



DELTA Newsletter

Number 7, April 1991

Note from the Editor — The DELTA Newsletter is designed to promote communication among scientists developing and applying computer technology in the collection, storage, analysis, and presentation of taxonomic data for the production of descriptions, keys, interactive identification, and information retrieval. To achieve this goal the DN will be issued in April and October of each year. Contributions in the form of short comments or long discussions and explanations are encouraged from all developers and users of DELTA format and similar systems. Comments on methods of application, suggestions for improvements, project descriptions, or criticisms of current technology are encouraged. Robert D. Webster, USDA/ARS/SBML, Bldg. 265, BARC-East, Beltsville, MD 20705, USA.

Draft definition of the DELTA format

22 April 1991

M. J. Dallwitz and T. A. Paine

This document is primarily for the benefit of programmers, and contains more detail than would usually be required by users of the DELTA format. Comments are invited; please try to provide them in time for incorporation before the next TDWG workshop (21–22 September 1991).

A. Pelaez is working on syntax diagrams to describe DELTA format. His address is Facultad de Ciencias UNAM, Coyoacan D.F. CP. 04510, A.P. 70–620, Mexico.

General Introduction

When taxonomic descriptions are prepared for input to computer programs, the form of the coding is usually dictated by the requirements of a particular program or set of programs. This restricts the type of data that can be represented, and the number of other programs that can use the data. Even when working with a particular program, it is frequently necessary to set up different versions of the same basic data — for example, when using restricted sets of taxa or characters to make special-purpose keys. The potential advantages of automation, especially in connexion with large groups, cannot be realized if the data have to be restructured by hand for every operation. The DELTA (DEscription Language for TAXonomy) system was developed to overcome these problems (Dallwitz 1980a). It was designed primarily for easy use by people rather than for convenience in computer programming, and is versatile enough to replace the written description as the primary means of recording data. Consequently, it can be used as a shorthand method of recording data, even if computer processing of the data is not envisaged.

Particular attention has been paid to the need to minimize coding errors. The data are written in free format — that is, there is no need to place data in particular columns. The characters may be assigned numbers in any order that suits the user (there is no need to group them by character types, as required by some programs). However, this order need not be adhered to when recording the attributes of a particular taxon. Thus, attributes that are unknown or considered unimportant can be omitted, and later added to the end of the list, if required. An incorrect attribute can be deleted, and the correct one inserted in the same place or at the end. Common character attributes may be made implicit — that is, only the corresponding unusual attributes need appear explicitly in the data.

The system is capable of encoding all of the types of character commonly used for identification and classification: unordered and ordered multistate (including two-state), counts, measurements, and text. Intermediates, ranges, and alternatives can be represented, and distinction is made between ‘variable’,

‘unknown’, and ‘not applicable’. There is provision for comments, which can be used to indicate such things as probability, rarity, uncertainty, qualification, amplification, or references.

There is some redundancy in the coding system, to aid the detection of errors. Most errors have only a local effect, so that a program can continue to scan the rest of the data for other errors.

A format-conversion program, CONFOR, converts DELTA-format data into natural language, or into formats required by several other programs. R. J. Pankhurst’s PANKEY package uses DELTA format directly (Pankhurst 1986).

Introduction to the Definition of the DELTA Format

The DELTA format has been described in successive editions of the DELTA User’s Guide (Dallwitz 1980b, 1984; Dallwitz and Paine 1986), and in the file CHANGES which is distributed with the program CONFOR. Because these publications also describe the operation of CONFOR and various other programs, much of the material in them is not directly relevant to the DELTA format. The material presented here has been extracted from these publications, edited, and annotated.

The essential components of DELTA-format data are normally the ‘character list’, the ‘taxon or item descriptions’, the ‘character types’, the ‘implicit values’, and the ‘character dependencies’. Other essential information can, in principle, be inferred from the above, but it is convenient for programming if it is specified directly: the ‘number of characters’, the ‘maximum number of states’, the ‘(maximum) number of taxa or items’, and the ‘numbers of states’. Programs do not need to be capable of reading a character list if they do not make use of the text contained in it; in this case, some of the latter information cannot be inferred, and must be supplied directly.

