DELTA Newsletter

ISSN 1320-2979

Number 12, April 1996

CONTENTS

EDITORIAL

The Newsletter has reached an important stage of maturity in this issue with a fine array of articles from a range of contributors, only one of them coming from the key boards of the editorial group. This means two things. Firstly there is a rising tide of interest in the application of the DELTA approach to biological (and archaeological!) studies, and a keenness to use DELTA in leading-edge technology such as HTML and Java. Secondly, the pleasing number of contributions indicates that the Newsletter serves a useful purpose, and if the inflow of articles continues, the Newsletter will too.

Several of the articles discuss or offer solutions to the oft-mentioned need for specific aids to the entering and editing of DELTA data. Users ought therefore be able to find something to suit themselves from among the selection mentioned in this and previous issues. Different approaches have different strengths and weaknesses, and we plan to include a guide to entry/editing tools in a future issue.

Three of the articles, by Pankhurst, Chapman & Choo, and Allkin, mention the databasing of non-morphological or text-type data, either within the same program as the DELTA format data, or through links to other databases. Some aspects of this subject relate also to planned enhancements to the DELTA format (see Newsletter No. 9), where text comments are to be given tags.

The World Wide Web and its relation to DELTA and the identification/query program INTKEY is the subject of two articles. We can expect developments of the promises made in these articles in the future. Application programs which complement or are at least partly alternatives to the CSIRO programs are mentioned in several of the articles, indicating that the original idea of DELTA as a basic format that can be use by a range of programs, is becoming increasingly an established fact. This again augurs well for the future, provided that the current DELTA standard is supported. Another future article will examine this matter. Finally, to add value to the Newsletter, we plan to provide back issues on the WWW - look out for our announcement on DELTA-L or look for it soon on the DELTA home page.

Terry D. Macfarlane*, Alex R. Chapman, 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

Designing a Graphical User Interface for DELTA: Some considerations

Mauro J. Cavalcanti

Departamento de Biologia Geral, Universidade Santa Ursula,
Rua Fernando Ferrari, 75, Botafogo, 22231-040, Rio de Janeiro, RJ, Brasil
E-mail: maurobio@omega.lncc.br

Since its introduction almost two decades ago, the DELTA system has gained increasing acceptance by taxonomists all over the world, with several large databases in zoology, botany and virology already prepared plus many others under development. While the DELTA format provides an excellent way of describing taxonomic data — up to point of being adopted by the International Working Group on Taxonomic Databases for Plant Sciences (TDWG) as a standard for data encoding and exchange —the programs that comprise the DELTA package still reflect its origins in the mainframe world. Frequent criticisms have been the lack of an integrated, user-friendly interface for running the programs, the lack of an adequate data entry/editing system, and the unavailability of DELTA on computer platforms other than DOS/Windows PCs.

Over the last two years or so, I have been involved in developing user-friendly tools for DELTA — of which my DIANA shells (available for MS-DOS and Windows) were the first products — and I have dwelled deeply on the issues of user interface design, data models, search algorithms, data structures and the relevance of all this to the development of reliable biological software. Perhaps more importantly, I have also undertaken several experiments with user interfaces and data structures more adequate to the presentation and representation (sensu Diederich & Milton, 1993 — anyone seriously interested in developing GUI's for biological software would be well advised to read that paper) of biological data. Here I intend to present some of the preliminary, tentative conclusions I have reached in the course of these experiments, in the hope of obtaining some useful feedback from DELTA users and developers.

A key issue: portability

I start from the principle that a Graphical User Interface (GUI) is the most desirable kind of environment for a computer user, by making any system easier to grasp by the novice and faster to use by the more experienced user — the phenomenal success of the Microsoft Windows operating environment is an indication that this is the direction to which the software industry points.

GUI's are available on the three main computer platforms in use today — PCs, Macs, and UNIX workstations. The DELTA programs that take advantage of a GUI (ie. INTKEY and INTIMATE), however, are only available on DOS/ Windows PCs (or on other platforms under DOS emulation). So, perhaps more than offering an intelligent, editing tool (more on this below), the DELTA system should be as portable as possible, not only because of the large number of both PC and Mac users, but also to face the widespread (and increasing) availability of UNIX workstations. WWW browsers like Mosaic and Netscape offer a relevant analogy here, since they are both available on the three main platforms above mentioned.

However, developing a GUI-based program is not an easy task, even for experienced programmers — the difficulties of writing such programs are often underestimated. However, there are now software tools that not only allow simplification of the process of software development for each of the widely used GUI platforms: MS-Windows, Macintosh, UNIX/XWindows, but also make the programs portable between them.

One of these tools is VIBRANT, a high-level, multi-platform user interface development library written in C by Jonathan Kans (kans@ray.nlm.nih.gov), Information Engineering Branch, National Center for Biotechnology Information, NLM, NIH, Bethesda, USA. It is distributed as part of the NCBI Software Development Toolkit. VIBRANT acts as an intermediary between an application and the underlying windowing system toolkit. It is available for free and comes with full source codes. With it, a GUI-based application can be written that runs without modification on any of the mentioned platforms — you just have to re-compile them on the target machine, using the native C compiler.

Commercial software libraries based on the C language also exist that perform similar functions, such as Zinc and zApp, but these are usually very expensive and do not include source code.

Therefore, not only the DELTA programs themselves, but as a matter of fact, all DELTA "third-party add-ons" (like Lander's DMSWIN, Gouda's TAXASOFT, or my own DIANA software shell for DELTA) should be greatly improved if (re-)written using such tools, by benefiting from portability and, consequently, from a larger user base.

A user-friendly editing tool

Another request from many DELTA users is the availability of a user-friendly, specialized data editor, that should release users from the cumbersome and error-prone creation and maintenance of DELTA datasets using an ordinary text editor or word processor.

Early attempts to provide such a specialized DELTA editor were those of Pankhurst's DEDIT and Gouda's DDATA (distributed as part of his TAXASOFT package). However, none of these editors are GUI-based, which may become a major drawback to many users of modern operating systems and application programs. Worse, since MS-Windows and other GUI environments have highly standard interfaces, with common control objects (dialog boxes, pulldown menus, etc.), the authors of these editors will have to write entirely new applications when porting their programs to such an environment.

My specialized "DELTA Coding System" (DELCODE) is a first attempt at providing a GUI-based highly interactive environment for the editing of DELTA datasets. A DOS version is finished and a Windows version is well advanced. However, it is not a production system, but rather a teaching tool (that is the way I am using it). It is the result of one of the experiments I mentioned above, and is itself experimental. DELCODE does not intend to be a rival of the undoubtedly powerful, Windows-based, DELTA editor currently under development by the DELTA team. Its primary purpose is to test algorithms, data structures, and user interface ideas. It features pulldown menus, dialog boxes, and full mouse support, but the DOS version is obviously not a full GUI-based application and detaches at several points from the "classical" CUA/SAA (Berry, 1988) standards.

Up to date, all DELTA editors implement what we might call a "form fill-in" interface (see Diederich and Milton, 1993), with which the user enters data, usually from the keyboard, in specific fields on pre-built forms. This system is also used by the Windows versions of some DELTA programs (namely, INTKEY4 and INTIMATE), to get information from the user. However, as one of the DIANA users has quite rightly pointed out, character-by-taxon data matrices are the logical way to manage taxonomic data (Coddington, pers. comm.). A matrix is convenient, already known to most systematists, and is the accepted "metaphor" for comparative data on taxa. Visually, a user can chase homologies across taxa, or taxa through morphological space. So, a GUI-based DELTA data entry/editing system should be able to provide the user with some sort of matrix-oriented editor.

One such a matrix editor for taxonomic data already exists and might provide a relevant paradigm for further development of a user-friendly, GUI-based, biological data entry and editing system. That is the editor of Maddison & Maddison MacClade's interactive character analysis program, available only for the Macintosh computer. A good summary description of this editor (and of MacClade itself) — for those who do not have access to a Mac or do not have a copy of MacClade — can be found in Maddison and Maddison (1989). Indeed, MacClade's editor behaves somewhat as an interactive "character designer", which helps the user "to think about the biology behind the data" (Maddison and Maddison, op. cit.). This is a very innovative and promising approach that should be pursued further in the development of a GUI-based DELTA editor.

Another of my experiments with GUI software for DELTA was the development of a matrix-oriented editor for DELTA. A crude prototype was implemented in Visual BASIC (that provides a rather limited spreadsheet-like custom control), but it has not yet been developed further. Parts of it have been incorporated into the code of the LORIS browser program for DELTA databases, currently under development.

Reusable components

In recent years, the demands placed on software for management of taxonomic data have increased dramatically in response to an ever-increasing concern that such data are the key to biodiversity conservation and sustainable use of biological resources. As a result, the very few software-developers engaged in the field of biodiversity information management (including the members of the DELTA development team) are being highly pressed to satisfy all user needs in less time and with more efficiency. As with other fields of human activity, biodiversity data management is also faced with a "software crisis" (Cox, 1986; Gibbs, 1994), and perhaps software engineering should also be enlisted as a "crisis discipline", along with conservation biology and cancer biology (Soulé, 1985).

Such a crisis in the development of biological software means that not only are better and portable software development tools (as the VIBRANT interface library above mentioned, or the more expensive commercial ones) needed, but also that a biodiversity information management package far more programmable by the end-user should be developed. The access to preprogrammed subroutines would allow systematists with limited programming experience to develop their own specialized programs for biodiversity data management (eg., in her/his own national language). In the future, it might be more fruitful for DELTA developers to put their efforts in this general direction, rather than trying to anticipate and meet (with scarce resources) the precise needs of the next generation of computer taxonomists.

Along this line of thinking, I have developed DELTA Library, a general-purpose library of routines for reading text files in DELTA format, designed to work with any language that supports Windows DLLs (Dynamic Link Libraries). They allow a programmer using any language that supports DLL calling, such as Borland Pascal/Turbo Pascal for Windows, C/C++, and Visual BASIC, to write an application that can access DELTA datasets without effort, with just one line of code for each routine's call. I have already been able to sucessfully integrate them with LORIS, an experimental "DELTA database browser" I am currently writing in Visual BASIC for Windows, and these routines are also being used as the back end of the Windows version of my DELCODE editor for DELTA.

Anyway, such libraries of pre-programmed routines would be only the first step towards actual "off-the-shelf" reusable software components — best exemplified today by Visual BASIC's custom controls (VBX's) — that may be used to build real applications in less time and with less effort (Udell, 1994). These and many other developments of the current software industry are likely to yield most rewarding results in the design and implementation of reliable software to successfuly face the data management issues posed by the the sheer diversity of the biota they are intended to help preserve.

References

Berry, R.E. (1988). Common User Access - a consistent and usable human-computer interface for the SAA environments. *IBM Systems Journal* 27: 281-300.

Cox, B.J. (1986). *Object-Oriented Programming - An Evolutionary Approach*. Addison- Wesley Publishing Company.

Diederich, J. and J. Milton. (1993). Expert workstations: a tools-based approach. In: *Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, Computer Vision*, R. Fortuner (Ed.). The Johns Hopkins University Press, Baltimore, Maryland, pp. 103-124.

Gibbs, W.W. (1994). Software's chronic crisis. *Scientific American* 271: 72-81.

Maddison, W.P. and D.R. Maddison. (1989). Interactive analysis of phylogeny and character evolution using the computer program MacClade. *Folia Primatologica* 53: 190-202.

Soulé, M.E. (1985). What is conservation biology? *Bioscience* 35: 727-734.

Udell, J. (1994). Componentware. *Byte* 19: 46-56.

DELTA Data Format and Pottery Classification

Mikko Louhivuori

Israel Antiquities Authority, Jerusalem (mikko@israntique.org.il)

Mike J. Dallwitz's DELTA data format has grown some strange branches in Israel where the Antiquities Authority has developed a computerized pottery classification system that is based upon it (the program is currently undergoing beta testing). Formal identification of inanimate objects found in archaeological excavations is not as far removed from biological classification as one might think. Admittedly, variations between types are man-made rather than natural and the time axis may be more significant than in the world of insects or plants. The underlying principles of classification are nevertheless rather similar and follow the logic of accurate observation and type definition.

