Hawaiian Natural Resources Monitoring Database

01 June 1996

Prepared by Philip A. Thomas

ACKNOWLEDGEMENTS

For support, guidance, and feedback during this project--and without whose efforts this project would not exist--I wish to thank Dr. Clifford W. Smith; Dr. Lloyd L. Loope; Robert Teytaud; everyone at The Nature Conservancy of Hawaii (particularly Guy Hughes [Maui], Coleen Cory [Oahu], and Roy Kam [Oahu], Debbie Roberts [Oahu], and Joan Yoshioka [Molokai]); personnel at the Big Island NARS office (Julie Leialoha, Bill Stormont); Randy Bartlett (Maui Pineapple Co.); the East Maui Watershed Partnership; the folks at the NBS/Haleakala field office; and all others who have played a part in making this happen. Particularly, for food for the body and mind while on Oahu, I am indebted to the Gordon Biersch Ecological & Philosophical Consortium (you know who you are!).

CONTENTS

CHAPTER 1. Monitoring Database overview

This chapter provides an overview of the Monitoring Database.

a. What is the Monitoring Database?

The Hawaiian Natural Resources Monitoring Database[1] is a software package designed as a tool for data entry and analysis for resource monitoring[2] by land managers in Hawaii. The Monitoring Database is designed in Paradox for Windows, chosen because of its combination of ease of use for end users AND flexibility of custom design (programming) capabilities.

b. Purpose of the Monitoring Database

The purpose of the Monitoring Database is to facilitate standardized and fully-documented data collection efforts by federal, state, and private agencies. Data collected by various agencies in different areas will then be comparable,[3] enhancing the value of each agency's work by allowing them to compare their data to similar data collected by other agencies. This comparability will allow a "big picture" approach to analysis of this type of scientific data never before possible in Hawaii.

c. History and Evolution of the Monitoring Database

The first precursor to the Monitoring Database was an idea to put into a database information on all plant (and invertebrate) taxa in Haleakala National Park (Maui). Information was converted from word processing documents and a database was formed.[4] Other related offshoot projects include a taxon-linked bibliographic tracking database, an alien species database ("Harmful Non-Indigenous Species" [HNIS]),[5] a database of plant pathogens in Hawaii,[6]

---

1. The Hawaiian Natural Resources Monitoring Database will hereinafter be referred to as simply "the Monitoring Database."

2. Monitoring in this sense includes collection of baseline information and information used to detect change in vegetation cover, disturbance by feral animals, etc.

3. assuming that standard field methods are followed

4. Ironically, the master list of this information is still maintained in word processing format.

5. This database is currently being developed and enhanced by the Research Corporation of the University of Hawaii funded by the National Biological Service "Hawaiian Ecosystems at Risk" project.

a database used for tracking the Federal Endangered/Threatened status of Hawaiian species,[7] and a prototyped, soon-to-be-developed database for tracking feral animal control efforts.[8] The main infrastructure common to all these databases (taxon information) was designed to be compatible with that of the Botany Department of the Bernice P. Bishop Museum (Honolulu).[9] (The museum's data structures are based on international biological database standards.)

The actualization of the Monitoring Database was sparked by a need of Guy Hughes (then with The Nature Conservancy of Hawaii [TNCH] [Maui]) to analyze data and incorporate information and provide graphic output of complex analyses.[10] Hughes' field methods were modified versions of those set forth in a document[11] compiled by Pat Dunn[12]. Recently, the idea of creation of a standard monitoring protocol was proposed at a meeting[13] of the East Maui Watershed Partnership[14] (EMWP) for the purpose of gathering data in the geographic region with which EMWP is concerned. Since then, TNCH has used the Monitoring Database to incorporate data from tests (for EMWP) in Waikamoi of new, somewhat modified field methods. Eventually, the methods that are settled upon based on these trials will be proposed as statewide

6. This database is being populated and used by the Botany Department of the University of Hawaii (Manoa).

7. The culmination of this project is the database currently used by the U.S. Fish & Wildlife Service in Honolulu.

8. See footnote 5.

9. Bernice P. Bishop Museum, 1525 Bernice St., P.O. Box 15000A, Honolulu, Hawaii 96717; voice (808) 847-3511.

10. Hughes, Guy D. 1995. Long-Term Biological Threat and Resource Monitoring, Waikamoi Preserve, East Maui, Hawaii, 1994. The Nature Conservancy of Hawaii. Honolulu, Hawaii.

11. Dunn, Patrick. 1992. Long-Term Biological Resource and Threat Monitoring of Hawaiian Natural Areas. Produced for the Hawaii Department of Land and Naural Resources, Division of Forestry and Wildlife by The Nature Conservancy of Hawaii. Honolulu, HI.