The components of DELTA-format data are normally identified by being embedded in ‘directives’ recognized by CONFOR and PANKEY. This embedding is not essential, but is highly recommended, because it makes it easier for users to use the same data files with different programs. For the same reason, it is recommended that the syntax of other CONFOR and PANKEY directives (for example, KEY STATES, EXCLUDE CHARACTERS) be used where appropriate, and that programs be capable of skipping unrecognized directives.

A CONFOR or PANKEY directive consists of a star (*), a control phrase of up to four words, and data. The star must be at the start of a line, or be preceded by a blank. A blank following the star is optional. The control phrase must be in upper-case letters. Only the first three symbols of each word of the control phrase are significant. However, it is recommended that the words be written in full, to make the directive as readable as possible. The data take different forms, depending on the control phrase, and in some directives are absent. A control phrase must be contained in one line, but its data may extend over several lines. A directive is terminated by the star at the start of the next directive, or by the end of the file.

Changes to the format

The following changes have been made to the DELTA format since the publication of Edition 1 of the DELTA User’s Guide (Dallwitz 1980b).

- *Edition 2 of User’s Guide (Dallwitz 1984)*. Text characters added. Implicit values added.
- *Edition 3 of User’s Guide (Dallwitz and Paine 1986)*. Character dependencies added (suggested by Pankhurst). Syntax of ‘variant items’ changed.
- *‘Changes’ file, as at March 1991*. Extreme values for numeric characters added.

It is expected that the format will continue to evolve. Future changes will be presented for discussion in the DELTA Newsletter before being implemented. Also, the following sections contain notes on features that might be considered obsolete and/or subject to change in future versions.

The character list

The taxa are described in terms of a list of characters, each of which consists of a feature and a set of states. Five main *types of character* are recognized: unordered multistate (UM), ordered multistate (OM), integer numeric (IN), real numeric (RN), and text (TE). A multistate character has a fixed number of states (one or more), whereas a numeric character has (in principle) an infinite number of states. (Note. One-state ‘characters’ are allowed mainly for convenience when using a hierarchy of ‘characters’ to represent a taxonomic or geographic hierarchy.) Table 1 shows an example of a character list. Characters 1, 2, and 3 are unordered multistate, 4 is ordered multistate, 5 is integer numeric, 6 is real numeric, and 7 is text. (For two-state characters, the distinction between unordered and ordered is arbitrary.)

Table 1. Example of a character list.

#1. striated area on maxillary palp <presence>/
1. present/
2. absent/
#2. pronotum <colour>/
1. red/
2. black/
3. yellow/
#3. eyes <size>/
1. of normal size <i.e. less than 0.5mm in diameter>/
2. very large <i.e. more than 0.5mm in diameter>/
#4. frons <setae>/
1. with setae on anterior middle and above eyes/
2. with setae above eyes only/
3. without setae/
#5. number of lamellae in antennal club/
#6. length/ mm/
#7. <comments>/

Each *character description* starts with a *feature description*. The feature description starts with a numero (#), which must be at the start of a line or preceded by a blank. The numero is followed by the character number, a full stop (.), and a blank. A blank between the numero and the character number is optional. The feature description is terminated by a slash (/), which must be at the end of a line or followed by a blank. For multistate characters, the feature description is followed by the *state descriptions*. A state description starts with the state number, followed by a full stop and a blank. It is terminated by a slash, which must be at the end of a line or followed by a blank. For numeric characters, the feature description may optionally be followed by the *units* in which the character is measured. The units are terminated by a slash, which must be at the end of a line or followed by a blank. The character numbers must be consecutive integers starting at 1, and must be in ascending order in the character list. State numbers must be consecutive integers starting at 1, and must be in ascending order within each character description. A slash *not* followed by a blank or end of line (e.g. and/or) is allowed, and does not constitute a terminating slash. A missing terminating slash should be detected at the numero marking the start of the next character description, or at the end of the directive, whichever comes first. (Note. CONFOR allows the use of single letters instead of state numbers (see STATE CODES directive in the User’s Guide), but this is not recommended.)