The archaeologists trying to arrange pottery assemblages into chronologically and culturally meaningful groupings face problems that probably are quite familiar to those involved in developing biological taxonomies. There is a need for a practical, flexible system that allows the mixing of classification criteria and that accepts a variety of data types, including morphological and scientific details, value ranges, pseudovariables and so on. The system must handle all kinds of exceptions that frequently wreck rigid typing systems. The DELTA format solves such problems elegantly. As a platform independent format it has the potential to be widely adopted for computerized ceramic classification.

A common standard is required when detailed pottery data is exchanged between databases over the Internet between diverse computing environments and independently developed software packages. Today, there is no commonly accepted data format and automated searches within the framework of wide area distributed databases is therefore currently not possible.

For an archaeologist one of the main attractions of the DELTA format is the way it allows him or her to concentrate on the diagnostic features only. The pottery expert may quickly define significant features in a spatially or temporally delimited assemblage and leave the description of shared aspects to the level of general type description. This helps the analyst to create sharper tools for ceramic typology, tools that may eventually grow into pottery expert systems developed for specific regions or archaeological periods. Such CD-Rom or diskette distributed systems may greatly assist other archaeologists working with similar materials.

We have implemented the DELTA data format pottery typology system using the relational database model. The program works in PC computers in Microsoft FoxPro 2.6 for Windows and may be connected to the Unix Oracle RDBMS SQL*Net server through TCP/IP and Microsoft ODBC layers. However, the specific programs used to implement the system are of secondary importance, the main point being that the software can read and write DELTA format ASCII files.

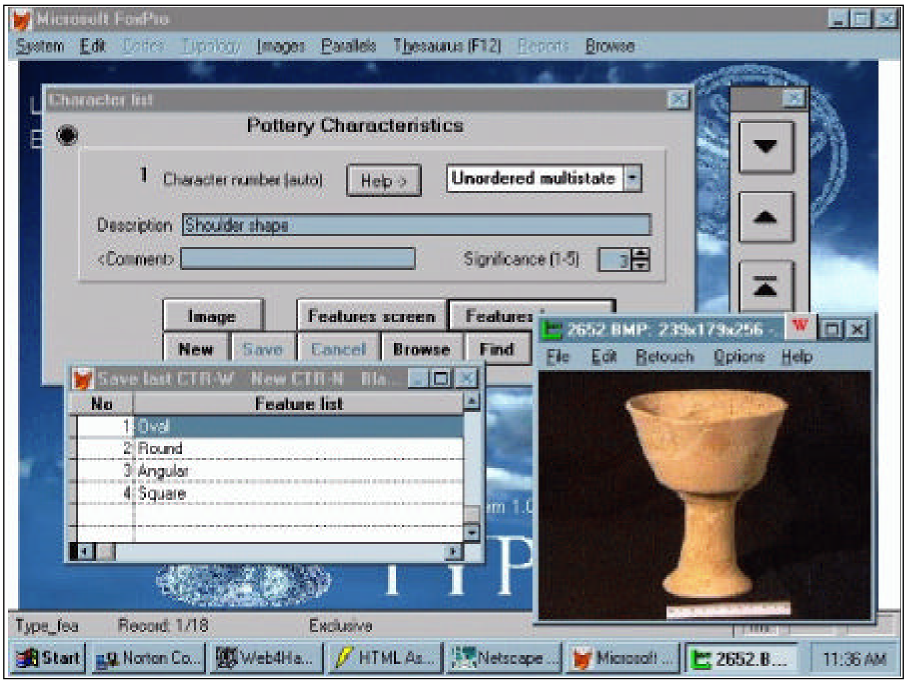
The structure of the character/feature definition tables is very simple but it nevertheless allows quite advanced SQL searches on different levels of detail:

Table TYPE_CHR		**&& Main character definition table**
TP_CHR_ID	NUM(6)	&& Internal character ID
TP_CHR_TP	NUM(1)	&& Code for character type*
TP_CHR_DE	CHR(60)	&& Formal descriptive name
TP_CHR_CM	CHR(60)	&& Free comment (*character types: unordered/ordered multistate, integer, real number, text)

Table TYPE_FEA		**Main feature definition table**
TP_CHR_ID	NUM(6)	&& Internal character ID
TP_FEA_ID	NUM(6)	&& Internal feature ID
TP_FEA_DE	CHR(60)	&& Formal descriptive name

As any DELTA format programmer knows, the main effort in application development using such structures is in the writing of the supporting code that validates user input and that assists him or her in creating and maintaining the code tables. The creating of natural language output from coded language also has many pitfalls and complications. However, it seems that GUI software developers are increasingly adopting a philosophy that focuses on ease of use rather than ease of programming, a view that can be found early on in DELTA data format documentation!

In addition to standard DELTA type definition modules, the IAA pottery typology system has a loose hierarchy



Pottery Characteristics

1 Character number (auto) Unordered multistate

Description: Shoulder shape

<Comment>

Significance (1-5) 3

No	Feature list
1	Oval
2	Round
3	Angular
4	Square

Type_fea Record: 1/18 Exclusive

Figure 1. A screenshot from the Fox Pro front end and the external image viewer.

that is used to express simple parent ->child ->child, child ... dependencies. Such relations may have geographical or chronological significance. Evolutionary relationships between type variations may occasionally have great importance in dating stratified layers. The current DELTA data format definition does not define such relations and suggests instead variant definitions that may lead to a certain degree of redundancy. (It is said that future definitions may include provisions for specifying a taxonomic hierarchy). Our implementation of a relational tree allows great flexibility in building the hierarchy tree but it requires the programming of a rather involved library of supporting procedural SQL code.

Table POT_CLASS		**General class definition table**
class_id	NUM(6)	&& Internal class ID
class_nme	CHR(40)	&& Formal class name
family_wid	NUM(6)	&& IAA Thesaurus links: Family
perf_wid	NUM(6)	&& Period range From
pert_wid	N(6)	&& Period range To
class_memo	Memo	&& Free format class description

Table TYPE_TRE		**Pottery type tree hierarchy**
ADDRESS	CHR(24)	&& f.ex. “000100010001”
LEVEL	NUM(2)	&& Level in tree (f.ex. 3)
CLASS_ID	NUM(6)	&& General class
TYPE_ID	NUM(6)	&& Internal type ID

In addition to these basic modules, the analyst of ancient pottery also needs a parallel type definition system that can be used to refer to published pottery objects. This module has a simple input screen that relies on the IAA Thesaurus for standard definitions and is independent of the DELTA format. The parallel list has an uncomplicated one-to-many relation to the type list.

Digital images are supported by a sub-system that is used to link picture files to the data. Each type definition stage can be illuminated with pictures. The program itself does not include a display engine but allows, instead, the installation of an external image viewer. This approach has the advantage that a single software package may support all kinds of image formats simply by letting the user install different kinds of external viewers.

The package is currently under beta testing by a selection of pottery experts dealing with different archaeological periods. The initial response has been promising and it seems that the DELTA data format may solve many of the problems that have plagued our previous attempts to computerize the processes involved in the creating of pottery typologies.

Recent Enhancements of the DELTA programs of the CSIRO Division of Entomology

M. J. Dallwitz, CSIRO Entomology

QuickWin versions of programs

QuickWin versions of Confor, Delfor, Key, and Dist are now provided. These run under MS-Windows, and are the same as the MS-DOS versions, except that much more memory is available. When you run these versions directly from Windows, you will be prompted to enter the full path name of the directives file. Nicholas Lander's DELTA Menu System has been updated to use these versions of the programs.

Intkey

A version of Intkey for MS-Windows was released in May 1995. It uses the same data files as the MS-DOS version, and the command languages are compatible. New features include: a completely new user interface; 'simplified', 'automatic', and 'manual' modes of operation; automatic loading of the user's preferred settings; provision for overlaying text on images; selection of character states from images screens; access to character notes from image screens; the ability to view images of any size and number of colours on any display; scaling and scrolling of images; the ability to have several images on screen at the same time; handling of JPEG (as well as GIF) images; the ability to load new data sets without terminating the program; selection of input files from subject-oriented lists (rather than only by file name); and a user-definable toolbar.

The simplified mode of operation provides facilities for basic identification, and for displaying descriptions and illustrations of taxa. It is intended primarily as an introduction to using the program, but is quite powerful, and many users may find that it satisfies all their needs. This mode has only 12 first-level menu options, compared with 37 in the other modes. Of these 12, only 6 are specific Intkey operations; the other 6 are operations common to most Windows programs — window housekeeping (e.g. tiling), help, and program exit. The dialog boxes are also simpler, having at most 4 buttons, compared with up to 10 in the other modes. The command line (for bypassing the menu system) is not available. A short but complete help window (194 words) is displayed when the program starts, and may be kept on screen while the program is being used. If this window is not closed, messages suggesting a course of action (depending on context) are displayed whenever there is no activity for a certain length of time.

In automatic mode, various actions such as the display of images are carried out automatically (but may be cancelled by the user if not required). This mode works exactly the same as simplified mode; the only difference is that the full range of operations is available. Therefore, once simplified mode has been mastered, the transition to automatic mode presents no difficulty, and the other options can be explored if and when they are needed. In manual mode, all actions must be specifically requested.

Toolbar buttons may be defined by the author of a data set (or by the user) to represent any command or sequence of commands. These can be used to provide quicker access to frequently used commands, to specify which dialog boxes are displayed, and to extend the range of commands available in simplified mode. As the cursor moves over menu options and toolbar buttons, brief descriptions of them are automatically displayed.

Text can be overlaid on illustrations without being embedded in the image files. This makes it easier to alter the text, or to have the text in several languages. The text that is overlaid on the character images can be taken automatically from the character list. Character states can be selected from these overlays by clicking either on the text of the required states, or on the relevant part of the illustration itself.

The following changes were made in both the MS-DOS and Windows versions. When text characters are Used, the settings of Match Unknown and Match Inapplicable are ignored, and substrings separated by '/' can be used (see online help for details). A new Set parameter, DiagLevel, replaces Tolerance+1 for use with the Diagnose command, and the name of the Set parameter Diagnose has been changed to DiagType. In the calculating of the separating power in the Best command, attributes which are sometimes (but not always) inapplicable are now treated as though they were always applicable; that is, '/–' in an attribute is ignored, whether it is coded explicitly or is implicit because of a character dependency. Controlling characters with a separating power of 0 are now not displayed in the Best dialog box.

Intimate

A new program, Intimate, has been provided to create and edit the text overlays used by Intkey. The boxes in which the overlays are displayed can be graphically repositioned and resized. The program outputs CHARACTER IMAGES and TAXON IMAGES directives which are used by Confor when creating Intkey data files.

Confor

Facilities for generating natural-language descriptions in Microsoft Rich Text Format were contributed by C. M. Weiller and G. F. Weiller of the Australian National University. This format can be imported into MS Word and some other word processors.

The TAXON IMAGES directive was added (as an alternative to CHARACTER FOR TAXON IMAGES). The directives ADD CHARACTERS and EMPHASIZE CHARACTERS now accept either taxon names or taxon numbers in their input. Repeated use of these directives is cumulative. The directive DISABLE DELTA OUTPUT was added; it prevents the use of OUTPUT DESCRIBE and OUTPUT SUMMARY in Intkey. A directive OMIT SPACE BEFORE UNITS was implemented. Blank comments are now omitted from natural-language output, instead of a space being inserted (this allows blank comments to be used to break attributes which are too long to fit on a single line).

Key

The directive TREAT CHARACTERS AS VARIABLE was added. Its effect is to change the values of specified attributes to 'variable', which allows the construction of more robust keys. A new sample directives file, KEY6, illustrates its use. Δ

DELTA NEWSLETTER BACK ISSUES

Limited back issues of *DELTA Newsletter* numbers 1--12 are now available. These issues contain much useful information and advice and provide an invaluable reference to any biologist interested in the area of computer-aided taxonomy. They can be ordered individually or as a set. Please indicate which issues you require and how many of each. Each issue is a fixed price of $5.00 each, which includes postage and handling.

Please note that extra-Australian personal cheques cannot be dealt with; instead make International Money Orders payable to **The DELTA Newsletter**.

