DELTA Newsletter

ISSN 1320-2979 Number 9, October 1993

CONTENTS

EDITORIAL **1**
PRELIMINARY SUGGESTIONS FOR NEW FEATURES FOR THE DELTA SYSTEM
by M.J. Dallwitz, T.A. Paine and E.J. Zurcher **2**
COMMENTS ON NEW FEATURES FOR DELTA SYSTEM by R.J. Pankhurst **14**
A REPLY TO RICHARD PANKHURST'S COMMENTS by M.J. Dallwitz **16**
CORRESPONDENCE ON DELTA ENHANCEMENT PROPOSALS
by J.H. Kirkbride and M.J. Dallwitz **18**
SOME QUESTIONS AND NOTES ON THE NEW FEATURES by E.J. Gouda **22**
A REPLY TO ERIC GOUDA'S COMMENTS by M.J. Dallwitz **23**
SUBSCRIBE TO THE DELTA NEWSLETTER **24**

EDITORIAL

Welcome to number 9 of the DELTA Newsletter. In this special issue we focus on the suggested new features for the DELTA system and gather together much of the recent discussion and comment. As a result of the need to make space to present the debate many of the regular features have been carried forward to the next issue.

In the main article Mike Dallwitz presents a detailed explication of the proposed new features, illustrating most with useful examples. Most proposals have been tentatively accepted by the DELTA development team, however, there are others presented which need much further consideration. Comment, criticism and discussion of all the proposals by the DELTA user community is encouraged, and in many cases necessary, in order to discern the full impact of the suggested changes to the DELTA system.

The subsequent articles by Pankhurst, Dallwitz, Kirkbride and Gouda encapsulate some of the debate stimulated by the original document and should further encourage readers to submit their own comments or suggestions to the DELTA development team. The Kirkbride and Dallwitz article is an edited summary of their correspondence over the Internet, which highlights the increasingly important role that this form of communication is playing world-wide. Details on how to access DELTA System files via the Internet are provided at the top of the DELTA registration form on page 23 of issues 8 and 10 of the DELTA Newsletter, as well as the method of signing up to the Taxacom Technical mailing list, via which DELTA program updates are announced.

The main article was first distributed over the Internet in September 1993, and some of the following articles are in response to this original document, however, the version published here contains corrections and updates to 10 December 1993. This issue of the DELTA Newsletter has been held over to be distributed with the next (ie. number 10, April 1994) because of the special nature of this issue, the desire to bring the latest debate to those members of the DELTA community without access to the Internet, not to mention the usual funding constraints.

Alex R. Chapman, Terry D. Macfarlane*, Nicholas S. Lander
Western Australian Herbarium
Department of Conservation and Land Management
PO Box 104 Como, WA 6152
AUSTRALIA

Telephone +61 9 3340500 Fax +61 9 3340515 Email (via M.J. Dallwitz: INTERNET md@ento.csiro.au)

* Based at Manjimup Research Centre, Department of Conservation and Land Management, Brain Street, Manjimup, WA 6258

Preliminary Suggestions for New Features for the DELTA System

M.J. Dallwitz, T. A. Paine, and E. J. Zurcher

CSIRO Division of Entomology, GPO Box 1700, Canberra ACT 2601, Australia
Phone +61 6 246 4075 Fax +61 6 246 4000 Internet delta@ento.csiro.au

Introduction

This discussion assumes familiarity with the current DELTA system - that is, the DELTA format, and the DELTA programs developed in the CSIRO Division of Entomology, as described in Edition 4 of the User's Guide to the DELTA System (Dallwitz, Paine and Zurcher 1993). The User's Guide describes more than 150 'directives' used by the programs. The DELTA format comprises those directives which define the meaning of the coded descriptions: CHARACTER TYPES, IMPLICIT VALUES, DEPENDENT CHARACTERS, CHARACTER NOTES, CHARACTER LIST, and ITEM DESCRIPTIONS (plus a few redundant directives such as NUMBER OF CHARACTERS which are convenient for programming). The rest of the directives specify how the programs are to process the coded descriptions for various purposes, such as the generation of natural-language descriptions or keys. Most of the new or enhanced directives described below are extensions of the DELTA format, but some indicate how the new DELTA format will (or could) be used by new programs to overcome deficiencies in the capabilities of current programs.

The central program of the new DELTA system will carry out the functions of the current CONFOR and DELFOR, as well as data entry and editing. A name for this program has not yet been decided. For the time being, we will refer to it as 'CONFOR', and add the qualifiers 'old' or 'new' if necessary to avoid ambiguity.

An enhanced DELTA format will serve as a data-exchange medium. It will necessarily be considerably more complex than the current format, because of the extra features to be supported. We envisage that the old format will be a subset of the new, with the possible exception of some rarely used features such as 'variant items', which may be replaced by more general but incompatible constructs. However, at this stage we would not rule out a complete redesign of the format if this seems desirable. In that case, separate provision would need to be made for reading existing data files.

The new CONFOR will be able to read and write complete DELTA-format files, but this facility will normally be used only for data exchange. Data input and manipulation will normally be interactive. Therefore, in designing most aspects of the format, we can lean towards ease of programming rather than ease of use. Several people have suggested that many directives could be dispensed with, and the information they contain could be inferred or placed elsewhere. For example, the number of characters and numbers of states can be inferred from the character list, and the character types, dependencies, and notes could be embedded in the character list. Some of these proposals would certainly make the current CONFOR easier for users, but they seem to have no advantage in the context of the new CONFOR. They would often make programming more difficult, especially for storage allocation and for ignoring unwanted information (e.g. character notes). It might even be better to split some existing directives; for example, the 'exclusive' property of characters, now expressed as the character types EUM and EOM, would more logically be expressed in a separate directive (or eliminated completely).

The user interface and the internal workings of the new CONFOR will be able to shield users from many of the complexities of the current system. However, the 'attribute' (cell of the data 'matrix' - see Section 2.2 of DELTA User's Guide for definition) may remain an exception, because there may be no simpler way of expressing the potentially complex information in an attribute (for example, order of states and association of comments with particular states). The user will have the option of entering an attribute directly in DELTA format, as in the current program ENTITEM, or building it up piece by piece via menus. In the latter case, the details of generating the required syntax will be handled by the program. The natural-language interpretation of the attribute will be visible during entry and editing of the attribute. This will reduce the need for the user to view the attribute in DELTA format, but it may sometimes be necessary to do so, particularly while editing an existing attribute. The format of attributes should therefore be kept as simple as possible.

Accepted Proposals

The proposals in this section have been tentatively accepted by the development team as desirable and feasible, but are completely open for discussion and changes.

Command syntax and usage

Command words will not be case sensitive.

It may be desirable to distinguish between 'interactive' CONFOR commands and 'batch' CONFOR directives. The interactive commands would be similar to INTKEY commands, and would need be suitable for incorporation in a hierarchical menu structure. The batch directives would be similar to the current CONFOR directives, and would be executable only from within files (although they would be assembled and edited interactively).

The syntax of INTKEY commands was made different from that of the current CONFOR directives to simplify interactive entry of commands. In INTKEY, a single command may specify the required action, a set of taxa, and a set of characters, whereas in CONFOR at least three directives would be required for this. In most INTKEY commands, 'space' is the only delimiter required, but parentheses (to separate a set of taxa from a set of characters) and quotes (to enclose a parameter which has internal spaces) are sometimes required.

