

ADLIB - Abstract Data LIBrarian
A Bibliographic Retrieval System for Data Set
Abstracts with Interaction with the FIND System

James Nagy
Robert C. Kohberger
John W. Wilkinson

Research supported in part by the Eastern Deciduous Forest Biome,
U.S. - International Biological Program, funded by the National
Science Foundation under Interagency Agreement AG-199, 40-193-69,
with the Atomic Energy Commission - Oak Ridge National Laboratory.

NOTICE: This memo report contains information of a
preliminary nature prepared primarily for
internal use in the US-IBP Eastern
Deciduous Forest Biome program. This in-
formation is not for use prior to publication
unless permission is obtained in writing
from the authors.

ABSTRACT

A series of programs for bibliographic retrieval of data set abstracts are described. The programs are designed to operate with FIND for a total information system. The programs store and retrieve data set abstracts. The programs are written in IBM Assembler Language.

KEYWORDS

Bibliographic Retrieval, Data Set Abstracts, Management Information System.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Richard Park and Mr. Jay A. Bloomfield for their help and constructive suggestions. Special thanks go to Mrs. Sue Walling and Miss Cheryl Frank for their typing of the report.

Introduction

ADLIB - Abstract Data Librarian - is a series of programs that allow for storage and retrieval of data set abstracts as well as providing a programming link with the FIND series of programs which manipulate the numeric data on the database. The FIND system has been described in IBP Memo Report 72-61. The programs are written in IBM 360 Assembler Language, and are designed to operate under the ALPHA telecommunications system. The ALPHA system is a procedure for remote teleprocessing whose most attractive feature is that remote users have access to all system utilities and compilers on the IBM Operating System. The applicability of ADLIB to time sharing systems other than ALPHA has not been fully explored. If a time sharing system at an IBM facility allows for execution of object time load modules then only slight modifications of the programs will need to be made. Currently the UNIVAC 1108 system is being examined to assess the amount of re-programming necessary for the ADLIB programs to run on such a machine.

The ADLIB programs were written so that the Lake George site investigators could easily determine what data were available for their use and to gain access to these data. The ADLIB system tells users what data are available and where on the FIND system they are located. The FIND system actually retrieves the desired data.

Although the Biome Central Information Office has an excellent bibliographic retrieval system, (ORLOOK) there are several advantages to the ADLIB system. Primarily, ADLIB is part of an integrated system for data management. ADLIB and FIND are linked

so that once a user determines that there are data that he needs (through ADLIB), he may immediately retrieve these data from the database (through FIND). Also, because the ADLIB system has been designed specifically for data set abstracts, it is very efficient - although not as flexible as the ORLOOK system - in its retrieval capabilities. Since the programs are resident at the site the difficulties, delays, and costs associated with long distance transmission are alleviated.

Abstracts of any length may be stored on the system. The form of the abstracts is the same as that described in "Documentation and Submission of Data Sets", EDFB Memo Report #72-38 (Brooks and Sayrs). The abstract is shown in Figure 1. For Lake George use the field <DS NAME> is given to name of the data set on the FIND system. An absolute maximum of 4,294,967,295 (one full computer word, $2^{32}-1$) abstracts may be stored, but because of the structure of the retrieval algorithm the actual maximum number of abstracts retrievable is closer to 50,000. This is more fully explained in the section on storage structure. In addition to storing abstracts, the ADLIB system allows for a search and retrieval of abstracts. The system searches an author name, subject category, and/or keywords. These are the only fields that may be searched. Experience to date indicates that these are the fields which most completely describe data. While it may be argued that other fields on the abstract are just as important, because of the retrieval algorithm a limit on the number of fields to be searched had to be made. If necessary, additional fields may be added to the retrieval section. The abstract numbers found from the retrieval are presented to the user and he may then

Figure 1

DATA SET DOCUMENTATION FORM

< DATATITL > _____

< INVESTIG > _____

< MENTOR > _____

< SITE > Coweeta Lake George Lake Wingra Oak Ridge Triangle other _____

< PROCESS >

Terrestrial Primary Production Terrestrial Secondary Production Terrestrial Decomposition Terrestrial Mineral
Cycling Land-Water Interaction Hydrology Meteorology Aquatic Primary Production Aquatic Secondary
Production Aquatic Decomposition Aquatic Mineral Cycling

< KEYWORDS > _____

< PARMLIST > _____

< TAXON > _____

< DATATYPE > Raw Reduced Summarized Literature Derived other _____

< GEOGDESC > _____

< SPONSOR > _____

< DISSEM > This data set may be sent on request to anyone. This data set may be sent on request to anyone partici-
pating in IBP. This data set may be sent on request to anyone in the EDF Biome. This data set may be sent on re-
quest to anyone at the _____ site in the EDF Biome. This data set may be sent only with the author's per-
mission. The data sharing agreement must be signed by the requestor before this data set is sent. The data sharing
agreement need not be signed.

< DS NAME > _____

< DATA ABS > _____

< EXPDESIG > _____

<EXP METH> _____

<DATADATE> _____

<ERORBND> _____

<FLD DESC> _____

<MEDIA> _____

<MISSDATA> _____

<NULLDATA> _____

<AVAIL> _____

<ADD DESC> _____

<ADDAVAIL> _____

<SUP DOCU> _____

<COMMENT> _____

<D SOURCE> _____

print all or some of the retrieval abstracts.