The descriptions of the features, states, and units may contain *comments* enclosed by angle brackets (<>). The opening bracket must be at the start of a line, or be preceded by a blank, a left bracket, or a right bracket. The closing bracket must be at the end of a line, or be followed by a blank, a right bracket, a left bracket, or the slash which terminates that part of the character description. Note that the above definition implies that comments may be nested. Programs may treat nested comments as a single comment, but must ensure that internal brackets are correctly matched. Unmatched brackets should be detected at a terminating slash, at the numero marking the start of the next character, or at the end of the directive, whichever comes first. Angle brackets not in the contexts described above are allowed, and are treated as ordinary symbols, that is, they do not delimit comments. (Note. CONFOR omits character-list comments from much of its output: in particular, they do not appear in natural-language descriptions.)

They may contain any kind of subsidiary material, such as definitions of the terms being used, or references. In some contexts, such as interactive identification, a feature description may be displayed in isolation; comments should therefore be used, if necessary, to make the feature description convey the nature of the character. There is now a CONFOR directive CHARACTER NOTES which may be a more appropriate place for some of the comment material formerly incorporated in the character list. This directive is based on an idea originated by Pankhurst in his ONLINE program.)

The feature and state descriptions should start with lower-case letters (except for proper nouns, etc.). If output is to be automatically typeset, any necessary typesetting marks should be included.

The lines of the character list may optionally contain *sequence numbers*. These guard against accidental disordering of the lines, and facilitate correction of the data by identifying each line uniquely. A sequence number is a positive, real number (for example, 21.43), which starts in the first column of a line and is separated from the data by at least one blank. The sequence numbers must be in ascending order, except that if the first column of a line is blank, then any sequence number is permitted on the next line. If the first non-blank line of the character list has a valid sequence number, then the whole list is assumed to have sequence numbers; otherwise, they are assumed to be absent. (Note. The use of sequence numbers is no longer recommended, and programs need not be capable of handling them. CONFOR can be used to remove them.)

CONFOR and PANKEY require the character list to be preceded by *CHARACTER LIST.

Range of character numbers

A *range of character numbers* has the general form

$$c_1-c_2$$

where c_1 and c_2 are character numbers, and c_1 is less than or equal to c_2 . It denotes all character numbers from c_1 to c_2 , inclusive. For example, 6–9 denotes the characters 6, 7, 8, and 9.

Number of characters

The CONFOR/PANKEY directive to specify the number of characters in the character list is

*NUMBER OF CHARACTERS n

where n is a positive integer.

Maximum number of states

The CONFOR/PANKEY directive to specify the maximum number of states present in any of the characters is

*MAXIMUM NUMBER OF STATES n

where n is a positive integer. The number may be set larger than the actual maximum. It may be set to 1 if there are no multistate characters.

Numbers of states

The CONFOR/PANKEY directive to specify the numbers of states for multistate characters is

*NUMBERS OF STATES $c_1,s_1 c_2,s_2 \dots c_i,s_i \dots$

where c_i is a character number or range of numbers, and s_i is the number of states of the specified character(s). The number of states defaults to 2.

Example

The appropriate directive for the character list in Table 1 would be

*NUMBERS OF STATES 2,3 4,3

Character types

The CONFOR/PANKEY directive to specify the types of the characters is

*CHARACTER TYPES c_1, t_1 c_2, t_2 ... c_i, t_i ...

where c_i is a character number or range of numbers, and t_i is one of the following character types.

UM — Unordered Multistate. Multistate (including 2-state) characters in which the states are not arranged in a natural order.

OM — Ordered Multistate. Multistate characters in which the states are arranged in a natural order.

IN — Integer Numeric. Numeric characters that take only integer (whole-number) values.

RN — Real Numeric. Numeric characters which may take fractional or integer values.

TE — Text.

The default type is UM.

Example

*CHARACTER TYPES 2–3,OM 4,IN 6,IN 10–12,RN 13,TE

(Note. CONFOR also recognizes the ‘exclusive’ types EUM and EOM, which do not allow the coding of multiple values in the taxon descriptions. This facility is rarely used, and would be better implemented as a separate directive.)

Taxon descriptions

A *taxon description* consists of one or more ‘item descriptions’, each of which describes one form or variant of the taxon. Usually one item per taxon is sufficient. However, it may be desirable, for example, to represent two or more subspecies as separate items within one species (taxon), or to represent one variable taxon by several items.