Delimiter symbols

The special delimiter symbols will be user definable and redefinable. The defaults will be as in the current DELTA format, plus the symbol '|' to indicate that the following symbol is to be interpreted literally, and '!' to indicate that the next two symbols are a hexadecimal representation of a byte. The new definitions will not come into force until the end of the redefining directive.

Example:

```
*DEFINE DELIMITERS
#1. |\ <literal>
#2. |! <hexadecimal>
#3. $ <start of directive>
#4. |* <start of character, item, etc.>
#5. { <opening bracket>
#6. } <closing bracket>
. . .
#.
```

With these definitions in force, the start of a character list might read

```
$CHARACTER LIST
*1. {synonyms}/
```

The literal marker, applied to an alphabetic character, will protect it from case changes, e.g. |mRNA.

On output, there will be provision for omission of '|' and '!', for passing them through unaltered, and for substituting other symbols. If '!' is omitted, the hexadecimal codes will be replaced by the corresponding byte.

All delimiters will be effective as single symbols. In particular, delimiters will not need to be associated with spaces, as they are in some contexts in the present DELTA format. (This may require changes to existing data. For example, 'and/or' will have to be replaced by 'and|/or'. It should be possible to carry out these changes automatically.)

Spaces and line length

There will be no limit on line length, but a short length is recommended to facilitate viewing of the files. Lines may be broken or any number of spaces included, at any space or after any delimiter except angle brackets, parentheses, and decimal points.

Alternative languages

The command words, error messages, and help in CONFOR will be readily translatable into other languages. This will be implemented as in the current INTKEY).

There will be provision for alternative-language versions of all appropriate data elements, such as character lists and comments. The alternative-language versions will be flagged by means of 'coded comments' (see below). Data elements which are not flagged as being in a particular language will be assumed to be in a default language. In CONFOR, it will be possible to restrict the user's view to any language or set of languages.

Examples:

```
*ALTERNATIVE LANGUAGES English French Chinese
*DEFAULT LANGUAGE English
*CHARACTER LIST
#1. <longevity of plants>/
1. annual <or biennial, without remains of old sheaths or culms>/
2. perennial <with remains of old sheaths and/or culms>/
. . .
*CHARACTER LIST <@French>
#1. plantes <longévité>/
1. annuelles <sans vestiges de gaines ou de chaumes>/
2. vivaces <avec vestiges de gaines ou de chaumes>/
. . .
*VIEW LANGUAGES English French
```

Keywords

Character and taxon keywords will be definable to represent groups of characters and taxa. However, the system will differ from the current INTKEY in the following ways. (1) The system-defined keywords (such as ALL and REMAINING) will be preceded by a period (e.g. .ALL). (2) Taxon names will not be defined as keywords. The purpose of these changes is to simplify the internal handling of the keywords, particularly when moving between different languages. The changes will also be made in INTKEY.

Indexed lists

There will be provision for defining numbered and alphabetic lists of entities such as references, countries, and taxonomic names. These lists will be used as character states or as 'coded comments' (see below).

Examples:

```
*NUMBERED LIST <@English> continents
#1. Europe
#2. Africa
#3. Asia-Temperate
```

. . .
*ALPHABETIC LIST references
#1. **Abel, D. J., and Williams, W. T.** (1985). A re-examination of four classification fusion strategies. *Comput. J.* 28, 439-43.
#2. **Brunt, A., Crabtree, K., and Gibbs, A.** (eds) (1990). Viruses of tropical plants. Descriptions and lists from the VIDE database. (CAB International: Wallingford.)
#3. **Burr, E. J.** (1970). Cluster sorting with mixed character types. II. Fusion strategies. *Austral. Comput. J.* 2, 98-103.
. . .

An alphabetic list need not be in alphabetic order when represented in DELTA format, but will be automatically maintained in alphabetic order within CONFOR.

In contexts where an element of a particular list is the only possibility (e.g. in attributes of type LIST - see below), a list element will be represented by its number alone. In other contexts, it will by represented by a 'comment' of the form <@*list-name number*>, e.g. <@ref 3>.

Lists will be able to contain references to other lists.

Example:

*ALPHABETIC LIST authorities
. . .
#10. Harms
#11. van Meeuwen
. . .
*ALPHABETIC LIST species_names
#1. Pericopsis elata (<@auth 10>) <@auth 11>
. . .

It will be possible to define relations between lists, such that an element of one list 'belongs to' an element of another.

Example:

*NUMBERED LIST regions
#1. Northern Europe
#2. Middle Europe
#3. Southwestern Europe
#4. Southeastern Europe
#5. East Europe
#6. Northern Africa
#7. Macaronesia
#8. West Tropical Africa
. . .
*RELATED LISTS continents regions
1,1-5 2,6-15 . . .

New character type: LIST

The states of these characters will be the elements of a numbered or alphabetic list (see above).

Examples:

```
*CHARACTER TYPE 5,LI
*CHARACTER LIST
. . .
#5. anatomical references/ references/
. . .
*ITEM DESCRIPTIONS
. . . 5,21/55 . . .
```

New character type: CYCLIC

This type would be used for information such as time of year. It would improve natural-language descriptions and calculation of distances.

Examples:

```
*CHARACTER TYPE 9,CY
*CHARACTER LIST
. . .
#8. flowers <whether all year>/1. all year/ 2. <not all year>/
#9. flowers <months>/ 1. January/ 2. February/. . . 12. December/
. . .
*ITEM DESCRIPTIONS
. . . 8,2 9,11-1 . . .
```

Character identifiers

It will be possible to define alphanumeric identifiers for characters. This will simplify the merging of separately maintained databases.

Example:

```
#<awn01>45. awns <presence>/
```

Alternative wordings for characters

Alternative wordings will be able to be incorporated in characters lists, and invoked selectively. The number of states in a character will be allowed to vary to correspond to the key-states currently in force.

Examples:

```
. . .
#38.0. upper glume <of female-fertile spikelets, mid-zone nerve number>/ nerved/
#38.1. upper glume of female-fertile spikelets/
        1. without nerves or with a single nerve/
        2. with two or more nerves/
```

. . .
#297. papillae <presence in the abaxial leaf blade epidermis>/
1. present/
2. absent/
#297.1. abaxial epidermal papillae <presence in the leaf blade>/
1. present on the leaf blade/
2. absent from the leaf blade/
. . .
*KEY STATES 38,~1/2~
*CHARACTER WORDING 38,1 297,1

Taxon names

An alternative form for a taxon name will be a list element.

Example:

Poa L./ or <@species 17>/

Selective omission of the authority will be possible only for the list-element form of the name, as in the example under indexed lists. (The current mechanism, where the comment in a name is interpreted as the authority, will not be supported.)

Synonyms

Synonyms will normally be represented in a character or characters of type LIST. The characters involved will be indicated in a SYNONYM CHARACTERS directive. The list will be the same as that used for the taxon names. A mechanism will be provided in CONFOR to simultaneously search all the synonym characters and the taxon names, so that a taxon can be accessed via any of its synonyms.

Coded 'comments' (subsidiary information)

Coded 'comments' will allow the incorporation of subsidiary information which will be interpretable by programs. Coded comments will be embedded in ordinary comments (that is, enclosed in angle brackets), and will comprise the symbol '@', a single-word 'comment identifier', and the coded information. A coded comment will be terminated by the next coded comment, or by the closing angle bracket of the comment. CONFOR will recognize the following coded comments.