Further sections of this report describe in more detail the storage structure, retrieval algorithm, program modules, and command words of the ADLIB system.

Abstract Storage and Retrieval

Each abstract is one member of a partitioned data set (PDS). The name and location of each abstract is stored in the PDS directory. Use of a PDS permits efficient storage of variable length abstracts. Also stored on the PDS are three symbol tables whose use will be described below.

Each different author, subject category, and keyword is assigned a prime number. The symbol tables contain a list of the authors, subject categories, keywords, and their associated prime numbers. The name of a given abstract is the product of the prime number associated with the author(s) subject category, and keyword(s). It is for this reason that, although a PDS allows for 4,294,967,295 entries, the limiting factor to the number of abstracts is the product of the prime numbers.

The desired author(s), subject category, and keyword(s) to be retrieved are found in the symbol table by a character search routine. If one or more of the words are not found in the symbol table, this means no abstracts meeting the specifications are on file and the user is notified which words were not found in the symbol table. If all the requested words are found in the symbol tables, the product of their associated prime numbers are determined. This product is then divided into all the abstract names; those who divide with zero remainder meet the desired specifications. Such a retrieval algorithm is much faster than a character by

character comparison routine.

When a new abstract is entered on the system, the new author(s), subject category and/or keyword(s) are entered on the symbol tables along with the new prime numbers. At this time, this operation must be done manually by the person inputing the abstract. Because of the limited number of abstracts foreseen to be entered on the abstract file, no automatic procedure has been written. An automatic procedure for updating the symbol table and generating prime numbers may be written if a subsequent need arises.

Program Modules

The ADLIB system was programmed in as modular a form as possible with each module performing a very specific task. This type of programming proved quite useful in de-bugging the system. The following is a list of the program modules that comprise ADLIB, and a description of their use.

1. ADLIB - The main control program which links to the other modules.
2. BEGIN - Initialization module which links to a user information module and gives, if desired, a list of all commands.
3. HELP - A user information routine which takes as input a command name and gives as output a complete description of that command.
4. ERROR - Outputs all error messages to the user.
5. SEARCH - Controls the search of the symbol table and abstract library. The output of this module is the list of the abstract names of the abstracts that meet the specifications in-

put. The search of the symbol tables is done by sub-modules: AUTHS, SUBS, and KEYS.

6. PRINT - Printing routine which takes abstract names and outputs the entire abstract taken from the PDS.
7. DEFINE - Routine which adds new symbols and prime numbers to the symbol tables. This must be done with manual input.
8. INPUT - Determines abstract name from symbol table using sub-modules: AUTH, SUB, KEY, and stores entire abstract on the PDS.
9. RECOG - A finite state machine which recognizes input typed as "AUTHOR= ", "SUBCAT= ", and "KEYWORDS= ". This routine is used in INPUT, DEFINE, and SEARCH.
10. EDIT - Routine which edits previously stored abstracts. (not operational as of 10-16-72)
11. MAINT - A linkage to system utilities which allow for dynamic maintenance of the ADLIB system. The following system utilities may be used:
 - IMASPZAP - modifies load modules
 - IEHLIST - list members of a PDS and VTOC
 - IEHPRGDM - scratch members of a PDS
 - IEPUPDTE - edit members of a PDS

Command Words

The commands presented are those that are actually typed on the remote terminal to invoke certain features of the ADLIB system. All commands are preceded by the "at" sign (@). This signifies to

the programs that command words are to follow. To end execution of the programs "@END" must be typed. In the following command descriptions, braces - {} indicate necessary information and brackets -[] indicate optional information needed for the command word.

1. @HELP [any valid command name]

Gives a description of the format and use of the given command name. If no command name is present, a list of all available commands and how to use the @HELP command is presented

2. @SEARCH {AUTHOR=author name 1,..., AUTHOR=author name n.,
SUBCAT=subject category, KEYWORDS=(keyword 1,...,
keyword n)}

Any number of fields may be searched (one, two, or three). Any number of keywords, authors, or subject categories may be searched. The output from this command is a list of abstract numbers which meet the required specifications.

3. @PRINT {(name 1,..., name n)}

The output of this command is a list of the entire abstract corresponding to the abstract name input. Any number of abstracts may be printed.

4. @DEFINE {Password,} {AUTHOR=author name 1, prime number, SUBCAT=
subject category, prime number, KEYWRDS=keyword 1,
prime number}

This command updates the symbol tables by adding the input fields in the command. This command is password protected so that unauthorized users may not alter the symbol tables.

5. @INPUT {Password,} {AUTHOR=author name 1,..., AUTHOR=
author name n, SUBCAT=subject category, KEYWRDS=
(keyword 1,...,keyword n)}

This command determines the abstract name. Immediately after this command, the user types in the actual abstract in free format. This command is also password protected.

6. @FIND

Links to the FIND system. After typing this the user is no longer operating under ADLIB and must refer to the report on the FIND system.

Appendix 1

Program Listings

Program listings are available from the authors upon request for the cost of postage and handling (\$5.00). Please make checks payable to Rensselaer Polytechnic Institute.