Name ____

Address ____

____

____

Country ____

Email address _______________________________________

Please send me the following DELTA Newsletter back issues. (Tick applicable box for each Newsletter requested and indicate if multiple copies are required.)

1❑x___ 2❑x___ 3❑x___ 4❑x___ 5❑x___ 6❑x___.
7❑x___ 8❑x___ 9❑x___ 10❑x___ 11❑x___ 12❑x___.

Total number of back issues: ___ @ $US5.00

Total payment enclosed: $US_____________.

Signed ________________________ Date _________

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

Using Intkey data files directly from the WWW

M. J. Dallwitz

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

In the current version of the interactive identification and information retrieval program Intkey, the data and image files must be on the computer's own disks, or on the disks of a LAN (Local Area Network) to which the computer is connected. To use the data available on the DELTA World Wide Web (WWW) site at http://muse.bio.cornell.edu/delta/, the files must first be downloaded to a local disk by means of ftp, gopher, or a WWW browser.

We will soon be modifying Intkey so that it will be able to access its files directly via the WWW. Note, however, that it will still be necessary to install and run Intkey itself on a MS-Windows PC — we do not intend to make a version which will run on a Unix server. When running this version of Intkey, it will be possible to load a data set by specifiying an initialization file (usually called intkey.ini) which is a URL rather than a local file. For example, for Leslie Watson's Angiosperm families data, the data would be loaded by specifying

http://muse.bio.cornell.edu/delta/angio/intkey.ini

rather than c:\angio\intkey.ini. As Intkey sets the default directory for other files to the directory of the initialization file, the data files (ichars and iitems) would automatically be taken from the same WWW directory, and the image files would be taken from the appropriate subdirectory.

When a data set is first used in Intkey, the user is given the option of adding the name of the initialization file, together with a description such as 'Angiosperm families', to an index. Once this has been done, the data set can be accessed on future occasions by selecting its description from the index, without needing to remember where files are located.

It will also be possible to initiate an Intkey session via a Web browser such as Netscape. The browser will be configured to use Intkey as a 'viewer' for Intkey initialization files, and Web documents will contain links to these files. By clicking on text such as 'Identify Angiosperm families', an Intkey session using the appropriate data files will be initiated. When the Intkey session is terminated, control will return the the Web document.

We will use the API's of the Netscape and Mosaic Web browsers for file access and caching. At the moment, we have this working for image files in Intimate, which is a program for annotating Intkey images, and also serves as a testbed for developing modules for Intkey. The process is transparent to the user. When the program wants a remote file, it starts Netscape (or Mosaic) if it is not already running, and asks it to fetch the file.

If you want to test access of WWW files in Intimate, you should use Version 2 or later of Netscape, and a version of Intimate *later than* the release of 5 January 1995 (at the time of writing, the appropriate version is in the file test/intimat@.exe at the DELTA WWW site). We have set up a demonstration using the beetle-larvae data available at the DELTA WWW site. You should download and install the data as described in the accompanying documentation, but there is no need to download the image files (unless you also want to use the data with Intkey).

Also download the file test/ctimages, and place it in the same directory as the data files. Then start Intimate, select ctimages as the directives file, select illustrate/characters (or taxa) from the menu bar, and select one of the bold entries in the dialog box. The corresponding image will be downloaded and displayed. Files downloaded in this way will be cached by Netscape, and, as long as they remain in the cache, will be available without downloading again, both in the same Intimate session and in later sessions.

References

Lawrence, J. F., Hastings, A. M., Dallwitz, M. J., Paine, T. A., and Zurcher, E. J. (1995 onwards). 'Larvae of the Elateriformia (Coleoptera): Interactive Identification and Information Retrieval for Families and Sub-families'. URL http://muse./bio./cornell./edu/delta/

Watson, L., and Dallwitz, M. J. (1992 onwards). 'The Families of Flowering Plants: Interactive Identification and Information Retrieval.' URL http://muse./bio./cornell./edu/delta/

A Possible Future Of DELTA Tools For The Web

Mike Huben

Email: mhuben@world.std.com WWW URL: http://world.std.com/~mhuben

The growth of the Internet, and more specifically the World Wide Web, has brought us a wealth of new possibilities for convenient DELTA tools. The latest generation of Web browsers have capabilities far beyond the rather static features of the Web's HyperText Markup Language (HTML): specifically the Java language and environment. Java holds the promise of sophisticated, portable Graphical User Interfaces (GUIs) for DELTA tools that could either make full use of Web capabilities, or run standalone on isolated personal computers.

The near future of computing will involve near-seamless networking applications. With these applications, you will be able to perform your daily computer tasks from any site, and not really care how remote you are from the server for the data or the server for the computation. Web browsers supporting Java are the first widespread utilities supporting these applications, and lots more are in development. Eventually, you will not much care what kind of a box you have in front of you or what operating system it is running, because all that those will do is support the application layer which you interact with. For example, if all you do is run Netscape, you don't have to care if you are on UNIX or DOS or Macintosh. And now, with Java, running Netscape is all you might have to do.

The key idea behind Java is that programs should move between computers as seamlessly as documents now do via the web. We all remember the bad old days when we had to invoke an ftp program to download a formatted document to our own disk, and then print or edit the file with a separate program to be able to inspect it. Now, with web browsers, all that is a mouse click. In these days, to run a program, you need to run a program to download it from another site onto your own disk, install it, then invoke it. Imagine if all it took was that same mouse click through your web browser? That's what Java-capable browsers can do.

What is Java? The buzzword description is: "Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language." (Java is extensively documented at the "Java(tm): Programming for the Internet" web site, http://www.javasoft.com/index.html) The features that interest us most are that it is very portable, and can be readily used to access information across the Internet. Java is a full-scale language: HotJava, for example, is a web browser written entirely in Java.

Java alone doesn't buy us everything, though. Java applications can run standalone outside of Web Browsers, but if we run them as applets inside Netscape 2.0 (and eventually other Web Browsers), then we get a whole package of desirable features without any work:

- Portability across PC's, Macs, and major UNIX machines.
- Simple, uniform GUI development across all those platforms.
- Full Web Browser hypertext ability.
- Full GUI support of all major languages, including Asian languages.
- Full GUI support of numerous image formats.
- Access to data and programs both local and from across the Internet.
- Fully threaded processing to eliminate many waits.

What would be the place to start? I'd say start with a substitute for INTKEY. The vast majority of interacting with DELTA information is probably by users of INTKEY. Imagine downloading the latest INTKEY in Java from Australia with the data from anywhere for your favorite taxa from Netscape without ever actually having to store or configure anything on your machine. When you'd keyed your specimen to genus, you could click the hypertext link to key it to species. And when a morphological term was unfamiliar, you could click it to bring up an illustrated hypertext glossary page.

There are lots of real issues to consider before we blythely hype Java as the cure for the common cold, let alone the future of DELTA tools:

- Java may fizzle or Netscape may drop it.

- Java applets are currently rather inefficient and buggy in their current beta state. This will improve eventually.
- Java applets will likely require more memory and speed than the current highly optimized programs. This problem will go away as machines increase in memory and speed.
- Java is not yet ported everywhere: it will soon be on Macs, and then probably on Netscape for Macs.
- The security of Java might prohibit simple means of saving results from Java applets.
- Web development is still at an early stage: the vast majority of what is developed today is very fragile, hard to test, and prone to breakdown.
- Java is Object Oriented, which might be an obstacle for many of today's tool developers.

On the bright side:

- Java is Object Oriented: development in Java may prove more rapid and efficient than development in older languages.
- Netscape and Java can run on isolated, unnetworked machines with local data.
- Netscape 2.0 finally has enough GUI widgets to do a proper job.
- Oracle/Java interfaces are under development, which could integrate well with future database efforts for DELTA. This might enable remote editing of databases.
- Direct usage of an INTKEY substitute could be available for taxa through a single mouse click, from the Madison's "Tree Of Life" for example.

The possibilities for further integration of systematics information through the web are impressive. Imagine following the trail from a key to the specimen upon which a key is based to the museum database where that specimen was cataloged to the identifier of that specimen to his publications, through a citation index to a paper discussing the new name of the taxon, and back to the key from the references in the paper? We can do this all right now through a complex manual search: imagine when it is as easy as a few clicks of the mouse?

As more and more information becomes available over the web, I don't think systematists will fall behind: indeed we seem in the front. Java may be the appropriate choice for the next leading edge systematics application. Δ

WORKSHOP ANNOUNCEMENT

INTRODUCTION TO THE DELTA SYSTEM

An introductory course to the DELTA system will be held at the Australian National University, Canberrra, Australia, July 8th-12th 1996.

The course will cover:

the establishment of character lists
data coding
output of natural language descriptions
generation of printed keys
development of, and using INTKEY sets
images & INTIMATE (INTkey IMage AnnoTation Editor)
the output of files for phenetic and cladistic programs.

The course will be led by:

Drs Carolyn Weiller (Research School of Biological Sciences, ANU),

Les Watson (Albany and RSBS) and

An Van den Borre (School of Biological Sciences, University of Sydney).

The course is part of series of workshops for the Graduate School, ANU. Further details and registration information are available from the following URL:

http://biology.anu.edu.au/Pages/Sem/Bioinfws/bws11.html

Registration closes May 10th.

Cost: $100

If unable to access the above URL, please contact me by email at the address below for registration details.

Dr. Carolyn Weiller
Research School of Biological Sciences
Australian National University
CANBERRA, A.C.T. 0200
Phone: 61 6 249 4841
Fax: 61 6 249 4437
email: carolyn@rsbs.anu.edu.au

TAXASOFT DELTA Editor

by Eric J. Gouda

Jungfrau 107NL - 3524 WJ Utrecht Netherlands
Email gouda@cc.ruu.nl

The way one has to produce and maintain several DELTA data files, directive or batch files, is a severe threshold for many taxonomists to use DELTA. This especially for smaller data sets. To make the SPECS, CHARS and ITEMS files, mostly simple ASCII text editors are used. You need to renumber all your data if you want to insert a character or character state. There are many things you have to keep in mind if you change something in one of the files, this because of the linkages between the specifications, character descriptions and the code of the items (taxa). An-easy-to use interface to the DELTA data is necessary to make it easier for non-computer specialists to use DELTA.

TAXASOFT started as a program to process DELTA files to produce keys for identification in an easy and interactive way. When for the first time I had some students try out the program, I soon became aware of the problems described above. I then started to develop an interface to make the use of DELTA format much easier. The most important module of this interface became the Taxasoft editor (DDATA.EXE), that can be started from the Taxasoft menu or run as a stand-alone program.

DDATA is not just an text editor, it is a real DELTA Data Maintenance program that hides the way the data is stored from the user. It combines the contents of the three DELTA files within two windows, one window that gives you the character list and item (taxa) names list, and another window that allows you to enter the data. It lets you access your data rather like a database, instead of plain files, the contents of which are more difficult to understand.

The specification file (SPECS), holds several specifications about the characters descriptions file (CHARS) and items data file (ITEMS) as:

- character type of each character
- which characters are linked
- from which characters a new paragraph starts
- reliabilities of characters
- implicit values for multistate characters
- dependencies between characters

With DDATA.EXE all data mentioned can be created, and can be edited simultaneously within one window. You can insert, delete or move characters, character states and items whenever you like, and the files will stay consistent with each other. There is an easy interactive way of selection, to link characters together, set implicit values and dependencies, etc.

The Taxasoft interface to the DELTA files

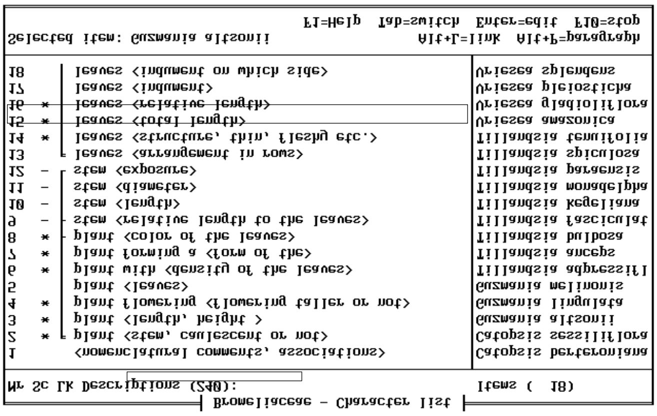
The following divided window, see fig. 1, will appear on the screen after starting DDATA. The left part is a scroll list of the character descriptions. The right part is a scroll list of the item descriptions. The header will give information about the data and abbreviations of the columns.