12. Pat Dunn was formerly a stewardship ecologist with The Nature Conservancy of Hawaii.

13. [...{ask Nancy Glover?} details introduction of standard monitoring protocol at first big EMWP meeting]

14. [put a description of EMWP here...]

standards for situations to which each respective method is applicable.[15] [<---have this entire paragraph reviewed by cliff...]

d. Where to get more/updated information

For the latest news about the Monitoring Database, subscribe to HIMONDB-L, the Hawaii Natural Resources Monitoring Database user's list (see section 1.f for details). Monitoring Database information, news, updates, and current and historical versions of the documentation are available for download from the worldwide web.[16]

e. How can I obtain a copy of the Monitoring Database?

Licenses to use the Monitoring Database are available free of charge[17] to qualifying agencies, organizations, businesses, private landowners, educational institutions, and individuals. Use of the Monitoring Database by a wide range of audiences is encouraged.

Licenses granted are licenses for USE of the Monitoring Database software; they do NOT transfer ownership of the software. The software may not be sold or redistributed in part or in whole except as explicitly detailed in the license agreement. One of the main reasons for this is to ensure that all users of the software are known to the Monitoring Database System Coordinator, so users can be apprised of updates to the software.

f. What about technical support?

[now rcuh] [if you help we'll trade] [pdoxwin support from borland & 3rd party] [ask & we'll give references for training, support, etc.] It is hoped that users will share their expertise with users in their own and other organizations, as well as participate in the evolution and development of the system by providing suggestions to and

15. [?include somewhere as a footnote or aside: methods SHOULD be designed based on expected analysis methods; hasn't been done like this, or at least I'm not aware of documentation to that effect; think this should be addressed by those who (should) address this kind of thing...]

16. http://www2.hawaii.edu/~halesci/MonitoringDatabase

17. Though no run-time version will be distributed, the low cost of the full Paradox for Windows (PDOXWIN) software--under $150--is not anticipated to be a major hurdle for users of the Monitoring Database. It is the responsibility of the user to have their own legal copy of PDOXWIN; PDOXWIN will not be distributed with the Monitoring Database.

working with the the Monitoring Database Project Coordinator via each organization's Database Administrator.

To ease the time burden on the Monitoring Database Project Coordinator, ONLY an organization's Database Administrator should contact the Monitoring Database Project Coordinator directly with PDOXWIN/Monitoring Database questions. Other users should first attempt to resolve questions from within their own organization (ask your Database Administrator or other designated in-house support person). If the problem cannot be resolved in this way, the fully-apprised Database Administrator should then contact the Monitoring Database Project Coordinator for support.

Additionally, an internet mailing list is available for users to receive the latest information about the Monitoring Database, as well as to ask and answer questions, provide and receive insights and tips, and have discussions with other users. Subscriptions to the list may be requested by sending e-mail to LISTPROC@HAWAII.EDU with a BLANK subject line, and the contents of the message being "SUBSCRIBE HIMONDB-L your name" (no quotes) from the e-mail account to be subscribed. After subscribing, you can send correspondence to the group at HIMONDB-L@HAWAII.EDU.[18]

Support for Paradox for Windows for questions not directly related to use of the customized portion of the Monitoring Database is available direct from Borland. Installation and configuration support is free at (408) 461-9166. Free automated support is available at (800) 524-8420. Pay-for-support for non-installation issues is available at (800) 523-7070, or via a 900 number (the 800# operator can supply you with this information).

18. Particularly if you are not familiar with internet lists, please READ AND KEEP the signon message you receive upon successful subscription; it contains valuable information regarding proper use of an internet list, as well as instructions for how to UNSUBSCRIBE from the list.

CHAPTER 2. Monitoring Database Terms and Concepts

This chapter, though basically a glossary, is placed near the beginning of the documentation instead of at the end because it is important for the reader to be at least somewhat familiar with the terms which will be used throughout this document. This section is designed to be used as a reference AND to be read through--if only skimmed--by all users of the Monitoring Database. It will be assumed from this point forward that the reader understands this section or will refer to it as necessary.

The terms presented here may be unfamiliar to the first-time reader, and are always used in very specific ways. Throughout the text in subsequent sections of this document, terms defined here are **boldfaced** for your convenience. Some concepts included here are not used elsewhere in the Monitoring Database documentation, but are relevant because understanding them will promote better overall understanding of the Monitoring Database.

