PROPOSED DATAEXCHANGE STANDARDS FOR ARTHROPOD COLLECTIONS

Ronald A. Hellenthal

It is of paramount importance that standard protocols be developed for exchange of electronically represented information between arthropod collections. However, because of the diversity, quantity, and complexity of the information that may be maintained by collections, issues relating to the representation, description, ownership, and control of transferred information can be quite complicated. Not the least of these issues is that of developing standard formats for the organization, structure and representation of exchanged information.

ALTERNATIVE DATAEXCHANGE FORMATS

A database consists of a group of related files (also called "tables") each of which contain a particular set of information in a prescribed format. For example, one file that might be maintained as part of a arthropod collections database could be called "Species Lots". This file could contain information about specimens of a species collected together at the same place and time. Several approaches to exchange of this file are possible. Intuitively, the most straightforward approach would be to agree on a uniform structure for this file as well as for each other type of file that might be exchanged between collections. For example, it might be agreed that specimen identification and collection data for this file when exchanged should include: order, family, genus, species, subspecies, author, collector, date of collection, locality, country, state, number of specimens, and collection method. Using standard database terminology, each of these discrete data elements is called a field (also "variable" or table "column") and the set of fields for each species lot (specimens of a species collected together) is called a record or table "row". Having agreed on the fields to be transferred for each record and their relative order, several formats can be used for the exchange of this information between computers and data management programs.

DELIMITED ASCII

One format commonly used for data exchange is called delimited ASCII. The acronym ASCII stands for "American Standard Code for Information Interchange", and describes a standard that assigns letters, digits, punctuation, and other printable and control characters the values of 0 through 127. These ASCII codes am used by most microcomputers (including the IBM PC and Apple Macintosh) and microcomputer peripheral devices such as printers and plotters and by most non-IBM mini and mainframe computers. The ASCII character set often is extended by additional characters and symbols associated with the values 128-255. However, the specific characters and symbols in this extended ASCII character set may vary substantially from one computer or computer peripheral device to another. Delimited ASCII means that the contents of each field is delimited by a unique character with a second different character used as a separator between fields. Empty fields are represented by paired delimiters without intervening data. If the quotation mark is used as the delimiter and the comma as the separator, a transferred record might appear as:

> "Diptera","Chironomidae","Chironomous","plumosus","",'Linnaeus","Berg", "1942","South Bend","USA","Indiana","23","sweepnet"

Most kinds of computers and many application programs can read and interpret information formatted in this way, so exchange of data in this format is relatively easy. Also, the length of the contents of each field can vary and trailing blanks in fields need not be transmitted. These characteristics of the delimited ASCII format help minimize the time required to transmit data over networks and phone lines and simplify the transfer of information between different types of data

management systems. Despite these advantages, there are three fundamental problems with this kind of data exchange format: 1) any occurrence of a field delimiter character within a data field can result in erroneous interpretation of the field, record and, in the worst case, all subsequent records in the database; 2) all fields must be present in all records since a missing field also will result in erroneous interpretation of the data; and 3) since the database file is undocumented, any misunderstanding between the sender and receiver of the data as to the number of fields, their definitions or order also can result in erroneous interpretations. Thus, the use of this standard forces the establishment of a uniform set of fields and imposes requirements on the contents of the information contained in these fields. While it may, on the surface, seem that avoiding a few specific characters in stored data is a minor inconvenience, this is not necessarily true. For example, image, binary and other non-text data generally must be represented as ASCII characters for exchange between computers. Therefore, it is not always easy to predict the exact contents of fields.

If we were to expand the file exchange format to include the full diversity of information that might be exchanged between collections, the basic simplicity of the format becomes its principal liability. This is because each record is likely to include information for only a small subset of defined fields and there is no way of adding or changing fields to meet special circumstances without the risk of misinterpretation of the exchanged data. While these restrictions may be acceptable for some types of information that might be exchanged between collections, the delimited ASCII format cannot be regarded as suitable for all kinds of data or all collections.