Nr=character number

Sc=Score ('*' scored, '-' inapplicable, ' ' not scored)

Lk=Linkage of characters, visualized by a vertical line

õ = start of paragraph.



```
                    Bromeliaceae - Character list
Nr Sc Lk Descriptions (240):                      Items (  18)
1        <nomenclatural comments, associations>   Catopsis berteroniana
2   *  ┌ plant <stem, caulescent or not>          Catopsis sessiliflora
3   *  │ plant <length, height >                  Guzmania altsonii
4   *  │ plant flowering <flowering taller or not> Guzmania lingulata
5      │ plant <leaves>                           Guzmania melinonis
6   *  │ plant with <density of the leaves>       Tillandsia adpressifl
7   *  │ plant forming a <form of the>            Tillandsia anceps
8   *  ├ plant <color of the leaves>              Tillandsia bulbosa
9   -  ├ stem <relative length to the leaves>     Tillandsia fasciculat
10  -  │ stem <length>                            Tillandsia kegeliana
11  -  │ stem <diameter>                          Tillandsia monadelpha
12  -  └ stem <exposure>                          Tillandsia paraensis
13     ┌ leaves <arrangement in rows>             Tillandsia spiculosa
14  *  │ leaves <structure, thin, fleshy etc.>    Tillandsia tenuifolia
15  *  │ leaves <total length>                    Vriesea amazonica
16  *  │ leaves <relative length>                 Vriesea gladioliflora
17     │ leaves <indument>                        Vriesea pleiosticha
18     │ leaves <indument on which side>          Vriesea splendens
Selected item: Guzmania altsonii          Alt+L=link  Alt+P=paragraph
                    F1=Help  Tab=switch  Enter=edit  F10=stop
```

Figure 1. Selecting from character or item list

The number of characters and items are given between parentheses.

The footer of the divided window gives the name of the selected (active) item, some keys that are important, and other information as number of temporarily deleted characters or items on the scrap (these can be inserted again).

Online help can be activated with the help key F1, and will give specific help windows depending on your input window (context dependent).

Character selection window part

Use the cursor control keys, viz.: Arrow keys, PageDown, PageUp, Home, End key to scroll through the character list to search a character description. When Enter is pressed, the character highlighted by the scroll-bar will be selected for editing. The window in fig.2 will appear. If an item is selected too, the item data can be edited, otherwise the character description itself can be edited.

To link characters in the list, just press Alt+L, which marks the start of character linkage. Press Alt+L again to remove existing link, otherwise go to last character to be linked and press Enter. To mark a new paragraph (or remove mark, if already marked) Alt+P is used.

You can delete a character and put it on the scrap by pressing Del key. All items will be updated directly, but **NO** data will be lost until the files are saved or until they are removed from the Scrap using F9 key. Deleted data can be inserted again, using the Ins key, at any location (see also the move function). If there is no character on the scrap an empty character record is inserted.

To move the current character to an other location, the F3 key can be used. This will be faster than deleting and inserting a character.

Items can be selected using the item selection list or using the item paging keys. These keys are: F6=Unselect item; F7=Select previous item; F8=Select next item .

To Stop the program and save anything that has been changed, you can use either F10 or Esc from this point. With the Tab key you can switch to the item list on the right side of the window, for item selection or other item function.

Item Selection window part

The right part of the window is for item selection and other major item functions. The same as with the character selection you can use the cursor control keys, to scroll through the items list. When Enter is pressed the item highlighted by the scroll-bar will be selected for editing. If this item was already selected, the item will be unselected. After selecting an item, you will return to the character selection window part.

It is also possible to search an item with the search function (Alt+S), and give an unique part of the name that must be searched. Items can be deleted and inserted again at another position.

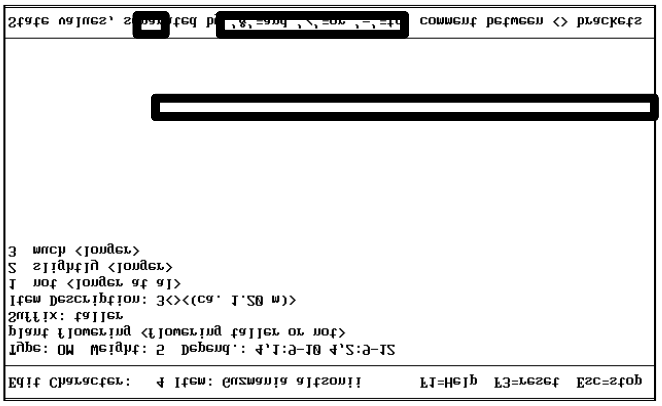
Other functions are: -Copy (duplicate) item at current position (Alt+C); -Change the name of the item at current position (Alt+N); -Unselect/Reselect item, without leaving the item selection menu (key F6); -Remove all deleted Items from the Scrap (key F9)

Using the Enter key to select an item or the Esc key, you will leave the item selection window.

Editing Character and Item data

By selecting a character from the character descriptions list, you will enter another window (fig. 2), where character description and specifications are shown and can be edited. When an item has been selected, the item data for this character can be edited in the same window.

Editing the item data you can use PgDn or PgUp keys to get the data for the next or previous character, or use function keys F8 or F7 to select the next or previous item while staying with this character. So it is possible to compare items in a sequence for any selected character, which can't be done with a text editor.



```
Edit Character:   4 Item: Guzmania altsonii        F1=Help  F3=reset  Esc=stop
Type: OM  Weight: 5  Depend.: 4,1:6-10 4,2:6-12
plant flowering <flowering taller or not>
Suffix: taller
Item Description: 3<><(ca. 1.20 m)>
1  not <longer at al>
2  slightly <longer>
3  much <longer>

State values, separated by '&'=and, '/'=or, '-'=to, comment between <> brackets
```

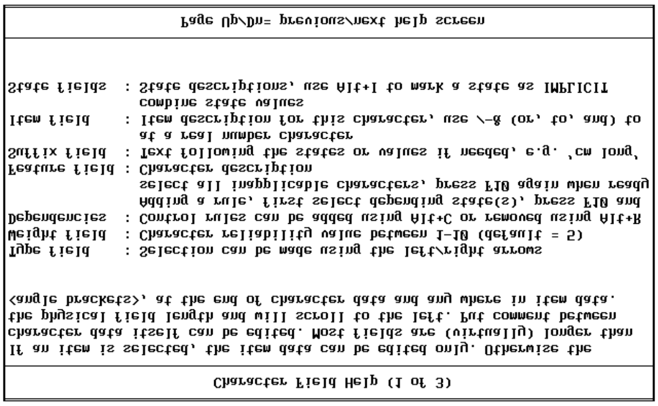
Figure 2. Editing character or item data

Data can be altered, inserting the DELTA code, with the great advantage that you can see the meaning of the code in the same window. The syntax of entered code will be checked.

Changes on data can be ignored using the function key

F3, as long you do not leave this window. The active item name is always in the header. The bottom line is used to give (context dependent) help on current the input field.

There are several help windows in this mode that can be activated with the function key F1. These windows can be viewed in sequence, using PgUp and PgDn keys. The first one is the 'Character Field Help' window that gives the explanation of the field function and input rules, see figure 3. The second one is the 'Editor Key Function Help' window, which gives all possible record functions (as reset, selecting next item etc.) and special function keys for activating implicit values; inserting or deleting dependencies etc. The last help window, is the 'Editor field help', which gives all edit functions within a field and between fields (as jumping to next, previous, last and first field, etc.).



```
                 Character Field Help (1 of 3)

If an item is selected, the item data can be edited only. Otherwise the
character data itself can be edited. Most fields are (virtually) longer than
the physical field length and will scroll to the left. Put comment between
<angle brackets>, at the end of character data and any where in item data.


Type field    :  Selection can be made using the left/right arrows
Weight field  :  Character reliability value between 1-10 (default = 5)
Dependencies  :  Control rules can be added using Alt+C or removed using Alt+R
                 Adding a rule, first select depending state(s), press F10 and
                 select all inapplicable characters, press F10 again when ready
Feature field :  Character description
Suffix field  :  Text following the states or values if needed, e.g. 'cm long'
                 at a real number character
Item field    :  Item description for this character, use /-& (or, to, and) to
                 combine state values
State fields  :  State descriptions, use Alt+I to mark a state as IMPLICIT


              Page Up/Dn= previous/next help screen
```

Figure 3. Character field help

Most fields are (virtually) longer than the physical field length and will scroll to the left, data can be entered up to 1 Kbytes. If you want to see more data, there is a function to view the data of the current field in full screen.

It is also possible to insert extended ASCII characters by using the Alt+<number> key combinations or activating a special token selection function (Alt+S).

Δ Δ Δ Δ

Institutional DELTA Databases: A Case Study

Alex R. Chapman and Mike Choo

WA Dept of Conservation & Land Management
Email: alexc@calm.wa.gov.au, mchoo@calm.wa.gov.au

A number of workers at the Western Australian Herbarium have had a long involvement with developing descriptive datasets using the DELTA methodologies in support of their own research projects. More recently DELTA has been used as the central tool for producing a new descriptive catalogue of the WA angiosperm flora and to produce a detailed interactive key to the state's rare and endangered flora, both of which are now nearing completion.

The creation of such institutional DELTA datasets and the need to manage them and link them to other corporate data has led to the development of an institutional design for the coordination of descriptive datasets. Although detailed discussion of this institutional design will be left for a separate paper, we would like to focus here on the four general database issues we consider integral to such a scheme.

1. Coordination of related DELTA character lists

The need for coordination between DELTA datasets initially arose when preparing the character list for the Rare and Endangered flora Database (RED). For this we started with a well known and comprehensive angiosperm DELTA character list (Watson and Dallwitz, 1994). This became the foundation of our comprehensive 'institutional character list' to which substantially new characters could be added, and existing ones chosen for new projects. From this we created a subset of 150 characters as the RED project character list. We also identified a much smaller subset of 'core characters' which were mandatory to include and score for any subsequent character list. This allowed some level of comparability and consistency between descriptions for any two coded taxa. This was particularly important for a project such as RED which included representatives from families as diverse as the Orchidaceae, Asteraceae and Myrtaceae.

2. Linking to fundamental datasets

A second component of the institutional design was initially established as part of the Descriptive Catalogue of WA Plants project. This set out to score a small number of core characters for all 11,000 odd species of angiosperm flora in the state. Perhaps the major pitfall in compiling such a large dataset is the problem of maintaining currency with the ongoing nomenclatural and taxonomic changes to the state's flora. Fortunately the WA Herbarium had established a comprehensive Census database some years before, and the project could avail itself of an associated simple but powerful PC database application that allowed users to define a database structure while maintaining a link to the Census master lists. As nomenclatural changes were recorded in the main Census database, users could interrogate their own datasets to pick up changes that needed to be addressed.

By coding DELTA descriptions directly into a standard relational database application not only could links to fundamental datasets be established, leading to great efficiencies, but the data was available in an accessible format so that in the future access to the descriptive data could be made from external databases.

3. Creation of a DELTA translator

In order to fully utilise a relational database application for capturing descriptive data, while maintaining the flexibility of coding the data in a DELTA format for subsequent output in various forms, a DELTA translator was required. Because the DELTA format was well documented and robust a translation script could be written to output captured data as standard DELTA coded ITEMS files. In the first instance the DELTA translator was enabled in this one direction only.

A further advantage of adopting an off-the-shelf database package as the data capture tool was that an efficient environment became automatically available for use in entering and editing the descriptive data. This went some way to assisting those users unwilling to master an ASCII editor to edit their ITEMS files.