For more detailed information about any topic relating to Paradox for Windows in general, refer to the Paradox for Windows manuals and/or the online help. If your questions are still unanswered, refer to section 1.f for other support options.

a. child table

The child table (in a **referential integrity** relationship) is the table containing fields whose values are validated against another table (the **parent table**); values of [a] specific field[s] in the **parent table** must already exist before those values can be entered in the linked field of the child table. When **referential integrity** is established, this relationship is permanently enforced. If only a **table lookup** relationship is established, a parent-child relationship is enforced *only* at the time of initial data entry (into the table which would be the child table *if* **referential integrity** were established); in this case, the value in what would be the **parent table** may change in the future independently of the value entered in the would-be child table. The terms "child value" and "child record" refer (respectively) to the value of a field or record of a table in a parent/child relationship. Child tables will always be BELOW **parent table**s in a **table relationship diagram**. ("**Dependent table**" and "slave table" are synonyms of "child table.")

b. core tables

"Core tables" refers to those tables (.DBs and related files) which were originally distributed with the Monitoring Database. Generally, these tables are all related using **referential integrity**, and are therefore functionally a single entity in some senses. The structure of core tables should NEVER be altered (see sections 3.b.i, 3.b.ii.(3)(c)).

c. database

There are (at least) two valid definitions of the term "database." Unless otherwise specified, the first of the following definitions is implied throughout this documentation.

i. database (1)

One definition of database is the computerized system as a whole, including forms, reports, etc. (the term "Monitoring Database" in this documentation will implies this sense of the word).

ii. database (2)

Another definition of database refers to the TABLES only and their integrated relationship. (**Table lookup** defines only a collection of related tables; the use of **referential integrity** "forces the issue" in this strict sense, because then the tables are no longer independent of one another. Microsoft has taken this concept so far in their Access database management software that the entire database [in the sense of the first definition] that each Access database comprises only one [huge] file [i.e. tables, forms, etc. are not discete DOS files as they are in PDOXWIN].)

d. dependent table

(See **child table**.)

e. field

[...] A field corresponds to a *column* in spreadsheet lingo.

f. form

A PDOXWIN form can be generally thought of as a "screenful of information." Forms are the interface between the database user

and the data stored in the database's tables; all menus of the Monitoring Database are also forms. Forms can display and/or manipulate data, and are typically used for data entry and display of graphical information to the user on the screen.

g. lookup table

(See **parent table**.)

h. master table

(See **parent table**.)

i. metadata

Metadata is information about other data. For example, metadata includes definitions of codes used to identify taxa, as well as descriptions and actual maps of plots and/or stations surveyed. Without appropriate and legible metadata, the value of a data set (and therefore the data collection effort as a whole) will be reduced greatly, perhaps making the entire data set meaningless. (Quite frankly, if metadata is not recorded in a coherent and retrievable fashion, you might as well not go to the trouble and expense to collect the data in the first place!)

j. Monitoring Database Personnel Groups

The following personnel groups comprise a complete functionally-oriented organizational structure for use and administration of the Monitoring Database:

- Monitoring Database Project Coordinator (System Designer)
- Database Administrator
- Programmer (within a user organization)
- Site Manager
- Data Entry
- Manager-user
- Internet User

These specific terms describing Monitoring Database Personnel Groups will be used throughout all system documentation based on the descriptions of each group in Chapter 3.c.ii.

k. normalization

Normalization of data refers to storing the data in a particular format which conforms to a relatively complex set of rules. For most purposes, all you need to know is that (1) normalized data is generally much easier to work with when querying data or generating reports; therefore (2) you should almost always use normalized tables upon which to base queries and reports (the Monitoring Database includes tools that allow you to create normalized tables from your field data[19] at the click of a button).

l. password security

Password security is used in the Monitoring Database to help ensure data integrity and consistency of data format among users of the software. Unless your site has been customized, password security should be virtually transparent to system users; in the standard distribution, running the Main Menu automatically provides password access to the system without user intervention.

It is likely that the only time non-programming users would ever encounter the password security feature in the standard distribution of the software would be when attempting to create new queries or reports AFTER having closed the Main Menu. In this case, the system will prompt the user for a password in order to access the tables. To remedy this situation, simply run the _MAINMNU form and leave it open while you're working.

m. parent table

The parent table (in a **referential integrity** relationship) is the table in which valid values must already exist before related information in a **child table** can be entered. When **referential integrity** is established, this relationship is permanently enforced. If only a **table lookup** relationship is established, a parent-child relationship is enforced *only* at the time of initial data entry (into the table which would be the **child table** *if* **referential integrity** were established); in this case, the value in what would be the parent table may change in the future independently of the value entered in

19. Although virtually all other tables in the Monitoring Database are stored as normalized tables, because of inherent difficulties in creating and maintaining data entry forms for completely normalized data in some cases, FIELDATA.DB and THRETMON.DB are *not* stored as normalized tables. Automated (pushbutton) procedures are provided as part of the Monitoring Database to normalize--as well as perform other interpretive functions--for all standard Data Set Types.