TABULAR ASCII or SDF

An alternative data exchange format that removes the limitation about the kinds of characters that can be included in data fields can be called tabular ASCII or system data file (=SDF) format. This format also requires that all fields be present in a prescribed order but substitutes the requirement that each field be of a prescribed length (usually 1-255) characters for the use of field delimiter and separator characters. If the contents of a field includes fewer characters than the capacity of the field, trailing blanks are added to make up the difference. Thus the contents of, for example, the fifth field of each record will begin at the same relative character position with respect to the beginning of each record. Since records are defined by position rather than by specified delimiters, virtually any printable character data can be transmitted in this format. However, without independent knowledge of the type and length of each field, the contents of each record, field, and even the number of fields contained in a record may be difficult to determine. Furthermore, since blank spaces may have to be added to many fields, data transfer may be considerably slower than that of files in the delimited ASCII format.

dBASE III

Both of the file formats described previously have no internal documentation and, therefore, may be subject to misinterpretation. The internal file structures used by most data management systems solve this problem by including as part of each database file a header record. This record provides such information as the number of fields in each record, the number of records in the database file, and for each field, the order, name, size, and type of data stored. Exchanging information between systems using this format is desirable because there never is a problem associating fields with names, and because some non-character data formats e.g., number, date, logical, etc. can be supported. It is relatively easy to select any subset of the fields in a file for exchange, and the order of the fields contained in each record is not important. The major problem with this is that most of the database management systems use different data formats that generally are proprietary. Thus, effective exchange of data in native database file formats may require general adoption of a common type of database management software. Such a requirement is impractical with arthropod collections where a wide variety of microcomputer and mainframe computer and database management systems am currently in use and, in some cases, are required for consistency between collections within an institution.

Among the commercial microcomputer-based database management systems in common use, the dBASE III file structure is unique in that its internal database file format has been adopted by a large number of different software packages. These include dBASE III PLUS (Ashton-Tate), Clipper and McMax (Nantucket Corp.), dBXL and Quicksilver (WordTech Systems), FoxBASE+/MAC and FoxPro (Fox Software), PC-File (ButtonWare), and others. Database management systems using the dBASE III file format have been adopted by many arthropod collections involved in computerization projects that are using IBM PC and compatible DOS microcomputers, and by several of those using Apple Macintosh systems. R:BASE (Microrim) database structures are used by a few institutions, with database structures such as dBASEN (Ashton-Tate), Paradox (Borland International), and FileMaker (Claris) used by other collections. Nearly all data management systems for the IBM PC and many for the Apple Macintosh can convert files stored in the dBASE III format to their own internal structure, and most (but not as many) of these can convert their file structures to the dBASE III format for export to other programs. Nearly all of these systems also support the delimited and/or tabular ASCII formats, although this requires supplemental entry of field name, length and type information for each database file. Several commercial data conversion programs such as Data Junction (Tools & Techniques, Inc.) also can convert to and from the dBASE III format from a number of other file formats (including those used by spreadsheet programs, etc.). Some field types (e.g., the dBASE Memo format) are not readily exported to other database management systems, and since field name and data type conventions can vary somewhat between systems, some editing and/or other conversion operations may still be required unless a lowest common denominator approach is used in each system. This has the major disadvantage of removing some of the most powerful features of the database management system in the interest of compatibility.

DOCUMENTATION OF DATABASE STRUCTURES

It may be evident that no single format for data exchange has emerged from the previous discussion and, therefore, none is proposed here1. Rather, what is recommended is selection of the most appropriate of the three format options (delimited ASCII, tabular ASCII, dBASE III) for each type of database file, with the caveat that a separate file containing information about the structure of each database file also be exchanged. This structure file provides the information essential for decoding and translating the database file.

The primary components of the file should include the following:

1) The Format/Structure file be of a dBASE, delimited ASCII, or tabular ASCII format.
2) One record (i.e., line) be present for each field in the database file to be exchanged.
3) Each record of the ifie contain the following information:

a) For tabular ASCII format files:

columns 1-10	field name
column 11	field type C=character, N=numeric, L=logical, D=date, M=memo for dBASE ifi format ifies only
columns 12-14	field length in bytes characters
columns 15-17	number of decimal places numeric fields only
columns 18-37	descriptive field name

b) For delimited ASCII format files:

"field name","fleld type","field length",'number of decimals", "descriptive field name"

c) For dBASE format files database structure:

Field	Field Name	Type	Width	Dec
1	FIELD_NAME	Character	10	
2	FIELD_TYPE	Character	1	
3	FIELD_LEN	Numeric	3	0

1. There is an ISO/ANSI approved data description language, ASN.1 (=Abstract Syntax Nomenclature), which provides a compact and portable means for transferring complex data. The standard specifies the data abstraction and provides encoding rules for data types into specific representation. The output is a standard ASCII print file which is both human and machine readable.