4. Management of ancillary datasets

Along with the DELTA datasets themselves, there are a number of associated data types that need to be managed in an institutional framework. Of great importance when creating interactive keys are the taxon and character images. Apart from standards required to prepare the sets of images, a standard naming structure and a method of documenting those taxa or characters already illustrated must be implemented. In a similar manner, we need to document the availablility of standard taxon descriptions. These are either natural language descriptions from DELTA or the result of other descriptive work such as Flora treatments, but the aim is to have a standard description for each taxon in the state available for viewing.

The utility of such an approach is highlighted when one considers the explosion of taxonomic publications now available on the World Wide Web (WWW). By establishing a database documenting the production of standard images and descriptions for each taxon, together with a standard file naming protocol, they can be sourced automatically by Intkey for Windows or a WWW browser.

A DELTA Database Engine

With the establishment of each of these components it has been possible to prototype a comprehensive GUI database application which allows an institution to manage the growing number of descriptive and ancillary datasets. Major components of such a DBMS system will include:

Database Engine which provides the mechanism for data input, editing, exception reporting, character list management and all other database manipulation facilities.

DELTA Interface which includes a *DELTA to Database* translator to import DELTA SPECS, CHARS and ITEMS files into the Database Engine and populate the databases; and a *Database to DELTA* translator which is a more fully functional version of the translator described in 3 above.

Administration module to handle initialisation and setup as well as backup and restore routines.

While this prototype contains functionality already being built into other tools (many discussed in this volume) such as a sophisticated DELTA editor or a RDBMS backend to the DELTA software and standard, this is the first time an institutional tool for the intelligent management of descriptive data from numerous potential sources has been proposed. Many issues remain to be resolved, such as the final relationship of the individual DELTA datasets to the institutional conglomerate, however, we believe it has the potential to bring great efficiencies to major institutional ventures such as state or national floras.

References

Watson, L. and Dallwitz, M.J.(1994). 'The Families of Flowering Plants'. CD-ROM, Version 1.0 for MS-DOS. (CSIRO Publications: Melbourne.)

DELTA in PANDORA

Richard Pankhurst and Martin Pullan

R.Pankhurst@rbge.org.uk / M.Pullan@rbge.org.uk
Royal Botanic Garden Edinburgh, 20A Inverleith Row, Edinburgh. EH3 5LR.

The DELTA data format

DELTA (Description Language for Taxonomy) was originally designed to capture descriptive taxonomic data in a flexible yet comprehensive way. Over the years the format has been expanded, allowing extra information about items to be stored. Typically this extra information is placed in one or more text fields and will cover such areas as nomenclature, bibliographic details and geographical information about an item. A disadvantage of this approach is that there is no distinction between text fields on the basis of their data content and therefore this 'satellite' data is somewhat unstructured and cannot easily be interrogated.

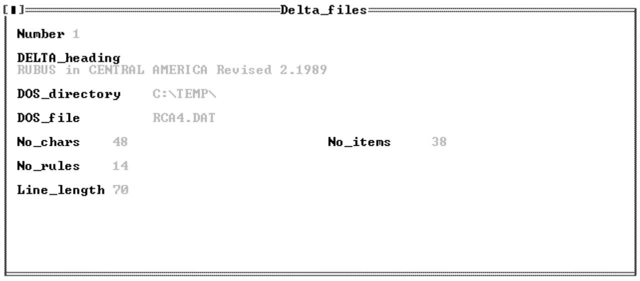
There are, however, a number of taxonomic data base systems that are designed to store and retrieve taxonomic, bibiographic and geographical information. It is therefore apparent that in order to provide access to the full richness of the data stored in a DELTA data set, it would be useful to merge the features of a taxonomic data base system with those of the DELTA format. Implementing DELTA in a relational database, as well as improving the storage of the satellite data would also mean that the character and item descriptions within the DELTA format could also be interrogated using the standard data base query tools available in data base systems.

With this in mind DELTA features have now been incorporated into PANDORA, and to our knowledge PANDORA is currently the only taxonomic database system that includes a comprehensive set of DELTA tools. One reason for this is that PANDORA has been developed in Advanced Revelation (AREV), a post relational database management system. Such systems allow the storage of multiple values in a single data field and do not require the field size to be defined. This makes it very easy to store and manipulate multi-state characters, and also to store data on items that can attain multiple states for a single character.

The representation of DELTA data in PANDORA

The DELTA Heading

In PANDORA the DELTA HEADING will be stored as a record in the DELTA_FILES file, the data in this file can be accessed via a data entry window (Figure 1). Each record in this file is given a unique identifier called the record key and this record key is used to determine the currently active DELTA data set. In each PANDORA data set there will be as many DELTA data sets as there are records in the DELTA_FILES file.



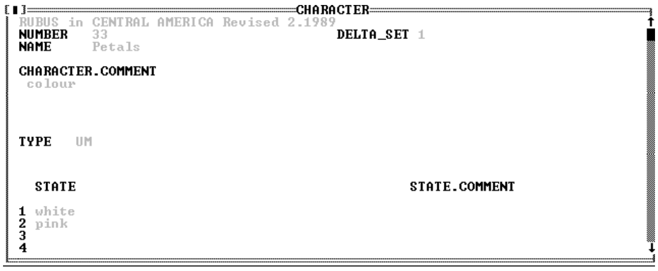
```
Number 1

DELTA_heading
RUBUS in CENTRAL AMERICA Revised 2.1989

DOS_directory     C:\TEMP\

DOS_file          RCA4.DAT

No_chars    48                    No_items    38

No_rules    14

Line_length 70
```

Figure 1. DELTA_FILES data entry window

Also included in each DELTA_FILES record are the title of the data set, the DOS path and filename of the input/ output data file, and the number of items characters and dependency rules defined in the original data set.

DELTA Characters in PANDORA

Character definitions are held in the CHARACTER file, accessed via the data entry window shown in Figure 2. Each DELTA data set defined in the DELTA_FILES file has its own character set comprising of a subset of the records within the CHARACTER file. In order to uniquely identify each character within the CHARACTER file, it is necessary to use two record keys, one to indicate which DELTA data set a character belongs to, the other to uniquely label the character within the data set. The DELTA data set identifier is the record key of the appropriate record in the DELTA_FILES file.



```
CHARACTER
RUBUS in CENTRAL AMERICA Revised 2.1989
NUMBER    33                                   DELTA_SET 1
NAME      Petals

CHARACTER.COMMENT
 colour


TYPE   UM


  STATE                                  STATE.COMMENT

1 white
2 pink
3
4
```

Figure 2. The DELTA CHARACTER data entry window

Each character defined in the CHARACTER file is assigned a name and , if required, a comment. The combination of the Number, Name and Character comment fields of the CHARACTER file are equivalent to the feature description section of a character description in a standard DELTA file. The CHARACTER record shown in Figure 2 would be represented as:

#33. Petals <colour>/

in a standard DELTA data file.

In a standard DELTA data file, the character type is normally stored under the *CHARACTER TYPES directive in the DELTA HEADING. In PANDORA, however, it is more convenient to store this information with each character. The character type is therefore stored in a single field of each CHARACTER record. All of the seven character types recognized in the DELTA format can be used, and when exported the *CHARACTER TYPES directive of the DELTA HEADING is reconstructed by analyzing each character as it is exported.

The possible states of a character are defined in two multi-value fields of the CHARACTER file: STATE and STATE_COMMENT. The name/value of each state is stored in a separate subfield of the STATE field and any comment associated with the state is stored in the appropriate subfield of the STATE_COMMENT field. When importing or exporting data from standard DELTA files, the order of the character states is preserved. The combination of the STATE and the STATE_COMMENT field is equivalent to the state descriptions section of a character definition and the CHARACTER record shown in Figure 2 would be equivalent to :

1. white/
2. pink/

in a standard DELTA data file.

The entire character description shown in Figure 2 is therefore equivalent to:

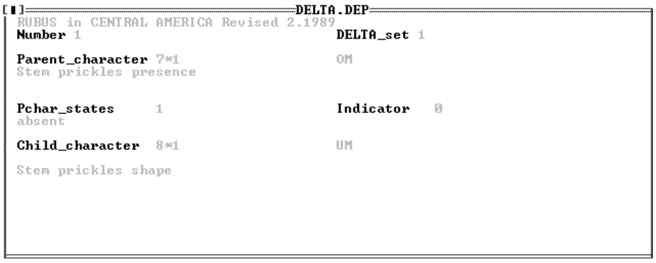
#33. Petals <colour>/
1. white/
2. pink/

and would have a *CHARACTER TYPES entry of :

33,UM

Character dependencies in PANDORA

In a standard DELTA data set the CHARACTER dependencies are defined in the DELTA HEADING under the *DEPENDENT CHARACTERS directive. When this information is imported into PANDORA each dependency is stored as a separate record in the DELTA_DEP file. This file is accessed via the DELTA.DEP data entry window shown in Figure 3.



```
DELTA.DEP
RUBUS in CENTRAL AMERICA Revised 2.1989
Number 1                                  DELTA_set 1

Parent_character 7*1                      OM
Stem prickles presence


Pchar_states      1                       Indicator   0
absent

Child_character   8*1                     UM

Stem prickles shape
```

Figure 3. The DELTA DEPendency data entry window

As in the CHARACTER file each record in the DELTA_DEP file has two record keys, one to identify which DELTA data set the character dependency definition applies to and the other to uniquely identify the current dependency rule within the current data set.

The parent and child (controlling and dependent) characters can only be selected from the PANDORA DELTA data set to which the current dependency rule belongs. In Figure 3, the current data set is 1 therefore only characters in DELTA set 1 can be selected.

The record key from the appropriate record in the CHARACTER file is placed in the appropriate field in the DELTA_DEP record when a character is selected. This record key will consist of two numbers separated by an asterisk; the second number of the record key identifies the PANDORA DELTA data set from which the character was taken and the first number identifies the character within that data set. The record in Figure 3 shows character 7*1 (Stem prickles presence) in the Parent_chracter field, 1 (absent) in the Pchar_states field and character 8*1 (Stem prickles shape) in the Child_character field. This record therefore states that character 8 of data set 1 is dependent upon character 7 of data set 1 and that character 7 cannot be used if character 8 is in state 1 in a taxon description.

This is equivalent to the following dependency definition from a standard DELTA file:

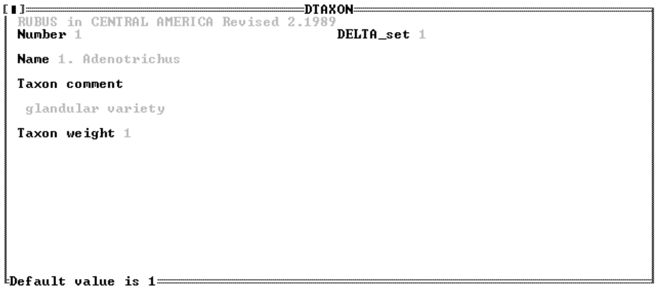
*DEPENDENT CHARACTERS 7,1:8

DELTA items in PANDORA

Data about DELTA items are held in 4 tables in PANDORA: DELTA_ITEMS, DTAXON, DELTA_MATRIX, and DELTA_COMMENT.

DELTA_ITEMS and DTAXON perform a similar role, and information equivalent to that held in the item name section of a standard DELTA taxon description is held in these files.