the would-be **child table**. Parent tables will always be ABOVE **child table**s in a **table relationship diagram**. ("**Master table**" is a synonym of "parent table.")

n. personnel groups

Personnel groups to which this document refers are functional definitions of personnel who are responsible for certain tasks with respect to the management of the Monitoring Database (see section 2.j and Chapter 3.c.ii for details).

o. query

A query is a question that you ask about information in the database. Queries allow you to summarize data (e.g. counts, sums, averages), calculate answers (using mathematical operations), and include only/exclude selected data. Paradox for Windows provides a "visual query-builder," making it easy for you to formulate and ask your own questions. Reports, graphs, and other output may be based on the answers to queries.

p. record

[...] A record corresponds to a *row* in spreadsheet lingo.

q. referential integrity

Referential integrity (RI) may be thought of as the "glue" that transforms a collection of related tables into a **database**[20] entity. Referential integrity enforces unbreakable links between data values of specified fields in related tables. In the Monitoring Database, referential integrity is always[?...] used in conjunction with **table lookup**. The two features perform related functions, the primary difference being that RI enforces data relationships PERMANENTLY, i.e. after initial data entry. Referential integrity performs some functions related to those of **table lookup**, but referential integrity is not the same as **table lookup**. Without referential integrity, the parent value can be deleted from the parent table after initial entry of a child value; one of the functions of referential integrity is to "keep up with" the fact that a parent value exists for *each* child value, and to perform tracking/housekeeping tasks to prevent parent values which have corresponding child

20. definition 2

values in some table from being deleted or changed in such a way as would otherwise "orphan" the child record. Referential integrity is the feature of Paradox for Windows which allows changes in the value of a field in a parent table to be automatically reflected in all related **child table**s.[21] One main difference in the functions of these features is that referential integrity is a two-way, permanent relationship (referential integrity is ALWAYS enforced; it is impossible for data to be "out of sync" between parent and **child table**s).[22] See **table lookup** for a comparison of features. Changes to values of **child table** fields used to establish this relationship cannot be changed to any values which would violate the relationship. Changes to values of **master table** fields used to establish the relationship either will be reflected in the values of the respective fields in related **child tables**, or will not be allowed at all (depending on options selected at the time of RI definition). For further information, consult the Paradox for Windows documentation. (NOTE: It is very important for Database Administrators and Programmers to understand the differences in table lookup and referential integrity; members of these personnel groups should seek further knowledge.)

r. report

A PDOXWIN report is output from the system, typically paper reports which are printed. Reports can contain detail, summary, or graphical information. Reports may be viewed on the screen, but are generally designed with hard copy in mind as the ultimate medium.

s. security access levels

[...]

t. SQL

21. In Paradox for Windows v.5 (and v.7), **referential integrity** is supported only "one level deep" (i.e. cascading update is not supported). See Paradox for Windows documentation for more information on this limitation.

22. For the Database Manager and Programmer: The permanent relationship created by enforcement of referential integrity is maintained in the .VAL files associated with BOTH the parent and **child table**s. See section 3.c.ii for IMPORTANT information regarding use of RI in the Monitoring Database.

u. table

v. table lookup

Table lookup is the feature which enforces existence of a key value in a **lookup table** for each record entered in what would be its **child table** (with respect to the current field[s]) in a referential integrity relationship. In the Monitoring Database, table lookup is almost always used in conjunction with **referential integrity**. Table lookup performs some functions related to those of referential integrity, but they are not the same (see 2.n). Table lookup is the feature which allows the user to peruse the would-be parent table (if **referential integrity** were established) and select a value from the list of values currently available in that would-be **parent table**. One main difference in the functions of these features is that table lookup is a one-way, temporary relationship (table lookup is only enforced at the time of data entry; subsequent changes to the master value are NOT* reflected in the **child table**). See **referential integrity** for a comparison of features. For further information, consult the Paradox for Windows documentation. (NOTE: It is critically important for Database Administrators and Programmers to understand the differences in table lookup and **referential integrity**; members of these personnel groups should seek further knowledge.)

w. table relationship diagram

A table relationship diagram is a graphical representation of the relationship among tables in a **database (1,2)**. It is often helpful to have access to the table relationship diagram of the database with which you are working when creating complex queries, reports, or forms which require related information from more than one table. Refer to section 3.a.i.(4) for an example of a table relationship diagram.

CHAPTER 3. Design of the Monitoring Database

[...]This section provides information to [...audience] [...RE: what?]

a. Overview of TMD Design

This section presents an overview of the design of the Monitoring Database, as well as insights into the reasons behind design decisions.

i. Tables and table relationships

Tables are the most basic and vital part of a **database (1,2)**. Each table is (conceptually if not actually) a file which contains multiple RECORDS of closely-related data. This section provides an introduction to the general relationships among tables in the Monitoring Database, as well as some background regarding why these relationships are as they are. (Except for having a cursory understanding of the hierarchical nature of the table relationships, most users will not need to understand the details of this section.)