   Taxonomic Database Working group (TDWG) of the IUBS Commission for Taxonomic Databases had endorsed their own data exchange language called XDF.

4	FIELD_DEC	Numeric	3	0
5	FIELD_DESC	Character		40

4) Exchanged database and structure files use the following name conventions:

characters 1- 8 file name (include only alphabetic characters and numbers; begin with a letter)
character 9 period "."
character 10-12 one of the following file extensions:
DBF - dBASE III file format
SDF - tabular ASCII file format
DLM - delimited ASCII file format

5) A name for the structure file be used that permits easy association with applicable database files.

PROGRAMS FOR DOCUMENTING THE STRUCTURE OF dBASE III FORMAT DATABASES

Within the dBASE III PLUS language command set and most dialects are commands that automatically can createand interpret structure database files except for the FIELD_DESC field. These are the commands "COPY TO STRUCTURE EXTENDED" and "CREATE FROM EXTENDED FILE". Therefore, it is relatively easy to develop programs that can convert dBASE format database files to or from any of the three recommended data exchange format alternatives. A public domain program that produces structure files in the recommended format including the FLELD_DESC field is available to arthropod collection curators without charge from R. A. Hellenthal (Department of Biological Sciences, University of Notre Dame, Notre Dame, Indiana 46556). This program runs on DOS machines independently of the dBASE command interpreter.

STANDARD FIELD NAMES

For data exchange purposes, database field names should have lengths of not fewer than 2 nor more than 8 characters. Field names must begin with a letter and may contain any combination of letters, numbers, and the underscore character "_". All other characters should be avoided. To facilitate electronic translation of fields between management programs, a field is included in the structure database file named FIELD_DESC. This field is used to equate the field names used by an individual collection management system with generic Descriptive Field Names, such as those used in the "Proposed Model and Database File Structures for Arthropod Collection Management" section of this

report. For example, consider records in the structure database file for family, genus, and species names. Using a tabular ASCII representation, the records might appear as:

FM	C	35	FAMILY
GN	C	20	GENUS
SP	C	30	SPECIES

Another collection may use additional fields, different field names, and/or a different order of fields. In this case, the records in the structure database also might include additional fields for subspecies, subfamily and subgenus, with the family field appearing last rather than first. The structure file representation for these fields might be:

GEN	C	30	GENUS
SBG	C	30	SUBGENUS
SPE	C	35	SPECIES
SSP	C	35	SUBSPECIES
FAM	C	30	FAMILY

By using the names contained in the FIELD_DESC field, equivalence between field names used by different database files or database management systems easily can be established. This is the first step in developing programs for transfer and translation of information between collections. Agreement on the names and kinds of generic descriptive fields also must be established. However, this seems premature unless this approach for data exchange is endorsed by cooperating collections.

STANDARD VOCABULARIES

Another issue is the problem of standardizing the representation of information within database fields. This is considerably more complicated than that of the equivalence of field names. However, a similar approach is possible. As part of the process of building "User Friendly Interfaces" for programs, programmers must develop a standard terminology that is used for data validation and in menu generation. The most appropriate place to store and maintain this kind of information is in database files. Therefore, development of standard structures and equivalence tables for this kind of information both is feasible and desirable. Use of computerized lists of taxonomic names, ecological terms, geographic localities, etc., where possible, can greatly simplify this task. For example, the U.S. General Services Administration maintains and regularly publishes a list of worldwide geographical location codes that commonly are used by Geographic Information Systems and mapping programs. BIOSIS, Inc. maintains and publishes a list of arthropod family names that are used in the Zoological Record that could form the basis of computerized tables of family synonymy. Where computerized species catalogs exist, they can serve the equivalent role for all taxonomic names. It probably is premature to propose specific structures for the maintenance of standard vocabularies, but the development of standards in this area could serve an important role in information exchange and collection data validation.