<@probability x> - probability or frequency of a state value.

<@x%> - alternative form of 'probability' comment.

<@rarely> - a low probability for a state value. The value of this probability will be settable.

<@about> - qualifies numeric values. The extent of the uncertainty would be specifiable by generalizations of the current ABSOLUTE/PERCENTAGE ERRORS directives.

<@?> - marks guessed values.

<@reliability x> - specifies a reliability for an attribute, to modify the overall reliability of a character. This information would be important for key generation (although a key-generation program to use it is not currently planned).

<@edit editing-commands> - apply TED editing commands to the natural-language description before output. (Should provision be made for alternatives for different languages?)

<@use n: s> - specifies alternative character values for particular applications such as classification or identification. n is an integer which specifies a category of uses, and s is a set of values of the form v_1-v_2/ v_3-v_4/... . The values appropriate for an application would be invoked (in place of the values in the attribute) by specifying a use-category in a USE ALTERNATIVE VALUES directive.

<@up> - attribute has been generated from information passed up the taxonomic hierarchy.

<@down> - attribute has been generated from information passed down the taxonomic hierarchy.

<@note: text> - uninterpreted comment (replacing the current 'inner comment' mechanism).

In addition, any numbered or alphabetic list name will be recognized as a comment identifier. The omission of coded comments from natural-language descriptions will be controllable independently for each identifier.

Examples:

*ITEM DESCRIPTIONS
. . . 10,1/3<@prob .1> 11,2/<@rarely>4 12,1<@ref 135 322> 13,2/<occasionally @5% @ref 54>3 14,1<@note: check in fresh specimens> 15,<@about>15-<@about>20 16,2<@use 1: 1/2> 17,8.5-10<@use 1: 7-12><@use 2: 9> 18,2/3<@use 2: 2> 19,1<usually truncate><@edit d; (;i;, ;d;);>

Use-category 1 gives a broader set of values, and would be invoked for identification. Use-category 2 gives a narrower set of values, and would be invoked for classification. The editing command for attribute 19 removes the parentheses from the comment, and places a comma before the comment.

Indefinite values

In numeric attributes, '~' will indicate an indefinitely small or indefinitely large number.

Examples:

~-5 - up to 5
5-~ - 5 or more

Specific, settable numbers will be substituted for the indefinite numbers where necessary (for example, for calculating a mean).

Delimiters in attributes

The following restrictions will apply to the positioning of the delimiters ',', '/', '&', and '-' in attributes.

The delimiters are not allowed in text attributes.
',' must precede the other delimiters.
The delimiters must not be adjacent, or be separated only by comments.
'&' must not be the next delimiter after '-', and vice versa.

Position and scope of comments in attributes

Comments will be able to be positioned anywhere in attributes, except at the start (i.e. not before the character number).

This added flexibility will worsen the ambiguity in determining the state value(s) with which the comment is associated. The resulting ambiguity in generated natural-language descriptions is no worse than in descriptions written directly in

natural language, and can presumably be tolerated. However, ambiguity will not be acceptable in the interpretation of some kinds of coded comments, particularly probabilities.

In mathematical notation, similar problems are usually solved by the use of brackets and rules of precedence.

Examples:

5,1&2/3<..> - comment applies to '3'.
5,1&2<..>/3 or 5,{1&2}<..>/3 - comment applies to '1&2'.
5,1&{2}<..>/3 - comment applies to '2'.
5,{1&2/3}<..> - comment applies to the whole attribute.

It may be possible to avoid the use of brackets, while obtaining sufficient functionality, by defining the scope of each type of comment in terms of the delimiters in an attribute.

The scope of ordinary comments and 'list' coded comments (such as references) can be undefined, because only natural-language descriptions are affected.

It would not normally be meaningful to associate probabilities with individual state values connected by '&' or '-', and it would be undesirable (though perhaps sometimes convenient through lack of information) to associate a single probability with two or more state values separated by '/'. Thus, it might be satisfactory to define the scope of probability comments to extend as far as the delimiter '/'.

Examples:

5,1&2<@prob .2>/3 - probability applies to '1&2'.
5,1&2/3<@80%> - probability applies to '3'.

The coded comments @about and @? would apply to the adjacent value.

The coded comments @reliability, @use, @edit, @up, and @down would apply to the whole attribute.

Dependent states

There will be a directive to indicate, for a set of list characters using the same list, that states of some characters may or may not be coded depending on whether the same state of another character is coded.

Example:

. . .
#40. infects/ hosts/
#41. does not infect/ hosts/
#42. <symptoms> chlorosis/ hosts/
#43. <symptoms> local lesions/ hosts/
#44. <symptoms> leaf curling/ hosts/
. . .
*APPLICABLE STATES 40,42-44
*INAPPLICABLE STATES 40,41 41,40:42-44

Mechanisms will be provided for producing appropriate natural-language descriptions, so that, for example, '40,35/77 41,52 42,35 43,35' might produce 'infects X (symptoms: chlorosis, local lesions), Y; does not infect Z.

Natural-language descriptions - insertion of parentheses

When incorporating attribute comments in natural-language descriptions, parentheses will be added if and only if the comment *follows* a state value. For example, '30,1/<occasionally>2<C. incompletus>' might produce 'Rachilla prolonged beyond the uppermost female-fertile floret, or occasionally terminated by a female-fertile floret (*C. incompletus*)'.

Natural-language descriptions - punctuation

The LINK CHARACTERS and REPLACE SEMICOLON BY COMMA directives will be replaced by three punctuation directives:

* PUNCTUATE ;. s_1 s_2 . . .
* PUNCTUATE ,. s_1 s_2 . . .
* PUNCTUATE ,; s_1 s_2 . . .

where s_i is a set of characters of the form

c_1:c_2: . . .

where c_j is a character number or range of numbers. The ';.' sets and the ',.' sets, taken together, must be mutually exclusive. The ',;' sets must be mutually exclusive, and each must be contained in one of the ';.' or ',.' sets.

The first punctuation mark in each directive is the 'internal' punctuation mark for each of the sets in the directive, and the second is the 'terminal' punctuation mark. Each character not mentioned in one of these directives is the sole member of a set with terminal punctuation mark '.'.

A description is divided into 'sub-sentences', which are the maximal sets of contiguous attributes in the description, such that each set is fully contained within all the punctuation sets to which any of its attributes belong. That is, an attribute terminates a sub-sentence if it belongs to a punctuation set to which the next attribute does not belong.

A sub-sentence is followed by the terminal punctuation mark of one of the punctuation sets in which it is contained; '.' takes precedence over ';'. Each attribute except the last in a sub-sentence is followed by the internal punctuation mark of one of the punctuation sets in which the sub-sentence is contained; ',' takes precedence over ';'.

Examples:

Each set of directives is followed by schematic representations of natural-language descriptions, in which each natural-language attribute is represented by its corresponding character number.

*PUNCTUATE ;. 13-15
*PUNCTUATE ,. 4-6 7-9
1. 2. 3. 4, 5, 6. 7, 8, 9. 10. 11. 12. 13; 14; 15.
2. 5. 8. 11. 14.
*PUNCTUATE ,. 1-9 10-12:16-18 13-15
*PUNCTUATE ,; 1-3 4-6 7-9
1, 2, 3; 4, 5, 6; 7, 8, 9. 10, 11, 12. 13, 14, 15. 16, 17, 18.
2; 4, 5. 10, 11, 16, 17.
*PUNCTUATE ;. 1-9 10-12:16-18 13-15
*PUNCTUATE ,; 4-6
1; 2; 3; 4, 5, 6; 7; 8; 9. 10; 11; 12. 13; 14; 15. 16; 17; 18.
2; 3; 5, 6. 10; 12; 17.