(1) hierarchical table relationships

Tables in the Monitoring Database are arranged in a hierarchical manner, i.e. each table may have a relationship to another table (or tables) both "above" and "below" it in a table relationship diagram (see section 3.a.i.(4)). Tables "above" a given table are considered to be its **parent table**s; tables "below" a given table are considered to be its **child table**s. Certain rules are enforced between **parent table**s and **child table**s to provide reasonability checks during data entry (to ensure a certain level of accuracy), and to ensure that all data can be easily and correctly interpreted during analysis and presentation (i.e. to allow for capture of appropriate **metadata**).

(2) rdbms rules & normalization

[...briefly mention; refer to other documents for details]

Although completely normalized tables are usually the goal of good **database (1)** design (based on pure

relational theory), sometimes practicality dictates changes in this ideal. Because of user interface design and maintenance considerations, "raw" field data in the Monitoring Database is stored in tables (FIELDATA.DB and THRETMON.DB) which are not 100% normalized. These tables are keyed down to the taxon level, but multiple observed values for each taxon are stored in the same record.

(3) table lookups & referential integrity

Table lookup and **referential integrity** are terms which refer to specific sets of rules enforced between **parent table**s and **child table**s. In the Monitoring Database, full use is made of both these mechanisms. Generally, they are used together when defining table relationships. Refer to the definitions of each term in Chapter 2 and in PDOXWIN documentation for further details.

(4) table relationship diagram for the Monitoring Database

[...] Figure 1 is the table relationship diagram for the Monitoring Database.[23]

ii. Password Protection in the Monitoring Database

The Monitoring Database takes advantage of PDOXWIN's password protection feature. Currently, this protection is completely transparent to the user. The Monitoring Database System Designer is the only person who actually needs to understand the security system, but all users should be aware of its existence in case a security-related issue is encountered.

(1) Rationale for security

Security features are built in to the Monitoring Database to ensure the integrity of the system (by disallowing users to change the structure of core components the database), and to ensure consistency among installations. Enhanced security features available in

23. The **table relationship diagram** in Figure 0 was current at the time of publication of this documentation. Refer to the TABLEDOC form for the most up-to-date version. (From the Main Menu of the Monitoring Database software, choose [...button clicks].)