An *item description* consists of the item name followed by a set of attributes. The item name starts with a numero (#), which must be at the start of a line or preceded by a blank. A blank after the numero is optional. The item name is terminated by a slash (/), which must be at the end of a line or followed by a blank. A slash not followed by a blank or end of line is allowed, and does not constitute a terminating slash. A missing terminating slash should be detected at the numero marking the start of the next item description, or at the end of the directive, whichever comes first.

The item name may contain *comments* enclosed by angle brackets (<>). The opening bracket must be at the start of a line, or be preceded by a blank, a left bracket, or a right bracket. The closing bracket must be at the end of a line, or be followed by a blank, a right bracket, a left bracket, or the slash which terminates the item name. Note that the above definition implies that comments may be nested. Programs may treat nested comments as a single comment, but must ensure that internal brackets are correctly matched. Unmatched brackets should be detected at the terminating slash, at the numero marking the start of the next item, or at the end of the directive, whichever comes first. Angle brackets *not* in the contexts described above are allowed, and are treated as ordinary symbols, that is, they do not delimit comments. (Note. Comments in item names were implemented before text characters, and often contained material, such as synonymy, which would now be better placed in text characters. These comments are now generally used for the authority, as in the example below.)

Example

Item name and comment.

Archaeoglenes nemoralis <Ford>/

An *attribute* consists of a character number, together with the *character values* (state numbers or numerical values) that apply to the taxon being described. The special symbols ‘V’, ‘U’, and ‘–’, represent ‘variable’, ‘unknown’, and ‘not applicable’, respectively. These are called *pseudo-values*. The simplest form of an attribute is

c, v

where c is a character number and v is a character value or pseudo-value. Attributes must be separated by at least one blank.

Example

With the characters defined in Table 1, the codes

1,V 4,3 5,- 6,8.5

represent

Striated area on maxillary palp present; or absent. Frons without setae. Number of lamellae in antennal club not applicable. Length 8.5mm.

The general form of an attribute is

$c\langle e_0 \rangle$

or

$c\langle e_0 \rangle, r_1\langle e_1 \rangle / r_2\langle e_2 \rangle / \dots r_n\langle e_n \rangle$

where c is a character number, r_i is a value or combination of values (see below), '/' is a separator denoting 'or', and ' $\langle e_i \rangle$ ' is optional extra information (a *comment*). Blanks or line endings are permitted within e_i , but not elsewhere within the attribute. r_i takes one of the forms

v

$v_1 \& v_2 \& \dots v_m$

$v_1 - v_2 - \dots v_m$

where v is any character value or pseudo-value, v_j is any character value (not a pseudo-value), '&' is a separator denoting 'and', and '-' is a separator denoting 'to'. Text characters are coded simply as $c\langle e_0 \rangle$.

Example

The codes

1,1/2<rare> 2,2/2&3<striped> 3,1-2 6,7-8.5 7<possibly two species>

represent

Striated area on maxillary palp present; or absent <rare>. Pronotum black; or black and yellow <striped>. Eyes of normal size to very large. Length 7 to 8.5mm. Possibly two species.

When the separator '-' is used with ordered multistate or numeric characters, the components of r_i must be in ascending order, and r_i denotes all values between v_1 and v_m .

Examples

The attributes 4,1-3 and 4,1-2-3 are equivalent, and indicate that setae may be on the anterior middle and above the eyes, above the eyes only, or absent. However, the attributes 2,1-3 and 2,1-2-3 are not equivalent: the former denotes colours between red and yellow (red, orange, and yellow, but not black), while the latter denotes red, black, yellow, and their intermediates.

For numeric characters, ' v_1- ' and/or ' $-v_m$ ' may be enclosed within parentheses, to denote *extreme* values, and there may be at most 3 *normal* values (those *not* enclosed in parentheses).

Examples

These attributes are valid:

5,(1-)2

5,(1-)2-3

5,(1-)2(-3)

5,(1-)2-3(-4)

5,(1-)2-3-4(-5)

These attributes are invalid:

5,(1-2-)3

5,(1-)2-3-4-5

(Note. It is proposed that a future version of the format will remove many of the restrictions in the above definitions. For example, embedding of comments within r_i will be permitted, and the use of extreme values will be permitted in association with multistate characters and with the separators '/' and '&'.)