The punctuation generated by the PUNCTUATE directives will be able to be overridden by punctuation in comments in attributes.

Example:

12,1<,>/2<.>

Natural-language descriptions - omission of words

Words at the start of the feature descriptions of the second and subsequent attributes in a sub-sentence (see above) will be omitted if they are the same as words at the start of the first feature description which could have appeared in the sentence.

Within a variable attribute, repeated words at the start or end of each state description will be omitted provided that the context is sufficiently simple. Detailed rules have not yet been worked out.

Provision may need to be made to mark some words (for example, articles and adjectives which precede different nouns) as not subject to omission by the above rules.

Caution is necessary in proposing rules for the omission of words or punctuation marks. It is easy to think of circumstances where the application of simple rules will produce improvements; it is harder to think of all the circumstances where those rules will produce unacceptable results. For example, consider the character '#1. leaves/ 1. with long hairs/ 2. with short hairs/ 3. without hairs/', and the attribute '1,1/2'. This currently produces the description 'leaves with long hairs, or with short hairs'. Omitting the repeated words and the comma gives 'leaves with long or short hairs', which is clearly preferable. However, applying that procedure to the attribute '1,1&2/2/3' gives 'leaves with long and short or short or without hairs'. The presence of comments would complicate matters further, and languages other than English need to be considered. Any general reduction in clumsiness of expression should not result in the introduction, even rarely, of wording which is ambiguous, misleading, or nonsensical.

The following example (provided by E. J. Gouda) leads to difficulties even with a simple attribute. When simple omission of repeated words is used with the character '#6. plant with/ 1. a few leaves/ 2. many leaves/ 3. very many leaves/' and the attribute '6,2/3', the resulting natural-language description is 'plant with or very many leaves'. The word 'many' needs to be flagged as not subject to omission. A simpler, and quite common, example is states of the form 'x' and 'not x'.

Natural-language descriptions - editing

It will be possible to edit natural-language descriptions during their creation. Specified TED editing commands will be applied to attributes after they are generated, but before output. A directive will specify the commands to be applied for every attribute generated from a given character, and a coded comment @edit will allow editing of an individual attribute.

Example:

* EDIT CHARACTERS #38. d;1 nerved;i;with a single nerve;

SQL interface

To be designed by WA Herbarium (see a future issue - Editors).

Other Proposals

The proposals under this heading have been tentatively rejected by the development team, or require more detail before they can be properly assessed.

Extreme values for non-numeric characters

Extreme values should be allowed for all character types and delimiters, e.g. 20,1(/2).

This may have no advantage over the coded comments <@rarely> or <@probability x>. For numeric characters, the parentheses denoting the extreme values can be taken through unchanged to the natural-language descriptions, but this is probably unsuitable for other character types.

State prefixes and suffixes

The state descriptions in a multistate character should have provision for a prefix and suffix, which would be output only once per attribute in a natural-language description.

Example:

#6. leaves/ with/
1. sparse/
2. dense/ hairs/

The attribute 6,1/2 would produce the natural-language description ‘leaves with sparse or dense hairs’.

This proposal does not cope with the situation where only some of the states have the same initial (or final) words.

Example:

#6. leaves/
1. with sparse hairs/
2. with dense hairs/
3. without hairs/

The attribute 6,1/2 should still produce ‘leaves with sparse or dense hairs’.

Advantages of using prefixes and suffixes are: (1) avoidance of problems with repeated words which must not be omitted; (2) potential for placement of comments before or after the suffix. E. J. Gouda (Jungfrau 107, 3524 WJ Utrecht, The Netherlands) has implemented suffixes in his DELTA programs, and gives the following example, which illustrates the placement of attribute comments before and after the suffix. (His version of the character-list syntax includes the character type, and places the state suffix at the end of the feature description.)

Example:

#3. RN. plant <length, height>/ cm tall/
#6. OM. plant with <density of the leaves>/ leaves/
1. a few/
2. many/
3. very many/

#7. UM. plant forming <form of the rosette>/ rosette/
1. a tubular/
2. a narrowly funnelform/
3. a broadly funnelform/
4. a subbulbous/

The DELTA description

3,<(15-)>35-60 6,1/<rarely>2 7,1/<sometimes>2/<rarely>4<(inflated sheaths)><(often tinged with red)>

produces the natural-language description

Plant (15-)35-60 cm tall, with a few or rarely many leaves, forming a tubular or sometimes a narrowly funnelform or rarely a subbulbous (inflated sheaths) rosette (often tinged with red).

There are two comments at the end of the last attribute, and the suffix is positioned between them. This mechanism would not be possible in the proposed new DELTA format, in which there is no limit to the number of comments which may appear at any position.

Submission of Proposals

Proposals for the new DELTA system, and criticism of existing proposals, are welcome, and should be submitted to the development team or to the editors of the DELTA Newsletter (Western Australian Herbarium, PO Box 104, Como WA 6152, Australia). Please make the proposals as specific and detailed as possible, and discuss possible disadvantages and difficulties as well as advantages.

References

Dallwitz, M. J., Paine, T. A., and Zurcher, E. J. (1993). 'User's Guide to the DELTA System: a General System for Processing Taxonomic Descriptions.' 4th edition. (CSIRO Division of Entomology: Canberra.)

Δ Δ Δ Δ

COMMENTS ON NEW FEATURES FOR THE DELTA SYSTEM

R.J. Pankhurst

Royal Botanic Garden, Inverlieth Row, Edinburgh EH3 5LR UK
Phone +31 552 7171 Fax +31 552 0382 Internet rjp@castle.ed.ac.uk

Complexity

The present proposals are for adding new features to DELTA. I am fond of telling audiences that the current DELTA format is straightforward and easy to grasp, and that it contains the very minimum of features which are needed to describe real life variation in plants and animals. In spite of this, there is a steady grumble from those who could or should be using DELTA that it is 'too complicated'. Some of these folk are just saying that they would like better interfaces. Well of course, and we are working on it, but don't hold your breath. However, some people really are saying that they think plant and animal variation could or should be represented more simply, and are acting accordingly. I know of several expert identification programs of which the authors have claimed that their work was necessary in order to 'do it better than the DELTA software does'. I also know of 4 identification projects started within the last year where DELTA software was considered and rejected for similar reasons.

So now we are considering adding more complexity to DELTA. I am personally in favour of some of the proposals, as below. But pause and reflect - is this really the right way to go?

From the way in which some existing DELTA features are used e.g. text characters, and several of the new proposals, it seems that DELTA users are already feeling their way towards a database, rather than a data format. Witness the inclusion of data on nomenclature, distribution, geography, bibliography etc which has nothing to do with morphology. There is a mention of an SQL interface. SQL is for interfacing to databases, but in the case of DELTA, what database? The only database systems of which I am aware which can contain DELTA data are PANDORA and ALICE (to a lesser degree). Is anyone proposing anything else? If so, it has to be sophisticated, and XBASE is inadequate. So, should we not be planning the DELTA database, instead of DELTA format?

General

There are several features of DELTA as it now stands which I think need criticising.