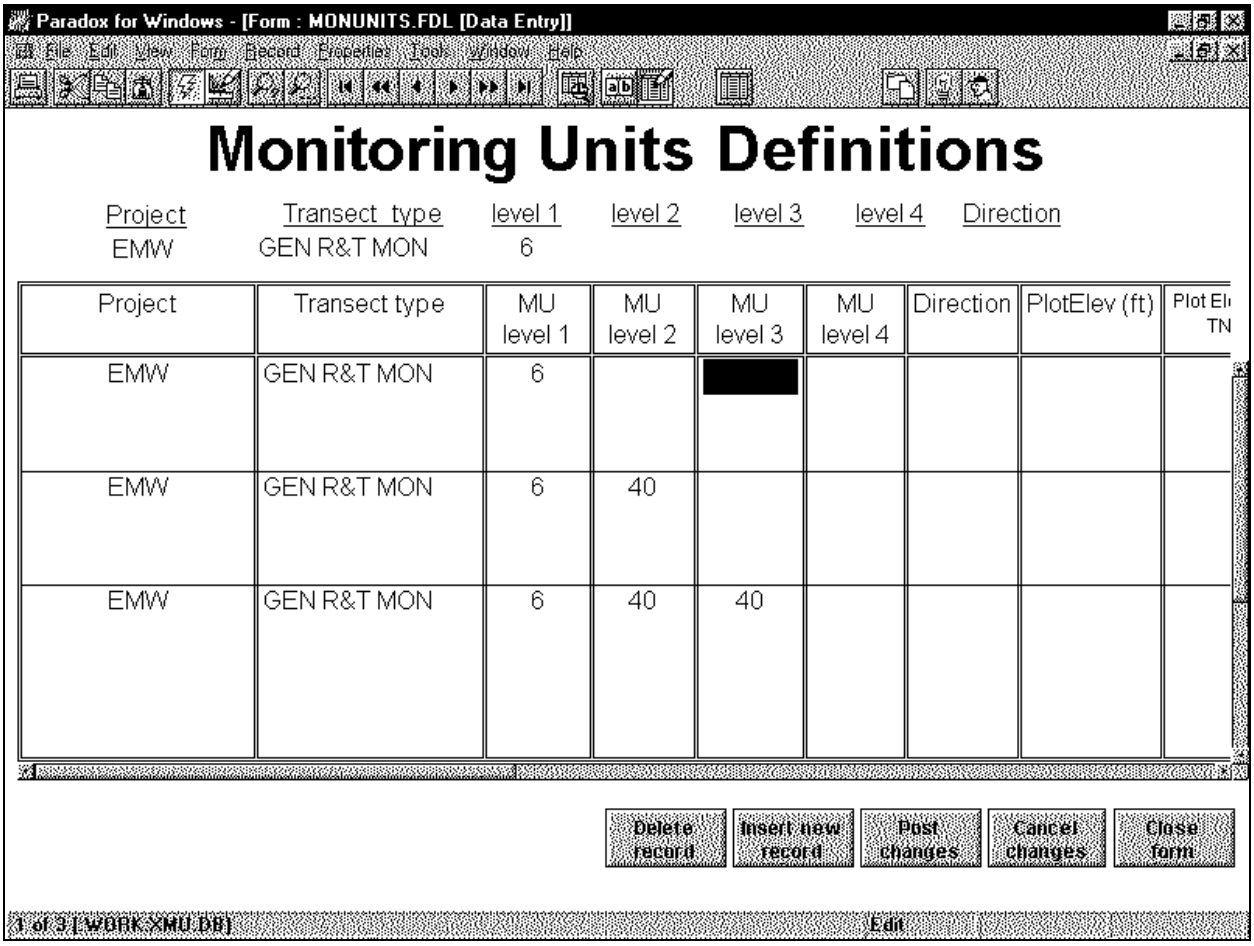

Figure 1

PDOXWIN (but not integrated into the initial release of the Monitoring Database[24]) allow for protection of sensitive data by ensuring that only authorized personnel can view and/or change data.

(2) Security levels

In the standard distribution of the Monitoring Database, there are two levels of security, those for: (1) all users, providing total access (read, write, insert, delete, update) to all data; and (2) the Monitoring Database System Designer, providing total access to all data AND allowing changes to table structure. The "all users" password is provided automatically by the system when the Main Menu (_MAINMNU form) is run; ALL security

24. Refer to section 3.a.ii.(3) for details of some enhanced security options.

permissions are revoked when the Main Menu is closed (or in the event of abnormal termination of PDOXWIN, such as a power outage). The Monitoring Database System Designer's password must be provided to the system manually. This password should *never* be used except by the Monitoring Database System Designer or upon his/her instruction; *improper use will result in incompatibility among versions of the software, and may cause your software to crash.*

(3) Possibilities for other uses of security

It is possible to set the system up to require users to log on and provide a password for each session.[25] Varying levels of access--e.g. read/write, read only, and NO access--can be provided all the way down to FIELD level. If measures such as these are deemed necessary, the Database Administrator of your organization should contact the Monitoring Database System Designer for details.

(4) Security loophole (RE: referential integrity)

Unfortunately, there is a flaw in Borland's implementation of security regarding unauthorized changes to the structure of tables. The good news is that this loophole will have ABSOLUTELY NO EFFECT if you use the Monitoring Database as it is delivered. Users of the Monitoring Database will ONLY have problems under

25. Note to [...system designer]: If only a single password (for general system logon) is needed, simply replace the [...???] subroutine with a null module and the user will be prompted for the existing password (which the [...???] subroutine formerly supplied automatically). If this is done, the general user access password should be changed to be unique for each site.

If multiple security levels--or unique passwords for individuals--are required, because of the number of tables for which security levels must be provided by restructuring the tables (and because this is a tedious and time-consuming task, even if automated as much as possible), it is recommended that only one PDOXWIN password for each LEVEL of security needed be assigned at the table level, then use a programmer-supplied cross-reference table (to which only the site's Database Manager has password access) which allows the system to translate the individual password, then supply to the system a PDOXWIN password which corresponds to the level of security for that user-defined password.)

In fact, this type of user-defined password security should probably be included as a standard feature in a future release of the Monitoring Database.

Another security enhancement feature could be a "system parameter table" [read by the [...???] subroutine] allowing the option of whether or not to automatically provide ANY password to the system.

particular circumstances related to ADDING NEW TABLES.