Attributes may be recorded in any order within an item. A missing attribute is equivalent to an attribute with pseudo-value U (except for variant items in a multi-item taxon, or if character dependencies or implicit values have been specified — see below).

Example

The item
 # Species A/ 1,1 3,2 5,2 6,9 4,1
 is equivalent to
 # Species A/ 1,1 2,U 3,2 4,1 5,2 6,9

The items of a *multi-item taxon* must be grouped together. The items are identified as belonging to the same taxon by having a plus sign after the numero of the second and subsequent items (#+). The first item is called the main item, and the other items are called variant items. Missing attributes in the main item denote characters with unknown values (or dependent or implicit values), in the usual way. Missing attributes in the variant items denote attributes that are the same as in the main item.

Example

The 2-item taxon
 # Species B (Australia)/ 1,1 2,1/2<rare> 3,1 5,3 6,5–6
 #+ Species B (New Guinea)/ 3,2 5,U
 is equivalent to
 # Species B (Australia)/ 1,1 2,1/2<rare> 3,1 5,3 6,5–6
 # Species B (New Guinea)/ 1,1 2,1/2<rare> 3,2 5,U 6,5–6

(Note. A program that does not implement variant items should nevertheless detect the ‘#+’ and issue a warning. CONFOR can be used with the INSERT REDUNDANT VARIANT ATTRIBUTES directive to produce DELTA-format data in which all the relevant information from the main items is explicit in the variant items. This process can be reversed by means of the OMIT REDUNDANT VARIANT ATTRIBUTES directive. It is proposed that future versions of DELTA format will have provision for specifying a taxonomic hierarchy, and for passing attribute information up and down the hierarchy. The ‘variant items’ facility, in its present form, will be then be redundant, and will be removed.)

Some characters have a common or ‘usual’ state value, which describes the great majority of taxa, and a rare or ‘unusual’ state value, which describes only one or a few taxa. It is possible to specify that the common value is to be *implicit* unless otherwise indicated (see implicit values, below). Then only the rare values need be explicitly coded in the items. Besides reducing the amount of coding, this has the added advantage that the common values can be omitted from natural-language descriptions.

Sometimes certain attributes imply that other characters are inapplicable. A common example is a character that specifies the presence or absence of some structure: if the structure is absent, then all characters that further describe that structure are inapplicable. If this *dependency* of the characters is specified (see character dependencies, below), then the inapplicable characters can be omitted from items, instead of being explicitly coded as inapplicable.

The lines of the item descriptions may optionally contain *sequence numbers*. These guard against accidental disordering of the lines, and facilitate correction of the data by identifying each line uniquely. A sequence number is a positive, real number (for example, 142.105), which starts in the first column of a line and is separated from the data by at least one blank. The sequence numbers must be in ascending order, except that if the first column of a line is blank, then any sequence number is permitted on the next line. If the first non-blank line of the item descriptions has a valid sequence number, then all of the items are assumed to have sequence numbers; otherwise, they are assumed to be absent. (Note. The use of sequence numbers is no longer recommended, and programs need not be capable of handling them. CONFOR can be used to remove them.)

CONFOR and PANKEY require the item descriptions to be preceded by *ITEM DESCRIPTIONS.

Number of taxa

The CONFOR/PANKEY directive to specify the maximum number of items is

*MAXIMUM NUMBER OF ITEMS n

where n is a positive integer. The number specified should be greater than or equal to the actual number of item descriptions.

Implicit values

The CONFOR directive IMPLICIT VALUES permits certain attributes or state values to be omitted from items. The omitted attributes or values are assigned default values. The directive takes the form

*IMPLICIT VALUES $c_1, s_1 : t_1$ $c_2, s_2 : t_2$... $c_i, s_i : t_i$...

where c_i is a character number or range of numbers, and s_i and t_i are state values. ‘: t_i ’ is optional. Numeric or text characters must not be specified.

