIT CONT 'Y' OR FILEIT CONT 'N'.
DECLARE OUTPUT         PIC X
                       VALID IF OUTPUT CONT 'S' OR OUTPUT CONT 'F' OR
                       OUTPUT CONT 'B'.
DECLARE PAUSE          PIC X.
DECLARE SEARCHCHOICE   PIC X
                       VALID IF SEARCHCHOICE EQ '1' OR SEARCHCHOICE EQ '2' OR
                               SEARCHCHOICE EQ '3' OR SEARCHCHOICE EQ '4' OR
                               SEARCHCHOICE EQ '5' OR SEARCHCHOICE EQ '6'.
READY CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.REFS SHARED READ
FN$WIDTH(80)    !clear the screen
PRINT SKIP 2,
  '               ** WELCOME TO THE REFERENCE DATA BASE **', SKIP,
  '               ----------------------------------------'
CONTINUESEARCH = 'Y'
WHILE CONTINUESEARCH CONT 'Y'
  BEGIN !while continuesearch
    PRINT SKIP 2,'WHAT DO YOU WANT TO SEARCH ON?', SKIP,
      '   1. Author', SKIP, '   2. Year', SKIP, '   3. Citation', SKIP,
      '   4. Keyword', SKIP, '   5. Reference number', SKIP 2,
      '   6. EXIT THIS PROGRAM', SKIP
    SEARCHCHOICE = *.'number of your choice'
    IF SEARCHCHOICE = 1 THEN
      BEGIN !author search
       PRINT SKIP
       OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
       IF OUTPUT CONT 'S' THEN !output to screen
          BEGIN !screen output
           ON TT
             FOR REFS WITH AUTHOR CONT *.'author' SORTED
                         BY AUTHOR, YEAR, CITATION1
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
           PRINT SKIP
            FILEIT = *."Y to print this list to a file, N to continue"
```

```
        IF FILEIT CONT ´y´ THEN OUTPUT = ´f´
      END !screen output
   IF OUTPUT CONT ´F´ THEN !output to file
     ON *.´file name´
       FOR REFS WITH AUTHOR CONT *.´author´ SORTED
                   BY AUTHOR, YEAR, CITATION1
          EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
   IF OUTPUT CONT ´B´ THEN !output both to screen and file
      BEGIN ! screen _file output
       ON TT ON *.´file name´
         FOR REFS WITH AUTHOR CONT *.´author´ SORTED
                   BY AUTHOR, YEAR, CITATION1
            EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       PRINT ´ ´
       PAUSE = *.´any key, then <carriage return> to continue´
      END ! screen _file output
   PRINT SKIP
  END !author search
IF SEARCHCHOICE = 2 THEN
  BEGIN !year search
   PRINT SKIP
   OUTPUT = *.´output choice--S (screen), F (data file), B (both)´
   IF OUTPUT CONT ´S´ THEN !output to screen
      BEGIN !screen output
       ON TT
         FOR REFS WITH YEAR CONT *.´year´ SORTED BY
                     AUTHOR, YEAR, CITATION1
            EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       PRINT SKIP
        FILEIT = *."Y to print this list to a file, N to continue"
        IF FILEIT CONT ´y´ THEN OUTPUT = ´f´
      END !screen output
   IF OUTPUT CONT ´F´ THEN !output to file
     ON *.´file name´
       FOR REFS WITH YEAR CONT *.´year´ SORTED BY
                   AUTHOR, YEAR, CITATION1
          EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
   IF OUTPUT CONT ´B´ THEN !output both to screen and file
      BEGIN ! screen _file output
       ON TT ON *.´file name´
         FOR REFS WITH YEAR CONT *.´year´ SORTED BY
                   AUTHOR, YEAR, CITATION1
            EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       PRINT ´ ´
       PAUSE = *.´any key, then <carriage return> to continue´
      END ! screen _file output
   PRINT SKIP
  END !year search
IF SEARCHCHOICE = 3 THEN
  BEGIN !citation search
   PRINT SKIP
   OUTPUT = *.´output choice--S (screen), F (data file), B (both)´
   IF OUTPUT CONT ´S´ THEN !output to screen
      BEGIN !screen output
```

```
       ON TT
         FOR REFS WITH CITATION1 CONT
           *.´word or phrase from citation´ OR CITATION2 CONT
           *.´--once again-- word or phrase from citation´ SORTED BY
           AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       PRINT SKIP
        FILEIT = *."Y to print this list to a file, N to continue"
        IF FILEIT CONT ´y´ THEN OUTPUT = ´f´
      END !screen output
   IF OUTPUT CONT ´F´ THEN !output to file
     ON *.´file name´
       FOR REFS WITH CITATION1 CONT
         *.´word or phrase from citation´ OR CITATION2 CONT
         *.´--once again-- word or phrase from citation´ SORTED BY
         AUTHOR, YEAR, CITATION1
            EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
   IF OUTPUT CONT ´B´ THEN !output both to screen and file
      BEGIN ! screen _file output
       ON TT ON *.´file name´
         FOR REFS WITH CITATION1 CONT
           *.´word or phrase from citation´ OR CITATION2 CONT
           *.´--once again-- word or phrase from citation´ SORTED BY
           AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       PRINT ´ ´
       PAUSE = *.´any key, then <carriage return> to continue´
      END ! screen _file output
   PRINT SKIP
  END !citation search
IF SEARCHCHOICE = 4 THEN
  BEGIN !keyword search
   PRINT SKIP
   OUTPUT = *.´output choice--S (screen), F (data file), B (both)´
   IF OUTPUT CONT ´S´ THEN !output to screen
      BEGIN !screen output
       ON TT
         FOR REFS WITH KEYWORD CONT *.´keyword´ SORTED BY
                     AUTHOR, YEAR, CITATION1
            EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       PRINT SKIP
        FILEIT = *."Y to print this list to a file, N to continue"
        IF FILEIT CONT ´y´ THEN OUTPUT = ´f´
      END !screen output
   IF OUTPUT CONT ´F´ THEN !output to file
     ON *.´file name´
       FOR REFS WITH KEYWORD CONT *.´keyword´ SORTED BY
                   AUTHOR, YEAR, CITATION1
          EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
   IF OUTPUT CONT ´B´ THEN !output both to screen and file
      BEGIN ! screen _file output
       ON TT ON *.´file name´
         FOR REFS WITH KEYWORD CONT *.´keyword´ SORTED BY
                   AUTHOR, YEAR, CITATION1
```