1) The custom of splitting up files into SPECS, CHARS and ITEMS.

I recall that the original reason for doing this was to allow for alternative character lists with different wording, but it seems that PANKEY users have rarely if ever found this necessary. We just use another file with different data in it.

PROPOSAL: Do away with having 3 separate files and combine them into one.

2) '&' between States:

This was originally introduced as a shorthand for description printing only. It is not used in any other context, and cannot be, since character states have to be mutually exclusive. The DELTA editor flags states with & and converts them into '/'.

eg.#1. petal <colour>/
1. white/
2. pink/

then 1&2 means 'white and pink'. This is different from either (completely) white or (completely) pink, and is really a third state. A key would have to say 'petals white', or 'petals pink' or petals 'white and pink' as they are all different. So what DEDIT does is to convert 1&2 into 1/2 with a warning. It would be better to have DEDIT optionally create a third state.

PROPOSAL: Abolish & between states and provide a utility to convert to additional states.

3) Convention on state ordering.

This is not 'official' DELTA but merely a matter of convention. I always advise PANKEY users to put states either all in ascending or descending order, with ASCENDING order preferred. Is it not more natural to put small before large, and absence before presence? I find the existing convention to be the wrong way round.

PROPOSAL: Recommend that states of ordered characters are put in ASCENDING order, not descending.

4) Character types EUM and EOM.

As far as I know, no PANKEY user has ever used these.

PROPOSAL: Abolish EUM and EOM.

In Favour of

DEFINE DELIMITERS. Might not use it myself, but think it's a good idea.

Unrestricted line length.

Mnemonics for characters in addition to character numbers. Fine, how about mnemonics for states too?

Comments linked to states in ITEMS. Fine.

Expressions of probability linked to states. Fine, Lebbe has done some work on this in Paris.

Not in Favour of

Synonyms. Should not be in DELTA format at all, but in a database. See comments above.

Dependent states for list characters. Is this different in some way from DEPENDENT CHARACTERS? Or have I not understood?

PUNCTUATE. This is needed but seems unnecessarily complicated. For the PANKEY

DESCRIP program (which has been in use for some time) there is a directive SENTENCE ORDER, e.g.

5-7,3.4 10,12,11 etc.

which means to put 3 to 7 in a paragraph with commas between 5,6 and 6,7 and 7,3 with full stop between 3.4 (and at the end). Is this not an easier way to do it?

SQL. This has been available to us in the programming of PANDORA and other database systems with Advanced Revelation, but have never found a use for it.

The AREV R/LIST language (same as PICK ACCESS) is better. SQL is restrictive, with fixed length fields of only a few data types.

Alternative languages in character lists. I don't see any need for that. Why not just have another data file with a translated character list in it?

Other Suggestions

How about a means to express DELTA directives in other languages? I don't think that this was mentioned.

I am uncertain about the value of having extra data types; not that those proposed are not sensible, but there are many more data types which could be suggested. Lebbe lists 4 of them in his thesis, and gives a reference to more.

Lebbe suggests a graph of relations (mostly hierarchical) between characters e.g. plant includes stem, leaves, flowers; inflorescence include leaves, flowers and fruit etc. Interactive programs could make use of such knowledge.

At present there is only one level of character grouping.

Lebbe has proposed ways of calculating with probabilities on states, both qualitative and numerical.

Questions

Although I would reckon myself `familiar with the DELTA format' as required, there are several things mentioned which I have not heard of before. Please, therefore, what are: ENTITEM; DELFOR; TED (some kind of cryptic text editor?); `the inner comment mechanism'; LINK CHARACTERS directive; REPLACE SEMICOLON BY COMMA directive?

Δ

Reply to Richard Pankhurst's Comments on 'Preliminary Suggestions for New Features'

M.J. Dallwitz

CSIRO Division of Entomology, GPO Box 1700, Canberra ACT 2601, Australia
Phone +61 6 246 4075 Fax +61 6 246 4000 Internet md@ento.csiro.au

The DELTA proposals presented here were written primarily to assist in designing new programs for the DELTA System of the CSIRO Division of Entomology. Many of the proposed new features of the programs require new kinds of data, so it is also proposed to enhance the DELTA data-interchange format accordingly. The distinction between the programs and the DELTA format is made in the first paragraph of the proposals printed here, but this was added since the draft of 2 September 1993, upon which Richard commented.

Complexity

The data that taxonomists use, and the operations they carry out on those data, are complex. If these data are to be represented and manipulated in computers, the data formats and programs must necessarily be complex too. The DELTA programs written in the CSIRO Division of Entomology are certainly quite complex. We have run many training courses in the use of the programs, and we find that 4 or 5 days of intensive training and practice are necessary to obtain a thorough grounding, and an appreciation of the power of the programs. (The cumbersome user interfaces of some of the older programs present an initial hurdle, but this is not a large part of the total learning effort. The improved interfaces being planned will be less daunting for beginners, and will improve the efficiency for experts, but will not greatly reduce the amount of time required to learn the system.) No participant at these courses has complained to me, at the end of the course, that the system is unnecessarily complex (perhaps they are too polite); but many have said that they have been impressed by the capabilities of the system, and several have later told me that they have continued to discover other valuable uses for the programs.

The enhancements in the proposed new DELTA format will certainly increase its complexity, but are necessary to provide facilities requested by users. The new user interface in CONFOR will try to make the rarely used features unobtrusive. Programs reading or writing the new format will easily be able to ignore most of the new features if they do not require them - for example, many of the enhancements to attributes can be treated as comments. Also, CONFOR will be able to output simplified versions of the data - for example, a character list in which only one of the alternative wordings is included), and an application programming interface (API) to the database system will be provided.

The proposed new version of CONFOR will, indeed, be a database system. The new DELTA format will serve for data transfer between different programs, and between versions of CONFOR running on different kinds of computer hardware. The format will normally be read and written only by computers, although the fact that it will be printable on paper and readable (with some difficulty) by people will be a useful insurance against data loss through the obsolescence of computer hardware and software. I don't agree that the DELTA format and programs should be restricted to 'morphology' (presumably including anatomy, chemistry, etc.) Other kinds of information are required in descriptions and for information retrieval, and may also be useful for identification (e.g. distribution and classification). The SQL interface would enable some of these other kinds of information to be brought into DELTA databases from, for example, specimen, nomenclatural, and geographical databases. It would not be a part of the DELTA format.

General

Multiple files. The splitting up of data and directives into several files is convenient for several reasons, but is not an essential part of the current CONFOR, which is quite capable of running from a single file (see Section 3.2, Data Organization and File Names, in the DELTA User's Guide). As the new CONFOR will be a database system, the question will not normally concern users, but the current flexibility will be retained for input and output of DELTA format for data exchange. Thus, data exchange with a program which reads and/or writes a single data file will be able to be accommodated.

'&' in attributes. I think there would be an outcry from users if '&' between state values were abolished. It would be difficult to generate new state descriptions automatically, as the old states usually need to be modified

too (perhaps by adding the word 'only'). It can also be undesirable for some purposes, particularly classification, to have the extra states, whether automatically or manually generated. Manual generation of extra states always remains a possibility, and application programs are at liberty to treat '&' as '/'. Another possibility is to break up the character. For example, the character

#1. hairs/
1. long/
2. short/

could be replaced by the two characters

#1. long hairs/
1. present/
2. absent/

and

#2. short hairs/
1. present/
2. absent/

The latter formulation would have advantages for identification, and possibly for classification, but would give clumsier natural-language descriptions, e.g. 'long hairs present, short hairs present' instead of 'hairs long and short', and 'long hairs present or absent, short hairs present or absent' instead of 'hairs long or short'.