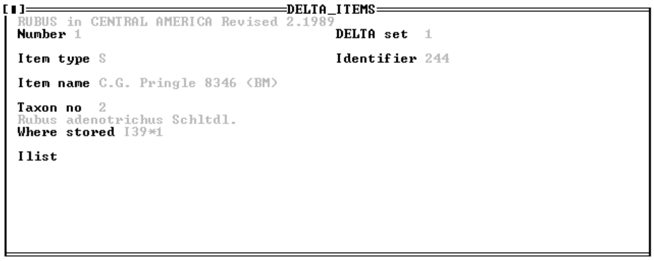
The DTAXON file holds information about DELTA items imported from standard DELTA data files. The name of the taxon associated with the each imported item , any comments about the item and the item weight are stored in each DTAXON record. All data from an imported taxon description is stored as is, and at present no attempt is made to link this data into the rest of PANDORA. For example, the taxon name in an item description is not linked to the nomenclature file (PTAXON) in the main part of the PANDORA system even though a record of that name may exist there. The same also applies to any characters describing bibliographic or geographical information. The DTAXON record shown in Figure 5 holds information about an ITEM called *Adenotrichus* in data set 1 (RUBUS in CENTRAL AMERICA) with a taxon weight of one.



```
[▮]================================DTAXON================================
 RUBUS in CENTRAL AMERICA Revised 2.1989
 Number 1                                   DELTA_set 1

 Name 1. Adenotrichus

 Taxon comment

  glandular variety

 Taxon weight 1









Default value is 1======================================================
```

Figure 4. The Dtaxon data entry window

The DELTA_ITEMS file holds information about DELTA descriptions that have been entered using PANDORA and which refer to taxa or literature references that are stored in the PTAXON, DESCRIP or RECORDS files of the PANDORA system. Data in this file in contrast to data in the DTAXON file is fully integrated into the PANDORA system, and details of synonymy, distribution etc. can be interrogated using the standard data base query tools.



```
[▮]==============================DELTA_ITEMS=============================
 RUBUS in CENTRAL AMERICA Revised 2.1989
 Number 1                                   DELTA set  1

 Item type S                                Identifier 244

 Item name C.G. Pringle 8346 (BM)

 Taxon no  2
 Rubus adenotrichus Schltdl.
 Where stored I39*1

 Ilist
```

Figure 5. The DELTA_ITEMS data entry window

There are four categories of item that can be stored in DELTA_ITEMS. These are:

1) Type S: Specimens whose characters have been encoded. This category will be used in reference to a specimen held in the RECORDS file. The record key of the appropriate specimen will be held in the IDENTIFIER field of each S type DELTA_ITEMS record. The name of the taxon will be the obtained from the latest determination of the specimen, this information is held in the DETS file. The record key defining this name will be obtained from the PTAXON file and will be held in the Taxon no. field of each S type DELTA_ITEMS record. In another field called Item name, a non-taxonomic name can be given to the item and should be used to identify the specimen used, e.g. collector's name plus the number assigned to the specimen by the collector.

2) Type D: Taxon descriptions derived from published material. This will be used in reference to a published description detailed in the DESCRIP file. The record key of the appropriate DESCRIP record will be held in the IDENTIFIER field of the DELTA_ITEMS table. The name of the taxon should be the same as in the DESCRIP record, and will be obtained from the PTAXON file and stored in the Taxon No. field. In the Item name field a non-taxonomic name should be given to the item that can be used identify the reference from which the description was taken.

3) Type E: Taxon descriptions based on neither

specimens nor published literature (These descriptions will usually be incomplete and restricted to a few characters). When used in the creation of a T type DELTA_ITEM (see below), any character states found in this type of item will override the states found in types S and D. For this reason this category has been called a taxon edit and its purpose is to allow a txonomist to make arbitrary changes in item descriptions.

4) Type T: A taxon description created by combining information from taxa and specimens in the same PANDORA DELTA data set.

The character state information relating to an item defined either in DTAXON or DELTA_ITEMS is stored in a record in the DELTA_MATRIX file. One taxon description will be held in each record of the DELTA_MATRIX and each character will be assigned a separate field. The order of characters in the DELTA_MATRIX file correspond directly with the order in which character records are defined in the CHARACTER file. I.e. the state information referring to character n in the CHARACTER file will be stored in field n of the DELTA_MATRIX records of the current PANDORA DELTA data set. There may be one or more states held in each character field of the DELTA_MATRIX. Any comments associated with a character state will be held in an equivalent record and field in the COMMENTS_MATRIX file.

The relationship between DTAXON, DELTA_ITEMS and the rest of PANDORA is summarized in Figure 6

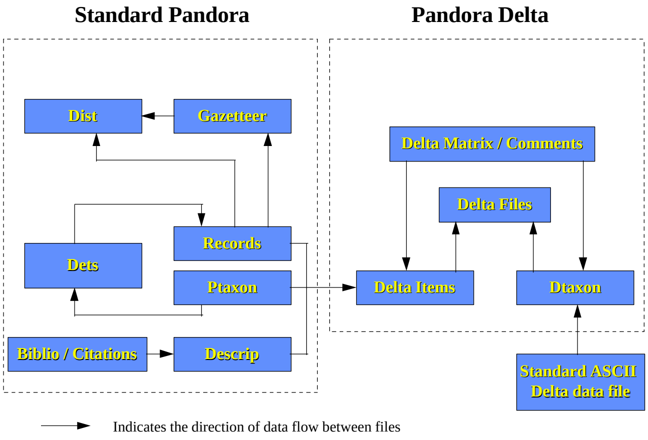


Figure 6. The relationship between PANDORA DELTA files and the rest of PANDORA

Searching, Editing and Creating DELTA data in PANDORA

As well as importing data from existing DELTA files, data can be entered directly into the CHARACTER, DELTA_DEP and DELTA_TAXON files using the data entry screens described above. However, to edit item descriptions special code has been written in order to control access to the DELTA_MATRIX file. The operations involved in editing the DELTA_MATRIX are described below.

Selecting the DELTA data sets for searching and editing

As stated earlier it is possible to store multiple DELTA data sets in the same PANDORA data set. Each of these DELTA data sets will have its own set of characters, dependency rules and item descriptions. There is, however, only one file each for characters, rules and items for every PANDORA data set. Therefore the information for each DELTA data set within a PANDORA data set will be mixed together.

This presents a problem when it comes to interrogating/ editing the DELTA matrix because the field definitions in the data matrix will change from data set to data set. For example character 1 in data set 1 may be petal colour whilst in data set 2 character 1 could be stem length. AREV allows this kind of data mixing because it applies no data typing to fields and hence any sort of data can be stored in any location in a file, even to the extent that data can be stored in fields that have no dictionary definition.

To overcome this problem and allow data entry or interrogation in the DELTA section of the database it is necessary to create a temporary view of the DELTA MATRIX that corresponds to the character set of the current DELTA data set. This is achieved by using the records in the CHARACTER file which define the character set for the current DELTA data set as a template for the field definitions in the dictionary of a file called PLANTS. For example, if character n of data set m (corresponding to record n*m in the CHARACTER file) is petal colour, then the name of field n in the dictionary of the PLANTS file would be changed to petal colour when data set m is selected. Each of the fields thus created are defined as symbolic fields, that is the data content of the field is calculated by applying a formula stored in the field definition. The formula for each field will be modified so that it points to the appropriate data field in the DELTA_MATRIX file.

Having established the dictionary it is then necessary to update the data portion of the PLANTS file so that there is one record in the plants file for every record in the DELTA_MATRIX file that is part of the currently selected DELTA data set.

The data is then in a form that can be accessed via standard data base query instructions. For example if it is known that petal colour is a character in the current data set it would now be possible to select all records in PLANTS in which petal colour is pink, as petal colour is a now defined as a field in PLANTS and pink is a possible value of petal colour.

Editing the DELTA_MATRIX

Description editing is achieved through a program that searches the CHARACTERS file for all the characters defined in the current DELTA data set. Each CHARACTER definition will map onto a field in the DELTA_MATRIX file according to the character number, ie character 1 will map onto field 1 of DELTA_MATRIX records. The user is able to select any of the items defined in the current data set and, through use of a series of popup windows listing the characters and their states, will be able to modify the character states assigned to the ITEM. New ITEMS can be created by first creating an item definition record in the DELTA_ITEMS file and then calling the item editor to enter the character states for the item. Such new DELTA_ITEMS will be assigned type S, D or E.

Creating composite taxa

Composite taxa are taxa whose characters and states are derived by combining data from one or more of the possible DELTA description types stored in PANDORA, i.e. specimen/ literature based descriptions and also imported DELTA descriptions of the taxon.

The taxon description created will have a definition record in the DELTA_ITEMS file. This record will be assigned type ‘T’. As such descriptions refer to taxa, there will be a record key from the PTAXON file pointing to the definition of the taxon name. This record key will be stored in the Taxon No field of the DELTA_ITEMS record. The new DELTA_ITEM will also have an associated DELTA matrix record with the character states of each of the component items combined into a single description.

A list of the ITEMS used in creating the new item is stored in the ILIST field of the DELTA_ITEMS record defining each type T item. This list can be updated at any time so that new items can be added to it or erroneously included item can be removed. After such changes the record in the DELTA_MATRIX will be updated to reflect the changes in the character state information.

Exporting DELTA data

The contents of the DELTA matrix can be exported into a standard ASCII DELTA file. The data can be either exported into a single DELTA or it can be exported into three separate files, for heading (SPECS), characters (CHARS) and taxa (ITEMS).

The DELTA HEADING will be derived from the appropriate definition record in the DELTA_FILES file, and by default the output file will be the same as the original input file. The *DEPENDENT CHARACTERS directive will be built up by combining all the records that belong to the selected DELTA data set in the DELTA_DEP file. Similarly the *CHARACTER TYPES directive will be formed by extracting the character type data from each of the records in the CHARACTER file the belong to the current data set.

All the CHARACTER definitions will be exported for the appropriate character records into the CHARACTER DESCRIPTIONS section of the output file and the relevant records from the DELTA matrix file will be exported into the Item Descriptions section of the output file. Although DELTA data sets often only contain descriptions of taxa, the export routine will allow export of any of the description types found in the DELTA_ITEMS and DELTA_TAXA files into an ASCII data set.

Conclusion

This article has briefly described the DELTA features incorporated into PANDORA. The benefits of incorporating the DELTA format into a data base system have been demonstrated and some of the technical difficulties in implementing this have been described. The work described here is, however, only the first stage of on ongoing project to more fully integrate DELTA and PANDORA. If you wish to assess the DELTA features of PANDORA and possibly comment upon desirable future developments that could be incorporated, there is a free PANDORA demonstration system available from the authors on 3½” floppy diskettes, via anonymous FTP at:

ftp.rbge.org.uk in directory /public/PANDORA

or via the World Wide Web at:

http://www.rbge.org.uk/science/pandora.home Δ

The ALICE Database System: recent news

R. Allkin

Alice is a suite of programs for building and managing relational databases which contain information about living organisms. For those who know nothing of Alice, the Introduction below summarises what it can do whilst subsequent paragraphs describe recent developments.

It has been some while since readers of the DELTA newsletter have heard from us. During the last two years we have been very active, owing largely to an expansion of our team to include two new full-time programmers and part-time support staff, and have seen rapid development. We have also contracted out production of our user documentation. Third-party programs, developed by colleagues elsewhere, have become available for displaying images or distribution maps and generating HTML pages for publication of data on the World Wide Web.

Introduction

Any Alice database may include common and Latin names, various classes of synonym, geographical distribution at three user-defined levels, habitats, economic importance, notes and free text. Users may define any number of their own DELTA-like descriptors to record ecological, morphological or biotechnological observations or record administrative information about taxa. Any fact about a taxon can be linked to one or more citations.

Alice is robust, easily used by biologists and does not require sophisticated equipment. Data quality, data security and the ability to exchange data readily are important to scientists. Alice is designed to enforce rules that prevent the entry of illogical or inconsistent data. Alice has excellent data export facilities. Data can be passed to other programs (such as ***Confor***), other databases and to future generations of software. Data can be imported into an Alice database from other sources such as data sets exported from DELTA or the Brahms herbarium management system.

Main developments

The main developments have been to enrich the Alice data structure, thereby increasing the scope and functionality of the system, and to develop migration tools which allow existing users of Alice to take advantage of this new functionality.

The first significant change has been to make Alice “Network-aware”. This means that programs can be used absolutely securely within a Windows environment and that users can share databases on a network server. The second significant development has been a broadening of the taxonomic ranks which Alice supports (thereby increasing its appeal to a wider audience including horticulturists and gene-banks) and adopting a data structure that allows more flexibility in the categories of name (preferred, synonym, basionym, misapplied) and class of taxon (accepted, provisional, unidentified). The data model has been further generalised to make management of the geographic hierarchy more flexible and to add classes of descriptor to help those users with large numbers of morphological or ecological descriptors. Other ideas are now being considered, such as defining different classes of citation or data source.

These changes are being progressively introduced into the Alice system. Further development of Alice System 2.0 has now ceased as our efforts centre on supporting Alice System 2.1 and development of Alice System 3.0.