```
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
           PRINT ' '
           PAUSE = *.'any key, then <carriage return> to continue'
          END ! screen _file output
       PRINT SKIP
      END !keyword search
    IF SEARCHCHOICE = 5 THEN
      BEGIN !reference number search
       PRINT SKIP
       OUTPUT = *.'output choice--S (screen), F (data file), B (both)'
       IF OUTPUT CONT 'S' THEN !output to screen
          BEGIN !screen output
           ON TT
             FOR REFS WITH REFNUM = *.'reference number' SORTED
                         BY AUTHOR, YEAR, CITATION1
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
           PRINT SKIP
            FILEIT = *."Y to print this list to a file, N to continue"
            IF FILEIT CONT 'y' THEN OUTPUT = 'f'
          END !screen output
       IF OUTPUT CONT 'F' THEN !output to file
         ON *.'file name'
           FOR REFS WITH REFNUM = *.'reference number' SORTED BY
                       AUTHOR, YEAR, CITATION1
              EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
       IF OUTPUT CONT 'B' THEN !output both to screen and file
          BEGIN ! screen _file output
           ON TT ON *.'file name'
             FOR REFS WITH REFNUM = *.'reference number' SORTED BY
                       AUTHOR, YEAR, CITATION1
                EXECUTE CDD$TOP.DTR$USERS.SYSCNG.DELTA.REFS.LISTREFS
           PRINT ' '
           PAUSE = *.'any key, then <carriage return> to continue'
          END ! screen _file output
       PRINT SKIP
      END !reference number search
    IF SEARCHCHOICE = 6 THEN !exit
      CONTINUESEARCH = 'N'
    FN$WIDTH(80)
  END !while continuesearch
FINISH REFS
END-PROCEDURE
```

APPENDIX I. PROCEDURE LISTREFS

Procedure LISTREFS is called by Procedures REFERENCES and SEARCH_REFS to format references that are output to the terminal screen or to a data file.

```
PROCEDURE LISTREFS
!Written by Carol N. Gerlitz, April 1987
 PRINT SKIP, ´REFERENCE NUMBER: ´, REFNUM(-), SKIP,
       AUTHOR||´, ´|YEAR||´, ´|CITATION1|CITATION2 USING
       T(80), SKIP, ´   KEYWORDS: ´|KEYWORD USING T(80)
END-PROCEDURE
```

APPENDIX J. PROCEDURE LIST_REFS

Use Procedure LIST_REFS to interactively retrieve references from REFS-DTR by any selection criteria and list the information to the terminal screen. Note that the output includes the REFNUM and KEYWORD fields:

In response to the prompt:	Enter:
DTR>	READY REFS <return>
DTR>	FOR REFS WITH fieldname CONT [or other logical operator] string SORTED BY AUTHOR, YEAR, CITATION1 <return>
CON>	EXECUTE LIST_REFS <return>

```
PROCEDURE LIST_REFS
!written by Carol N. Gerlitz, April 1987
  ON TT
      PRINT 'REFERENCE NUMBER: '|REFNUM, SKIP,
        AUTHOR||', '|YEAR||', '|CITATION1|CITATION2 USING T(80), SKIP,
        '   KEYWORDS: '|KEYWORD USING T(80), SKIP
END-PROCEDURE
```

APPENDIX K. PROCEDURE FILE_REFS

Use Procedure FILE_REFS to interactively retrieve references from REFS-DTR by any selection criteria and list the information in a data file that can be printed later:

In response to the prompt:	Enter:
DTR>	READY REFS <return>
DTR>	FIND REFS WITH fieldname CONT [or other logical operator] string SORTED BY AUTHOR, YEAR, CITATION1 <return>
DTR>	EXECUTE FILE_REFS <return>

The procedure will prompt you for a file name and create a data file in which the output includes the REFNUM and KEYWORD fields.

```
PROCEDURE FILE_REFS
!written by Carol N. Gerlitz, April 1987
  ON *.'filename for this list'
    REPORT
    SET LINES_PAGE = 60
    PRINT COL 1, 'REFERENCE NUMBER: '|REFNUM, SKIP,
      AUTHOR||', '|YEAR||', '|CITATION1|CITATION2 USING T(80), SKIP,
      '    KEYWORDS: '|KEYWORD USING T(80), SKIP
    END_REPORT
END-PROCEDURE
```