State orders. The ordering of states is a matter of user preference, and does not affect the DELTA format or programs in any way.

Character types EUM and EOM. I agree that these types should be abolished. The distinction was needed for a classification program which is no longer in use. If required again, it would be better implemented as an ad-hoc directive.

Character identifiers. I agree that identifiers or mnemonics for character states would be useful. It might be better to place the identifier after the number, e.g. '#45<awn01>. awns <presence>/ 1<p>. present/ 2<a>. absent/'.

Synonomy. Many applications need at least a simple treatment of synonymy, to gain access to information when the correct name is not known. Also, the synonymy is often required in natural-language descriptions.

Relations (dependencies) between indexed lists. The relations between indexed lists will provide a less cumbersome way of expressing some dependencies, particularly those for distributions and classifications. If these dependencies are expressed in the current way, the number of characters required rises exponentially with the number of dependency levels. For example, the single 'regions' character in the example in the proposals would need to be replaced by the nine characters 'regions of Europe', 'regions of Africa', ... 'regions of the Antarctic', and more than 50 characters would be required for the TDWG Level 3 units.

Punctuation. I would certainly like to simplify the PUNCTUATE directive (which would not be part of the DELTA format). However, it is necessary to include provision for semicolons, and to give appropriate results when characters are missing (through being uncoded, inapplicable, or excluded) as shown in the examples.

Alternative character lists. There are two separate proposals here: 'alternative wordings' within a character list, for use where occasional wording differences are required for different purposes; and self-contained alternative character lists for use with separate languages, as in the current system, but with an added identifier so that the list can be easily invoked as required.

Alternative languages for DELTA directives. The interactive commands in the new CONFOR will be easily expressible in other languages, as in the current INTKEY. (However, translators may prefer not to do so, as in the French INTKEY translation by Pierre-André Loizeau.) Provision for alternative languages for the DELTA format itself would make programming for data exchange more difficult, and is probably unnecessary, as the format would not usually be read by people.

Extra data types. I am not against adding further data types if specific proposals are made, and there is support for them from users.

Questions. ENTITEM, DELFOR, and TED are programs supplied with the CSIRO DELTA package, and were mentioned only incidentally in the proposals. TED is indeed an editor: the @edit feature needs a syntax to describe the required editing, and a cut-down version of the TED syntax is probably as good as any for this purpose. The LINK CHARACTERS and REPLACE SEMICOLON BY COMMA directives are CONFOR directives for controlling natural-language output (not part of the DELTA format), and, as stated, would be replaced by the more general PUNCTUATE directive. 'Inner' comments are nested comments in attributes. Earlier editions of the DELTA User's Guide did not state whether or not these were allowed. This ambiguity had to be removed, and I chose to allow nested comments, as the programming involved is little more difficult than disallowing them (they still have to be detected). As an interim facility for CONFOR (not part of the DELTA format), I added a directive OMIT INNER COMMENTS to allow their omission from natural-language descriptions.

Δ

EDITED CORRESPONDENCE ON DELTA ENHANCEMENT PROPOSALS

Joseph H. Kirkbride Jr.* and M.J. Dallwitz

*Systematic Botany and Mycology Laboratory, Dept. of Agriculture, Beltsville Maryland, USA
Internet jkirkbride@asrr.arsusda.gov

From J.H. Kirkbride, to Mike Dallwitz, 23 September 1993

Dear Mike:

I would like to return to my 'favorite' subject, character prefixes and suffixes or omission of words in natural language outputs, i.e., descriptions and keys. I still favor suffixes and prefixes over omission of words because it is much easier to deal with the explicit than the implicit, especially in a large character list, i.e., 200 or 300 characters. When you have to put in the prefix and suffix, it is obvious exactly what will happen on output in each character. With omission of words, characters and their states must be very carefully compared to understand what will happen on output. The unknown which has to be logically worked out with precision of language is always much more difficult than the explicit.

I like your proposal for a LIST character very much; it will deal with a number of things that are now difficult or impossible to incorporate into DELTA format. In your examples for "Coded `comments' (subsidiary information)" there is 13,22/ <occassionally @5% @ref 54>3. I am assuming that `@ref 54' refers to an entry 54 in a LIST character. Is that correct?

From Mike Dallwitz, to J.H. Kirkbride, 24 September 1993

Dear Joe:

Could I have some more information about your ideas for omission of words in natural-language descriptions? There was nothing in the draft proposal about CHARACTER prefixes. If you are in favour of them, please provide a detailed proposal, including syntax, advantages, and disadvantages. The size of the character list is not directly relevant, as the omission only takes place within 'subsentences'. Are you in favour of the STATE prefixes and suffixes (which are used only within an attribute)? If so, what are your comments on my objections to their use?

'@ref' refers to an 'indexed list' of references, such as the one given in the example. These lists are not identical with characters. A list might be used in any number of characters, or in none at all.

From J.H. Kirkbride, to Mike Dallwitz, 29 September 1993

Dear Mike:

I think that my expression 'character prefixes and suffixes' is causing confusion. I will attempt to explain at length what I was referring to. In my *Cucumis* character list, I have the following characters:

#30. petioles <overall shape>/
1. cylindrical <the sides parallel from base to apex IMPLICIT>/
2. claviform <club shaped, flaring upwards from a narrower base to a wider apex>/

#31. petioles <shape in cross section>/
1. sulcate in cross section <alternating longitudinal grooves and ridges in cross section>/
2. terete in cross section <smooth circular in outline, cylindric>/

#32. petioles <length in cm RN>/
cm long/

#33. petioles <aculeate or not>/
1. not aculeate/
2. aculeate/

#34. petioles <pubescence type uniform or 2/3 types on each petiole>/
1. pubescence a single type on each petiole <IMPLICIT>/
2. pubescence 2 different types uniformly intermixed on each petiole/
3. pubescence 3 different types in distinct zones on each petiole/

#35. petioles <pubescence type>/
1. glabrous/
2. hispid/
3. hispidulous/
4. villous/
5. pilose/
6. lanate/
7. hirsute/
8. antrorse-strigose/
9. retrorse-strigose/
10. setose/
11. scabrous/

#36. petioles <individual hair type>/
1. with nonbreakaway hairs <IMPLICIT>/
2. with breakaway hairs <nonglandular, multicellular, conical hairs consisting of an easily ruptured, multicellular foot with numerous, small, thin-walled cells and a uniseriate body with larger, thick-walled cells (Inamdar & Gangadhara, 1975; Inamdar et al., 1990)>/

I want all of these characters to appear together as a single sentence with the characters separated by ';', especially character 36. I could have a second sentence for character 36, but I prefer to have it unquestionably linked to 'petioles'. Therefore, you could get:

30,1 31,1 32,0.5-1.5 33,1, 34,2 35,2&3/6&2 36,1/2

This would appear as:

"Petioles cylindrical; sulcate in cross section; 0.5-1.5 cm long; not aculeate; pubescence 2 different types uniformly intermixed on each petiole; hispid and hispidulous or lanate and hispid; with nonbreakaway hairs or with breakaway hairs".