If the character specified by c_i does not appear in an item, then the character is assigned the value s_i (unless the item is a variant item, in which case the value(s) are copied from the main item). (Note. When using CONFOR to translate into DELTA format or natural language, the missing characters are omitted from the descriptions, unless an INSERT IMPLICIT VALUES directive is in force.)

If ‘: t_i ’ is present in the IMPLICIT VALUES directive, and the character specified by c_i appears without a value in an item, it is assigned the value t_i (unless the item is a variant item, in which case the value(s) are copied from the main item). (Note. When using CONFOR to translate into DELTA format, only the character number is output, unless an INSERT IMPLICIT VALUES directive is in force.)

Example

If the directive

*IMPLICIT VALUES 1–3,2:1 5,1

is in force, the attributes

1,3 3

are equivalent to

1,3 2,2 3,1 5,1

(except in natural-language descriptions).

(Note. The main purpose of implicit values is to improve natural-language descriptions by omitting common character states. CONFOR can be used with the directive INSERT IMPLICIT VALUES to obtain DELTA-format data without implicit values. However, this process cannot be reversed. Programs that do not implement implicit values should detect the IMPLICIT VALUES directive and issue a warning.)

Character dependencies

The CONFOR/PANKEY directive DEPENDENT CHARACTERS specifies sets of characters — the ‘dependent’ characters — that are inapplicable when certain other characters — the ‘controlling’ characters — take certain values. The controlling characters must be multistate characters. The directive takes the form

*DEPENDENT CHARACTERS $c_1, s_1 : d_1$ $c_2, s_2 : d_2$... $c_i, s_i : d_i$...

where c_i is a character number (the controlling character), s_i is a set of state numbers, and d_i is a set of character numbers (the dependent characters). s_i takes the form

$t_1/t_2/.../t_j/...$

where t_j is a state number. d_i takes the form

$e_1:e_2:...e_k:.../...$

where e_k is a character number or range of numbers. A dependent character may be associated with more than one controlling character. In an item description, a dependent character can take values other than ‘–’ only if each of its controlling characters takes at least one state value that does *not* belong to the set of states s_i specified for that controlling character.

Examples

*DEPENDENT CHARACTERS 4,2:16 9,1:20 10,1/3:12–13:20:30–33

The following combinations of attributes in an item are permitted.

4,2 9,1 10,3 12,– 13,– 16,– 20,– 30,– 31,– 32,–

4,2 9,1 10,3 (*equivalent to the above*)

4,1 16,1

10,1/2 12,1/–

10,1/2 12,1

9,2 10,2 20,1

The following combinations of attributes in an item are not permitted.

4,2 16,1

16,1

9,2 10,3 20,1

References

Pankhurst, R. J. 1986. A package of computer programs for handling taxonomic databases. *CABIOS* 2: 33–9.

Dallwitz, M. J. 1980a. A general system for coding taxonomic descriptions. *Taxon* 29: 41–6.

Dallwitz, M. J. 1980b. User's guide to the DELTA system. A general system for coding taxonomic descriptions. CSIRO Aust. Div. Entomol. Rep. No. 13, 71pp. +microfiche.

Dallwitz, M. J. 1984. User's guide to the DELTA system: a general system for coding taxonomic descriptions. 2nd edition. CSIRO Aust. Div. Entomol. Rep. No. 13, 93pp.

Dallwitz, M. J., and Paine, T. A. 1986. User's guide to the DELTA system: a general system for processing taxonomic descriptions. 3rd edition. CSIRO Aust. Div. Entomol. Rep. No. 13, 106pp.

M. J. Dallwitz and T. A. Paine, CSIRO Division of Entomology, GPO Box 1700, Canberra, ACT 2601, Australia. Telephone 61-6-246-4911. Fax 61-6-246-4000. Telex AA 62309.

Miscellaneous notes**First DELTA workshop in Portugal**

From 14 to 16 January 1991 an introductory workshop on 'Taxonomic Tools by Computer' in Coimbra (Portugal) was organized by Maria Teresa Almeida (Botanical Institute and Garden) with the participation of Les Marcus (American Museum of Natural History, New York), Antonio G.-Valdecasas, Elisa Bello and Jose Ma Becerra (Museo Nacional de Ciencias Naturales, Madrid). Theoretical and practical sessions were devoted to DELTA (the Spanish version), statistical and morphometric software, the Spanish Directory of Taxonomists (the program, DIRTAX, and the database included), and a set of programs dealing with museum collections (worked jointly with J. Northern, U.C.L.A.). Maria Teresa Almeida gave an introductory session to numerical taxonomy.