The new systems

Alice System 2.0 is a mature and stable system. Two new systems, 2.1 and 3.0, with enriched data structures and extended functionality are being developed. Migration tools for moving databases between systems are provided.

Alice System 2.1 is due for release in March 1996, is “Network-aware” and has significant functional and performance advantages, all of which derive from the improved data structure. System 2.1 comprises new versions, or replacements, of all System 2.0 programs and several new programs which are described below.

Alice System 3.0 is based upon a yet more sophisticated data structure. In particular for the taxonomic hierarchy and nomenclatural detail, and for handling DELTA-like descriptors. The principal data capture and editor program (***Ace)*** will be released for Beta-testing during April 1996.

New Programs available with System 2.1

Ace The new data capture program enables database authors to design, build, edit and explore their databases. It replaces the System 2.0 programs ***Alice*** and ***Nview*** while offering considerably more functionality. ***Ace*** will, for example, allow far greater flexibility when searching for taxa or editing scientific names.

Aview A program to allow those other than the database author to explore and use Alice databases in read-only mode.

Dbupdate A tool for users to convert their databases for use with System 2.1 or 3.0.

Dbunlock A tool by which users can unlock their own databases or programs following security or system violations in a windows or network environment.

DbAdmin A tool for the manager of an Alice database server to control access and permit database/application locking within a multi-user environment such as an institutional network.

AliceWeb A third-party tool for those generating HTML pages from Alice databases to publish their data electronically.

Links to DELTA

Two programs support data exchange between Alice and other databases and programs: ***Alex*** (for exporting data) & ***Sam*** (for importing data). It is now possible to pass data from Alice databasesto DELTA format and vice versa.

Alex allows users to write out any portion of their database (any subset of the types of data included and any subset of the species) into a number of standard data formats to take advantage of other programs or to send colleagues data. One of the data formats offered is DELTA. Thus a user with a botanical database could, for example, create from scratch a new DELTA data set by selecting the descriptive and ecological data in his Alice database for just those taxa that are trees and found in a particular vegetation type.

Several of our users have taken advantage of this option to generate keys from descriptions stored in their Alice databases. Besides exporting the user-defined Alice descriptors into DELTA, users can also select to export other types of data (provided that these can be expressed in DELTA). Distribution data is interpreted as a series of DELTA descriptors (e.g. “Continents”, “Countries in Africa”, “States in the USA”). ‘Habitats’, ‘uses’ or even ‘higher taxa’ can be expressed as separate descriptors. The nomenclatural detail, synonyms, free text and bibliographic citations attached to individual facts within Alice databases, however, cannot be expressed within DELTA (except as comments) and, consequently, such information is effectively lost when exported. For a more detailed description please refer to table 1 below.

Sam reads data sets in the Alice Transfer Format (ATF) and imports them into Alice databases whilst undertaking the hundreds of data integrity checks Alice carries out when data are entered interactively. In this way ***Sam*** can be used to create new Alice databases from existing data sets, or merge data into existing Alice databases. ATF data sets (in ASCII, dBASE, or XDF format) may include species descriptions and any other type of data normally stored in Alice databases. A document with the formal definition of ATF can be obtained from the authors.

Mike Dallwitz has included a “Translate to Alice format” option in ***Confor*** making it possible to feed DELTA data sets into new or existing Alice databases. We know of only one user who has done this so far and would like to know of any others who have either tried or would like to explore this option. We know of some modifications we would like to see to the ***Confor*** filter and are accumulating comments before submitting a request to Mike.

While DELTA descriptive data can be transferred quite easily to Alice databases there are two awkward areas. These relate to the scientific names (which are held in much more detail within Alice databases) and continuous descriptors which can only be passed to Alice databases through use of a “Key Characters Directive”.

Our immediate plans include:-

1. the addition of a query history mechanism to allow users to store, re-use and edit commonly asked and potentially complex queries;

2. the inclusion of new search operators such as “endemic”, “unique”, “less than” and “more than”;

3. the inclusion of new classes of user-defined descriptor such as continuous measurements.

Table 1. Alice to DELTA data transfer

When exporting data into DELTA format, ***Alex*** creates three separate files: “DELTA.SPC”, “DELTA.CHR” and “DELTA.ITM”. It is possible to include in a DELTA data set most of the categories of data stored within Alice databases and these are interpreted as described in the following table. Free text, nomenclatural data and bibliographic data are not exported into DELTA.

What was in the ALICE database	**What this becomes in DELTA**
1) ‘n’ user-defined Alice descriptors (e.g. Flower colour, Fruit shape, Beak length etc.).	‘n’ DELTA descriptors
Ordered vs Unordered	Ordered vs Unordered
Exclusive vs Non-exclusive	Exclusive vs Non-exclusive
2) ‘n’ different categories of “Economic Importance” for each species	One DELTA descriptor called “Economic Importance” with ‘n’ states
(e.g. Medicinal, Wood, Ornamental)	
3) ‘n’ different categories of Habitat	One DELTA descriptor called “Habitat” with ‘n’ states
(e.g. Oak Woodland, Pine forest, Swamp)	
4) Geographical distribution	
‘n’ user-defined units (e.g. Africa, Asia, Australasia etc.) at the top Geographical level (e.g. “Continent”)	One DELTA descriptor (e.g. “Continent”) with ‘n’ states.
'm' user defined units (e.g. Ghana, Kenya, South Africa etc.) at the middle geographical level (e.g. “Country”) and each attached to one of the units at the higher level.	‘m’ DELTA descriptors (e.g. “Countries in Africa”, “Countries in Asia” etc.) each with the appropriate number of states (e.g. “Ghana”, “Kenya”, “South Africa” etc.)
'p' user defined units (e.g. Arizona, California, etc.) at the lower geographical level (e.g. “State”) and each attached to a single unit at the middle level.	‘p’ DELTA descriptors (e.g. “States in Kenya”, “States in the United States of America” etc.) each with the appropriate number of states (e.g. “Arizona”, “California”, “Texas” etc.)

NB: The DELTA character definitions created by ***Alex*** include only those states that are recorded for at least one species in the exported data set. Thus an Alice descriptor “Flower colour” with 10 different colour options would become a DELTA descriptor of only 4 colours should none of the taxa being exported ever show the other six colours.

To receive more information

If you wish to receive further information about the Alice system or copies of our own newsletter (now distributed free of charge about once a year) then please contact us by email at: *alice@dial.pipex.com*

consult our ‘Home page’ on the World Wide Web at: *http://www.presence.co.uk/alice*

or write by regular mail to:

Dr. R. Allkin (clearly labelled “for Alice Software”),
c/ Royal Botanic Gardens, Kew,
Richmond, Surrey, UK. TW9 3AB.

A SPANISH AND ENGLISH DELTA EDITOR: EDEL95

Elisa Bello, Antonio G.-Valdecasas & Juan R. Elvira

Museo Nacional de Ciencias Naturales
c/Jose Gutierrez Abascal, 2 28006-Madrid, Spain

EDEL is an editor for the primary DELTA files (CHARS, SPECS and ITEMS). It was written at the Museo Nacional de Ciencias Naturales (CSIC, Madrid Spain) for pedagogical purposes in introductory DELTA workshops (A. G.-Valdecasas et al, 1990). Many taxonomists, who were interested in the DELTA system, became discouraged in their first steps towards CONFOR because the format of three primary files of DELTA were rather rigid. Even more, for spanish speaking taxonomists the elaborate english vocabulary of DELTA was a challenge that many of them, being unfamiliar with computers, were not able to overcome. Our editor EDEL builds SPECS, CHARS and ITEMS files in a smooth way by writing only once the information that must be present in more than one file and doing counts and balances automatically. Many suggestions have been incorporated to earlier versions of EDEL, thanks to comments of students of several workshops held in Spain (1990, 1992), Portugal (1992) and Panama (1994).

A new version (v.2.11, 1995) of EDEL (EDEL95) is ready and available via anonymous ftp from:

albia.museo.csic.es (directory: /pub)

and via WWW from *http:\\albia.museo.csic.es\tools.html* (from the option DELTA in spanish)

Following below there is a short summary on EDEL. The building of new files through EDEL is by sequential menus. Each menu is self-explanatory and only in exceptional cases users will need to look for information not available in the program screens, e.g. the meaning of IMPLICIT.

After entering EDEL95, we can choose the language of the subsequent menus: English or Spanish are available at this time. The path of the directory where the files SPECS, CHARS and ITEMS will be saved, should be specified or the default location will be used; then the menu driven screen allows the following options (fig. 1):

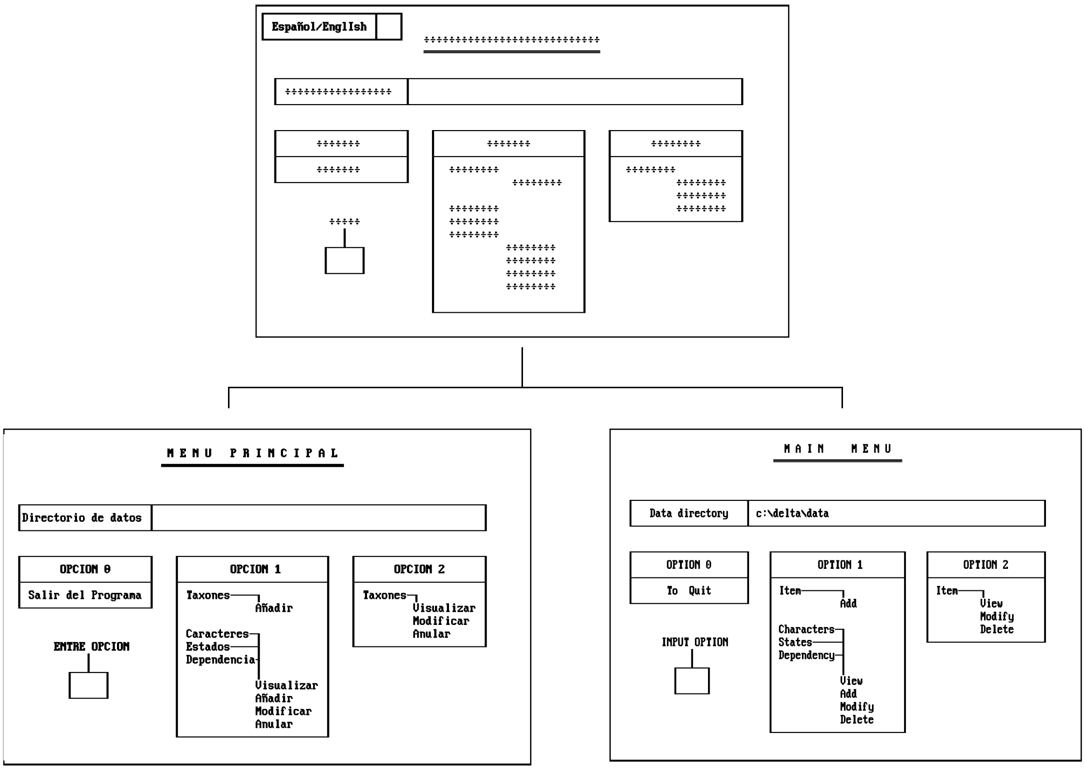
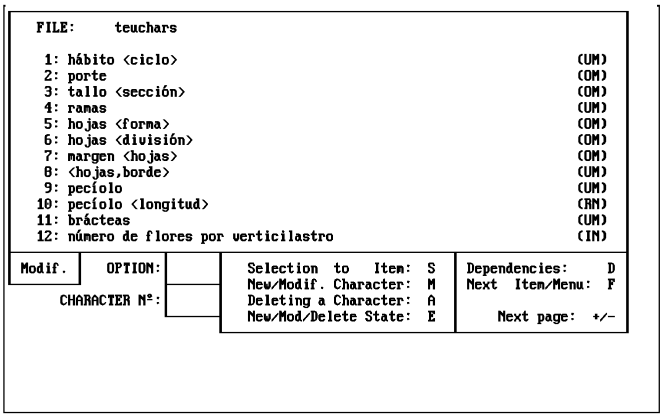


Figure 1.