For anyone using the Monitoring Database within the PDOXWIN environment WITHOUT the Monitoring Database System Designer password, effects of this security loophole will ONLY surface IF: (1) tables are created and linked to the core infrastructure of the Monitoring Database via referential integrity AND (2) an inappropriate action (e.g. deletion) is taken on the newly-created file BEFORE the referential integrity relationship is REMOVED. Although changes cannot be made explicitly to table structures without property security clearance, the creation of a referential integrity **child table** implicitly changes the structure of its **parent table** (its .VAL file is updated).[26] This type of problem cannot occur from within the PDOXWIN environment with files provided with the Monitoring Database, because these actions are prohibited without the Monitoring Database System Designer's password access. However, such actions CAN occur from within the PDOXWIN environment, since such actions are permitted on newly-created tables which are not protected by password security. Therefore, although a user still follows the guidelines for "foolproof" use of PDOXWIN (see section [..."rules for use of PDOXWIN"]),

26. Please DO call Borland and let them know that you want this loophole closed! The more they hear about this from users, the more likely it is that they will fix it in the future.

Additionally, let them know that you want **referential integrity** (RI) information to be REMOVED from the .VAL file and placed in its OWN file. The reason for this is that the .VAL file also contains "validity check" information (field pictures, table lookup information, etc.) which is actually UNRELATED to RI information; therefore, all this information is LOST and must be reconstructed for potentially ALL
TABLES IN THE DATABASE (a tedious and time-consuming process) in the event that the RI structure of ANY table in the database is breached. (RI structure can bewhich it can be by inadvertently deleting the .VAL file, deleting one of the related tables in the **database (1)**, or via another bug [known & reported by me, but not yet officially acknowledged by Borland; it's characterized by having unresolved references to a nonexistent "RESTTEMP.DB" interim table] which causes RI structure to become corrupt.

there is no way to guarantee that this problem will not occur under these circumstances.[27]

b. Changes to database structure

Because the Monitoring Database is based on Paradox for Windows, its flexibility is virtually boundless. Although there are certain restrictions regarding what changes should be made to existing data structures, Database Administrators/Programmers can add unlimited functionality to the data infrastructure provided by the Monitoring Database.

i. Core tables

In order to ensure compatibility among versions of the Monitoring Database, changes should NOT be made to core tables (those tables initially distributed as part of the Monitoring Database) without coordination with the Monitoring Database Designer.

ii. How to add functionality

Unlimited functionality can be added to the Monitoring Database by creating new tables and/or adding custom forms, reports, or queries. Custom functions can be integrated into the existing menu structure via the User Menu. The Database Administrator of your organization is responsible for maintenance and documentation of value-added components.

(1) Responsibility for changes

The Database Administrator of your organization is responsible for maintenance and documentation of value-added components. This includes the responsibility of converting all add-ons to new releases

27. Note the liberal use of the qualifying statement "within the PDOXWIN environment." OUTSIDE the PDOXWIN environment (e.g. in File Manager or at the DOS prompt), there are no guarantees at all, since PDOXWIN security constraints are not enforced outside the PDOXWIN environment. If, for example, ANY .VAL file is deleted which is associated with any file involved in a RI relationship, problems similar to these will occur.

Although I have not tested this idea (so complications may occur that I have not foreseen), the only level of protection against this in a DOS-type environment that I know of would be to set the "read-only" attribute of all .VAL files (see DOS documentation for details). This obviously would not work for other files (such as .DBs) which must be updated, therefore the value of this approach is limited. However, it might serve as an "early warning beacon" to those who attempt to cause (even implicit) changes to file structures, forcing them to manually change the file attributes for the .VAL files which would be affected by any of their actions and therefore acknowledge that they will be "implicitly" changing that .VAL file.

of the Monitoring Database. Therefore, all customized changes should be designed with modularity in mind, and be very well documented to ensure smooth transition between software releases.

(2) User Menu

A "user menu" is provided to allow site-specific custom functions to be incorporated directory into the menu framework of the Monitoring Database. There is a direct link to this menu from the first page of the Main Menu of the Monitoring Database. The source file (USERMENU.FSL) for this menu ONLY[28] is distributed in the main Monitoring Database directory; this file can be updated by the user to perform absolutely any user-defined function (including branching to an entirely new submenu system).

(3) User-defined tables

You have the ability in PDOXWIN to create user-defined tables. This means that you can store ANY specialized or additional data AND take advantage of the data infrastructure of the Monitoring Database.

(a) What you MAY do

You may add related tables to the Monitoring Database by creating new tables[29] within the working directory. You may then optionally create a **table lookup** relationship to the corresponding fields in an existing table. (You may, for example, create your own custom data set types and enforce data integrity based on your existing taxon codes and monitoring unit definitions.)

28. Typically, all other forms in the main Monitoring Database are NOT user-modifiable; they have the .F**D**L extension.

29. Hint: If you are creating a new table to be relationally linked to an existing table, use the "Borrow" button in the table creation dialog box to copy the field structure & validity checks [as a general rule, DO NOT borrow the referential integrity structure!], delete unneeded fields, then create additional fields as necessary. This is an easy way to ensure consistency between parent and child table structures.