Sessions were very intensive (more than ten hours daily) and the participants (30 from different Research Institutes and Universities in Portugal, including Azores) were very interested in the sessions. Free software including DELTA in Spanish and DIRTAX was distributed to the participants. Useful suggestions for future developments were recorded. The excitement demonstrated by the participants was a very good signal of the health of DELTA and the increasing demand of useful (and free, if possible) software for taxonomy and systematics in this Biodiversity decennium. The lecturers appreciate the Portuguese hospitality and friendship as well as the extensive help of the students that helped to organize the Workshop.

DELTA in Spanish

Through the kindness of Dr. M. J. Dallwitz we have been able to obtain CONFOR, KEY and DIST source codes and adapt them to a directives vocabulary in Spanish. The idea is to promote DELTA usage among Spanish speaking taxonomists. Traditional taxonomists are usually afraid of computers, and if the

command language is not their own, they are difficult to convince of the advantage of an automatic system. Command language in Spanish improves the friendliness of DELTA for our users.

A first version of DELTA in Spanish is being tested through several workshops, the first held in Madrid, May 1990 and the second in Portugal. One more is planned for this year in Spain. Some demonstrations have been done during the annual meetings of several Spanish scientific societies. DELTA is distributed free among the Spanish users.

EDEL (Editor DELta) is a program to automatically build the three basic files — characters, specifications and items — in DELTA. Despite Dallwitz' (Dallwitz and Paine 1986) assertion that it is not necessarily a rigid format for the manipulation of the taxonomic information, the files structure before running CONFOR is embarrassing enough as to discourage those taxonomists not very interested in computers. EDEL has been done to make easier that first data introduction by

1. inputing only once the information that must be present in more than one file; and
2. doing counts and balances of characters and states automatically.

In this way, we found that the taxonomist can pay more attention to the content of the biological information that she/he has to deal with, than to the formal aspects it should have to be understood by the computer. EDEL is limited to 60 taxa, 120 characters, and 24 lines of 80 columns text of character definition for each taxa. It is menu driven, and has controls at several states of the program, so, e.g., no more than one state of a character can be define as implicit, integer and real numeric characters cannot have states except for the unit of measurement, etc. EDEL is distributed free of charge in English and Spanish versions (directives in English or Spanish).

TRANSNT or TRANSNTE (English and Spanish version respectively) are programs that convert the output files from Dallwitz's DIST program (the files *.dis and *.nam) into a file that is readable by the NTSYS-pc package. NTSYS-pc (Exeter Publishing Ltd, 100 North Country Rd, Building B, Setauket, New York 11733), is a numerical taxonomic package developed by Dr. F. James Rohlf, that implements many of the procedures described in the text 'Numerical Taxonomy' by Sneath and Sokal (1973). The new versions of TRANSNT(E) distinguish between NTSYS-pc ver. 1.3 and later, and versions earlier than 1.3. In NTSYS-pc ver. 1.3, Dr. Rohlf (comm. pers.) introduced a new format for comment lines, which are preceded by a quote character ", ', or '. Versions earlier than 1.3 indicate in the first line the number of comment lines that follow.

Dallwitz's DIST program uses Gower's similarity coefficient (see for example Dunn and Everit (1982), 'An Introduction to Mathematical Taxonomy', Cambridge University Press, for a clear explanation) to convert the data matrix to a distance matrix. If there are no numerical data in the original data matrix, the TRANSNT allows one to convert the Gower matrix into a similarity matrix (sensu Sneath and Sokal 1973). If numerical data have been defined, it would be more correct to use the 'distance option' of TRANSNT.

TRANSNT(E) was written by *Juan Elvira & Antonio G.-Valdecasas* at the *Museo Nacional de Ciencias Naturales, Jose Gutierrez Abascal, 2, 28006-Madrid, Spain*, and is distributed free in its Spanish or English version.