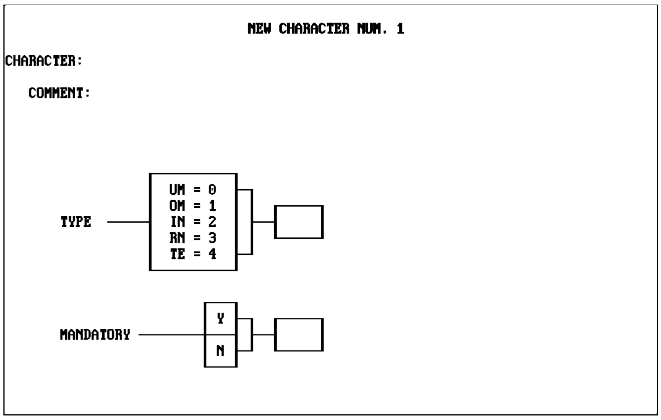
If we select add, append or delete character information or dependencies between them, the program will ask for the DELTA formatted basic file names and the text corresponding to the *SHOW directive (when files are new). Next, the editing character screen (Fig.2) will show:



```
FILE:     teuchars

 1: hábito <ciclo>                                        (UM)
 2: porte                                                 (OM)
 3: tallo <sección>                                       (OM)
 4: ramas                                                 (UM)
 5: hojas <forma>                                         (OM)
 6: hojas <división>                                      (OM)
 7: margen <hojas>                                        (OM)
 8: <hojas,borde>                                         (UM)
 9: pecíolo                                               (UM)
10: pecíolo <longitud>                                    (RN)
11: brácteas                                              (UM)
12: número de flores por verticilastro                    (IN)

Modif.   OPTION:          Selection  to  Item:  S    Dependencies:    D
                          New/Modif. Character:  M   Next  Item/Menu:  F
  CHARACTER Nº:           Deleting a Character:  A
                          New/Mod/Delete State:  E      Next page:  +/-
```

Figure 2.

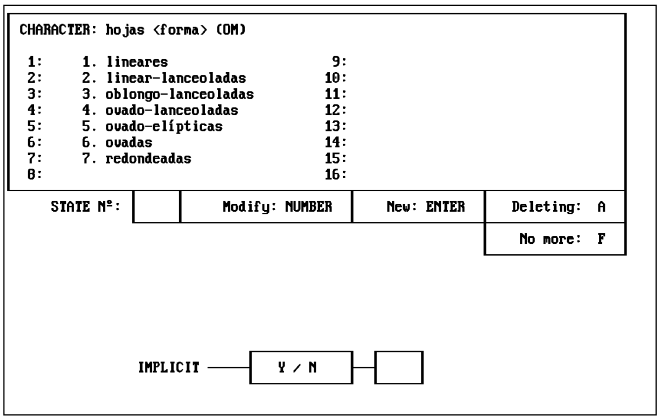
All kind of characters are displayed in the selection character screen. The option of obligatory character is available, as well as some controls against mistakes, at this step.



```
                    NEW CHARACTER NUM. 1
CHARACTER:
  COMMENT:

        TYPE ——— UM = 0
                 OM = 1
                 IN = 2
                 RN = 3
                 TE = 4

   MANDATORY ——— Y
                 N
```

Figure 3.

Once the character is defined, we change to the states of character screen.



```
CHARACTER: hojas <forma> (OM)

1:     1. lineares                   9:
2:     2. linear-lanceoladas        10:
3:     3. oblongo-lanceoladas       11:
4:     4. ovado-lanceoladas         12:
5:     5. ovado-elípticas           13:
6:     6. ovadas                    14:
7:     7. redondeadas               15:
8:                                  16:

  STATE Nº:      Modify: NUMBER    New: ENTER    Deleting:  A
                                                 No more:  F

            IMPLICIT ——— Y / N
```

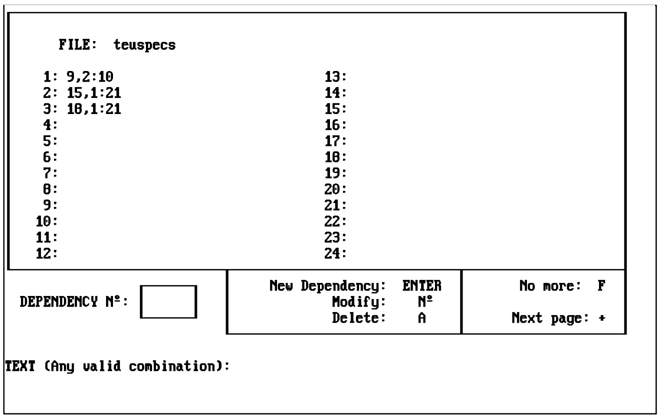
Figure 4.

This screen allows a maximum of 16 states definition per character. Real and integer numeric characters do not admit states in the CHARACTERS file.

For them, you can introduce the measurement unit in this screen, e.g. mm IMPLICIT states should be indicated now. This screen allows us to modify an already defined state, by answering the question:

TEXT TO MODIFY:

We can establish dependencies between characters or modify them with option D of the character editing menu (Fig.2) In EDEL95 we can specify up to 48 dependencies, that must be introduced following the DELTA standard format, e.g. C1,s:C2

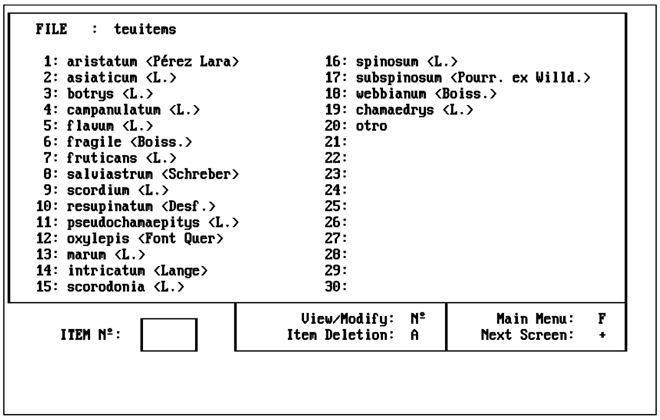


```
   FILE:  teuspecs

 1: 9,2:10                    13:
 2: 15,1:21                   14:
 3: 18,1:21                   15:
 4:                           16:
 5:                           17:
 6:                           18:
 7:                           19:
 8:                           20:
 9:                           21:
10:                           22:
11:                           23:
12:                           24:

                  New Dependency:  ENTER        No more:  F
DEPENDENCY Nº:            Modify:   Nº
                          Delete:   A         Next page:  +

TEXT (Any valid combination):
```

Figure 5.

The others options are: delete a previous codified dependency, display the next 24 dependencies in the screen and return to the characters editing screen.

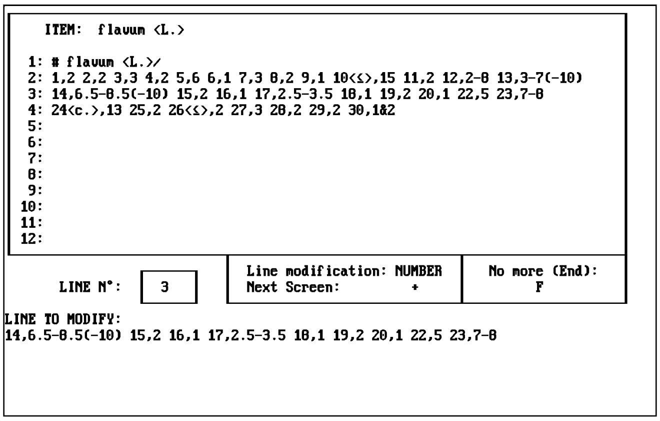
The ITEMS (option 2, Fig. 1) editing menu (Fig. 6) will ask the name of the taxon and additional COMMENTS. It should be written as we would like to see it in the final descriptions.



```
  FILE   :  teuitems

   1: aristatum <Pérez Lara>             16: spinosum <L.>
   2: asiaticum <L.>                     17: subspinosum <Pourr. ex Willd.>
   3: botrys <L.>                        18: webbianum <Boiss.>
   4: campanulatum <L.>                  19: chamaedrys <L.>
   5: flavum <L.>                        20: otro
   6: fragile <Boiss.>                   21:
   7: fruticans <L.>                     22:
   8: salviastrum <Schreber>             23:
   9: scordium <L.>                      24:
  10: resupinatum <Desf.>                25:
  11: pseudochamaepitys <L.>             26:
  12: oxylepis <Font Quer>               27:
  13: marum <L.>                         28:
  14: intricatum <Lange>                 29:
  15: scorodonia <L.>                    30:

                            View/Modify:  Nº      Main Menu:   F
     ITEM Nº:             Item Deletion:  A     Next Screen:   +
```

Figure 6.

The program moves now to the character editing screen and the option S is used to select characters and states for each taxon (Fig. 7). The selection is used for easy displaying of all edited characters and states. Any comment could be introduced at this time as well. The program allows us to define up to 60 taxa with 24 lines of characters/states/comments per item. Option 2 of the main menu is for the view, modify or delete items routine.



```
   ITEM:  flavum <L.>

 1: # flavum <L.>/
 2: 1,2 2,2 3,3 4,2 5,6 6,1 7,3 8,2 9,1 10<≤>,15 11,2 12,2-8 13,3-7(-10)
 3: 14,6.5-8.5(-10) 15,2 16,1 17,2.5-3.5 18,1 19,2 20,1 22,5 23,7-8
 4: 24<c.>,13 25,2 26<≤>,2 27,3 28,2 29,2 30,1&2
 5:
 6:
 7:
 8:
 9:
10:
11:
12:

                       Line modification: NUMBER     No more (End):
     LINE N°:    3     Next Screen:           +            F
LINE TO MODIFY:
14,6.5-8.5(-10) 15,2 16,1 17,2.5-3.5 18,1 19,2 20,1 22,5 23,7-8
```

Figure 7.

In any updating, backup files will be automatically created, and common information will be updated in all the concerned files just with a single entry. The program will keep an unmodified version of these files as *.BAK files.

More detailed information for beginners (both English and Spanish) is included with the program.

Some features about EDEL are:

- EDEL is freeware.
- Available throught Internet (FTP or WWW). See above for details.
- A single program is able to work with Spanish or English menus.
- Runs on any IBM compatible PC with 640 K RAM with a CGA card or better.
- There is no need for a hard disk.

Acknowledgements

We are grateful to Mike Dallwitz for his comments on this paper and Jose M. Becerra for his help and encouragement. This work is funded by CAM grant to E.B., and by the project PB92-0082.

References

Valdecasas, A.G.; Elvira, J.R.; Becerra, J.M. and Bello, E. 1990. 'Dallwitz's DELTA: CONFOR, KEY, DIST, TRANSTE'. Museo Nacional de Ciencias Naturales, Madrid (Spain). (Spanish User's Guide for the DELTA System).

∆ ∆ ∆ ∆

DELTA Registration and Order Form (March 1995)

Name: ____

Address: ____

____

____

____

Country: ____________________ Fax: ____________________

Email: ____

Please supply the following: $

____ Registration for all programs: single or first user $400 ________

____ Registration for all programs: additional users @ $100 ________

____ Registration for INTKEY only: single or first user $200 ________

____ Registration for INTKEY only: additional users @ $50 ________

____ Sets of disks (all programs) @ $30 ___ 3.5" ___ 5.25" ________

____ User’s Guides @ $20 ________

____ Primers @ $10 ________

____ Postage and handling (disks and/or manuals) $20 ________

TOTAL (AU$ in Australia and New Zealand; US$ elsewhere) ________

____ I enclose a cheque (payable to CSIRO Collector of Moneys)

____ Please debit my: Bankcard (Australia and NZ) / Mastercard / Visa

Card number: ________ ________ ________ ________ Expiry date: ____/____

Signature: ______________________________

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

Multi-user Registration

Name of Organization: ____

Names of Registered Users:

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

______________________________ ______________________________

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

To subscribe to the DELTA Newsletter, please complete the form below and return to the editors.

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

Name ____

Address ____

____

____

____

Country ____

Email address _____________________________________

Research interest / Taxonomic speciality ________________

____

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

Signed ______________________ Date __________