I would prefer to see characters 31 and 36 as:

#31. petioles <shape in cross section>/ * in cross section/
1. sulcate <alternating longitudinal grooves and ridges in cross section>/
2. terete <smooth circular in outline, cylindric>/

#36. petioles <individual hair type>/ with * hairs/
1. nonbreakaway <IMPLICIT>/
2. breakaway <nonglandular, multicellular, conical hairs consisting of an easily ruptured, multicellular foot with numerous, small, thin-walled cells and a uniseriate body with larger, thick-walled cells (Inamdar & Gangadhara, 1975; Inamdar et al., 1990)>/

After the feature description, there is a second phrase with an asterisk (*). Those words preceding the asterisk are the "character prefix", and those words following the asterisk and preceding the slash (/) are the "character suffix". The output would then appear as:

> "Petioles cylindrical; sulcate in cross section; 0.5-1.5 cm long; not aculeate; pubescence 2 different types uniformly intermixed on each petiole; hispid and hispidulous or lanate and hispid; with nonbreakaway or breakaway hairs".

This is almost identical to the proposal made by E.J. Gouda with the addition of prefixes and the asterisk as the separator and indicator. I prefer this because I feel that it is easier to use and understand than the 'omission' mechanism.

I would like to propose another feature for inclusion, the formula. In most botanical literature, two-dimensional, and sometimes even three-dimensional, organs have their size expressed as a formula. For example, 'leaf blade 10-12 cm long, 4-6 cm wide' is now usually expressed as 'leaf blade 10-12 X 4-6 cm'. This saves 13 characters in this expression, and in a large work can reduce a text by several pages. I would like a directive such as *FORMULA CHARACTERS which would work just like LINK CHARACTERS.

In my *Cucumis* database, I have the following characters:

> #44. leaf blades <overall length in cm RN>/
> X/
>
> #45. leaf blades <overall width in cm RN>/
> cm/

Therefore, you could get:

> 44,(2-)4-6 45,(1-)2-3(-4)

This would appear as 'leaf blades (2-)4-6 X; (1-)2-3(-4) cm;'. I then use a word processor to globally change 'X;' to 'X' to end up with 'leaf blades (2-)4-6 X (1-)2-3(-4) cm;'. I would prefer to have the directive *FORMULA CHARACTERS 44-45 result in the same thing. An additional directive would be necessary for the formula connector. 'X' would be a fine default value, but for some manuscripts submitted as electronic files it might be necessary to have the separator as an easily replaceable string of characters. The additional directive would allow the user to designate the formula separator.

There is some inherent danger in this type of directive, but it could be reduced by allowing formulae only for numeric characters that are consecutive. Also, at least three or more characters would have to be allowed in a formula for three dimensional organs. There would still be the danger of mixing numerical units, but that would be the responsibility of the user.

What has happened to me is that for some reason the second character of a formula is not scored, and the output ends up as 'leaf blades (2-)4-6 X hispid on the upper surface;'. CONFOR needs to able to deal with formula characters not scored or absent for some reason, so that the output is 'leaf blades (2-)4-6 cm long; hispid on the upper surface;'.

> **From Mike Dallwitz, to J.H. Kirkbride, 30 September 1993**
> Dear Joe:

Thanks for your further comments on the DELTA proposals. Your suggestion seems to be identical in principle to the one I described under 'state prefixes and suffixes' (only the syntax is different). I used the word 'state' rather than 'character' because the prefix occurs before the states, not before the character as a whole. I understand your point of view that it is easier to think about the results when the prefix and suffix are specified explicitly, and I will put this to the workshop. It could also be argued that automatic omission of words would usually produce the desired result, and it would be easier to take explicit action only if this fails. However, you still have not addressed what I consider to be the crucial argument against explicit prefixes and suffixes: it cannot cope with the situation where only some of the states have the potentially redundant words (e.g. with ..., with ..., without ..., as in the example in the proposal). Incidentally, in character 36 in your example, even the current CONFOR will produce slightly improved wording if you move the word

'with' to the feature line.

I will add something about formulas to the proposal. There needs to be provision for specifying different 'units' for complete formulas and for other contexts (incomplete formulas, keys, INTKEY).

From J.H. Kirkbride, to Mike Dallwitz, 30 September 1993

Dear Mike:

The particular example that you are using to discuss the question of prefixes and suffixes for the states of a character is not a good example. The character violates one of Robert [Webster's] basic 'rules', do not combine two characters together. Your example is:

#6. leaves/
1. with sparse hairs/
2. with dense hairs/
3. without hairs/

This character, as presented by you rolls together characters that should be done separately: 1) with or without hairs (present or absent); and, 2) hair density when hairs are present. This 'character' should be:

#6. leaves <presence or absence of pubescence>/
1. pubescent <with hairs>/
2. glabrous <without hairs>/

#7. leaves <density of pubescence when present>/
1. with sparse hairs when pubescent/
2. with dense hairs when pubescent/

Proper formulation of the characters eliminates the problem of 'with' and 'without'. Given the propensity for pubescence to vary in density, the prefix-suffix difficulty still remains in my character #7.

Strict adherence to your punctuation rules (and I do adhere to them strictly) is what leads to this problem. People do not like to see: Leaves pubescent or glabrous; with sparse or dense hairs when pubescent. It is not aesthetically pleasing. So they roll them together and get: Leaves with sparse or dense hairs or without hairs. This 'solution' is WRONG in my opinion because you can end up contrasting hair density against lack of hairs in a key or INTKEY. Many problems are generated by poor formulation of characters. Most taxonomists look at a leaf and encompass a hundred characters without clearly distinguishing each one, and they try to formulate their DELTA character list in the same way.

You are right about the 'unit' situation with formulae. The lack of data for one part of a formula causes this: both leaf length and width scored: leaves 2-7 X 0.5-1.5 cm only leaf length scored: leaves 2-7 cm long only leaf width scored: leaves 0.5-1.5 cm wide

There must be two 'units' for each part of the formula: 'X' or 'cm long' for leaf length; 'cm' or 'cm wide' for leaf width. In keys and INTKEY, the nonformula 'unit' should appear, i.e., 'cm long' and 'cm wide'.

From Mike Dallwitz, to J.H. Kirkbride, 1 October 1993

Dear Joe:

I don't agree that the formulation of the leaf pubescence character is bad, as the states form a natural continuum, and might be under the control of a simple genetic mechanism. There is certainly no difficulty in making the contrasts for identification, and it might well be the most appropriate for classification too. These decisions should be left to the judgement of the taxonomist, and we don't want to place unnecessary obstacles in the way of some formulations.

Some questions and notes on the New Features document (9 February 1994)

Dr E. J. Gouda

Jungfrau 107NL - 3524 WJ Utrecht Netherlands
Internet gouda@runner.knoware.nl

Numbered lists. What is the advantage of this in comparison to a multistate character?

Related lists. How would these be used?

Alternative wordings for characters. What's the reason for combining dependent characters this way? The eg. #38.0 combined with #38.1 doesn't make sense to me.

Indefinite values. Is 2,~5 the same as 2,-5 and will the latter be invalid? What about 2,5~ and 2,5-?

Natural-language descriptions - insertion of parentheses. Is it not better to handle preceding and following comment the same way? Then you can choose to replace angle brackets by parentheses or avoid parentheses and include them where necessary, e.g. 30,1/<occasionally>2<(C. incompletus)>. This will make the generation of natural language descriptions more flexible.

State prefixes and suffixes.

Wouldn't it be more consistent to use a general format for character descriptions (any type)

> #char_number. feature/ [suffix/] [1. state_1/ [2. state2/ [..]]]

where parts between '[]' are optional (numeric and text characters have no states), and where a comment can precede or follow the feature, suffix, and/or states? I think that the use of prefixes is not necessary when you omit the feature start words, which can be done easily.

> #6. leaves/ with/ 1. sparse/ 2. dense/ hairs/

is the same as

> #6. leaves with/ 1. sparse/ 2. dense/ hairs/

It can be very dangerous to omit duplicated words from state descriptions, that is the reason why I like to see the suffixes. It does not matter to me where they are, preceding or following the states. This will give the user more control.

The general form of an attribute could be:

char_number,[extreme_1*]value_1[*value_2[..]][*extreme_2]

where '*' is a separator (-/&), with optional comments that can only precede or follow values (so no comment preceding or following the character number). The last state could be followed by two comments, the first to be placed before the suffix, and the second after. Example:

> 12,<rarely>4-<mostly about>8<><at full grown spike>

In this example, there is to be no comment before the suffix, so the place is occupied by an empty comment '<>'. For this mechanism to work, it would be necessary to forbid multiple comments in general. For example, there seems to be no advantage in

> 30,4<often green><at least toward the tip><soon glabrous>

compared with

> 30,4<often green, at least toward the tip, soon glabrous>

The DELTA standard.

There is some thing about the DELTA format definitions (as a standard) that is not right. I think that a standard must be defined independent of the possible directives needed for an application. This is because many directives are related to one application only. Do you think it would be possible to separate the application linked directives from the DELTA format definitions? It would be nice to have standard test files for testing applications against the DELTA format definitions (all possibilities included).

I think the best thing to do is to keep the DELTA format as simple and clean as possible. Soon it will be only usable to computer specialists.

Reply to Eric Gouda's Comments on 'Preliminary Suggestions for New Features for the DELTA System'

M. J. Dallwitz

Numbered lists. A single numbered list could be used in several characters, ensuring that these are consistent. For example, a list of geographic regions could be used in characters such as 'native to', 'naturalized in', 'now extinct in' (there is another example in the proposals, under 'dependent states'). The 'list' characters would support the relations defined in RELATED LISTS and APPLICABLE STATES. Also, a list could be used for comments in characters of any type (e.g. references), as well as in special 'list' characters.

Related lists. The use of related lists could simplify data coding and improve natural-language descriptions and interactive identification. For an example, see Leslie Watson's Grass Genera or Angiosperm Families data. The present system, using character dependencies, results in an unavoidably clumsy representation of floristic kingdoms, subkingdoms, and regions.

Alternative wordings for characters. This has nothing to do with dependencies or combining characters. It is simply to provide alternative wordings for those characters that sometimes need it. In the present system, it is coped with by using alternative versions of the character list, and/or using the KEY CHARACTER LIST directive. Often, the wordings for most of the characters in the alternative lists are the same, which wastes space and makes it more difficult to maintain the lists.

Indefinite values. '2,-5' means that character 2 has the value 'minus 5', so can't be used to represent an indefinite value.

Natural-language descriptions - insertion of parentheses. It would be more consistent and flexible for the program to treat all comments the same, and not to insert parentheses. However, parentheses are usually necessary when the comment follows the state, and the present CONFOR always adds them. I will make it optional in the new CONFOR.

State prefixes and suffixes. There are three aspects to the use of state prefixes and suffixes.

(1) Suffixes can reduce the number of words in the character list, and make it easier to read, especially when the suffix is long (e.g. 'on the abaxial surface of the mature leaves').

(2) Prefixes and suffixes can be used to specify precisely which duplicated words can be omitted from natural-language descriptions. This argument applies equally to prefixes and suffixes: the dangers of omitting all duplicated words are similar in both cases. Perhaps a feature suffix is needed in addition to the state prefix. The former would restrict the omission of words between characters, and the latter within characters. I still think that provision needs to be made for the omission of words which cannot be placed in a prefix or suffix (because they are not the same in all character states). There would need to be a mechanism for specifying words which should not be omitted, but I haven't yet thought of a clean way of doing this.

(3) Prefixes and suffixes would give added flexibility in positioning comments, but would give rise to additional problems. The proposed use of an empty comment <> to force a comment to be placed after the suffix seems clumsy, particularly since this is the more common requirement. If the character had a suffix, almost every simple attribute like 12,1<comment> would need to become 12,1<><comment>. This would apply to most numeric characters, in which the'units' would be considered a suffix. Multiple comments of the type given in the example are certainly unnecessary, but 30,4<often green><@ref 134, 218><@use 2: 3/4> might seem more natural than 30,4<often green @ref 134, 218 @use 2: 3/4> I would appreciate more comments on these matters.

The DELTA standard. It is indeed necessary to distinguish between the directives belonging to the DELTA standard, and directives for particular applications. The principle for deciding into which category a directive falls is given in the first paragraph of the proposals. Eventually, I will draw up a definition of the standard. This will be fairly difficult, as there are borderline cases. For example, the method used for inserting parentheses in natural language descriptions (above) will determine how the data should be entered. The extensions to the DELTA standard were designed, in the light of experience, to cope with shortcomings of the current system. The programs associated with the format should be able to shield users from most of the complexity. For data interchange, programs will be able to ignore most of the features that they do not require (e.g. the 'coded comments'). Furthermore, CONFOR will have facilities for outputting simpler versions of the data (e.g. a version in a particular language).

About the DELTA Newsletter

A communications medium for botanical and zoological taxonomists interested in descriptive databases.

Topics

Computer programs for taxonomy
Data formats
Data interchange standards
Data capture
Data analysis
Database design
Description printing
Expert systems
Information retrieval
Interactive identification
Keymaking
Mapping systems
Taxonomic characters

Features

Articles
Program releases and updates
Spotlight on DELTA features
Technical tips
Database advertisements
Meeting notices and reports
Publication and database reviews and notices
Image issues
Letters
Personal profiles

Frequency
Twice yearly, April and October

Closing dates March & September

Submissions welcome.
Post, fax or email to:

The Editors
DELTA Newsletter
Western Australian Herbarium
PO Box 104
Como WA 6152
AUSTRALIA

Phone +61 9 334 0500
Fax +61 9 334 0515
Email (via M. Dallwitz) Internet md@ento.csiro.au

SUBSCRIBE TO THE DELTA NEWSLETTER

The DELTA Newsletter subscription list is being updated. **If you wish to receive future issues**, please complete the form below and return to the editors. A modest annual charge has been introduced to assist with distribution costs. The cost can be reduced for groups of readers by payment of one subscription (e.g. by your institutional library) and making photocopies as required. Exemptions from payment are possible for people in certain countries who cannot meet the fee. Please note that extra-Australian personal cheques cannot be dealt with; instead make International Money Orders payable to **The DELTA Newsletter.**

Please put me on the mailing list for future issues of the DELTA Newsletter.

Name _____

Address _____

_____

_____

Country _____

Email address _____

Payment (for 3 years) enclosed (tick applicable box).

	Overseas	Australia
Email	❑$US15.00	❑$A15.00
Air	❑$US45.00	❑$A30.00
Sea	❑$US30.00	❑Exemption requested

Send to:

The Editors
DELTA Newsletter
Western Australian Herbarium
PO Box 104
Como WA 6152
AUSTRALIA

Signed _____ Date _____