*(b) What you SHOULD **NOT** do*

[...] [never rename, delete, run TRU on a file w/ ri]

*(c) What you may do **AT YOUR OWN RISK***

You may add **referential integrity** to a user-created table AT YOUR OWN RISK. The reason for this emphasis is that resolution of problems created by errors in the procedures outlined in this section can be tedious and time-consuming. Assistance by the Monitoring Database System Designer with resolution of any problems which occur as a result of attempting any of these things will be chargeable at no less than $80/hour. This is not to discourage you from doing these things; hopefully, however, it will encourage you to be sure you know what you're doing before you try!

IMPORTANT: You should THOROUGHLY UNDERSTAND the implications of **referential integrity**--including PDOXWIN's security loophole regarding RI (see section 3.a.ii.(4))--before adding RI to any table associated with the Monitoring Database.

IMPORTANT: Before attempting to RENAME or DELETE any user-created table, ensure that RI is REMOVED beforehand! One idea to help safeguard your database is for your organization's Database Administrator to assign a master password (such as "DoNotDeleteUntilRIisRemoved") to each newly-created table with which RI is associated. This way, there will be no excuse for forgetting!

(4) Unlimited additional functionality

Users can use the full functionality of PDOXWIN to create custom forms, reports, queries, and/or scripts (macros/programs) for use with existing or new tables: there are no limits! Any customized feature can be

incorporated into the menu system of the Monitoring Database via the User Menu.

c. Implications of Password Protection to Changes in the Monitoring Database

(Programmer)[...]

i. Can't directly change core tables

[...] (refer to ?)

ii. Referential Integrity Loophole

[...](ability to IMPLICITLY affect structure of core tables by adding an RI link)

CHAPTER 4. Who does what

[...using top-down approach] Generally, each level should understand everything that "all users" should know; everything relevant to his own personnel group level; and everything relevant the functions of all subsequently-described personnel groups. [...]function of each group is to provide tech support for others within your own organizations, AND to provide support for your peers in other organizations

a. Personnel Groups

This section lists and defines each of the personnel groups targeted by this documentation.

From a "real world" perspective, this is a proposed plan only. Your organization may be structured such that one person may have the duties of more than one of these personnel groups, or the duties of one personnel group could be divided among personnel. However, the person who makes the final decision about distribution of responsibility regarding the Monitoring Database should understand the rationale behind assignment of the duties to these categories before altering this scheme.

i. Monitoring Database Project Coordinator (System Designer)

[...]

ii. Database Administrator

One person should be selected to be the Database Administrator within each organization using the Monitoring Database. The Database Administrator is overall in charge of the organization's use of the Monitoring Database. Programmers, Site Managers, and Data Entry Personnel should report to the Database Administrator for all matters regarding the Monitoring Database.

The Database Administrator should be the main person in the organization to interact with the Monitoring Database Project Coordinator (System Designer).
An organization's Database Administrator should at the very least be aware of (and log) every enhancement request submitted to the MDPCSD. This arrangement serves several

purposes: (1) it ensures that there is one person who is aware of all aspects of use of the Monitoring Database within the organization; (2) this person can act as both a central repository and central distribution center for relevant information about the system within his organization, among organizations who use the Monitoring Database, and as a liaison with the MDPCSD; and (3) the support load on the MDPCSD is alleviated because all support requests to him will therefore have been pre-screened by a knowledgable party to avoid triviality and redundancy.

iii. Programmer

[...] The Programmer described here is a programmer within a user organization (as distinguished from the Monitoring Database System Designer). (within an organization; should report to the Database manager)

iv. Site Manager

[...] [responsible for the data at a site; responsible to ensure that appropriate info is passed to the Database Administrator, and that changes recommended by the Database Administrator are implemented]

v. Data Entry

[...] [enter field data] [enter supporting data which has been decided upon by a Site Manager or Database Administrator]

vi. Manager-user

The Manager-user personnel group encompasses anyone who has access to the Monitoring Database but is interested only in its output, and may not possess (nor want/need to possess) detailed technical information or skills regarding details of the system. Typically, Manager-users would need to extract summary information from the system for education or management decisions.

At least three groups of Manager-users can be envisioned. The first type would include personnel who would receive hard-copy information only from the system, either periodically or by specific request. The only knowledge needed by this

group would be of what is available from the Monitoring Database. The next two groups would require direct access to the Monitoring Database (e.g. via a LAN). The first of these would be [get canned reports on their own, and/or create system-assisted queries {"Manager Query" function}...].
The most sophisticated level of Manager-user interaction with the system would be [...create their own PDOXWIN queries, reports, etc.; would require knowledge of the structure of the database--which is available online ()--and of PDOXWIN] [...read-only access passwords can be created for users with direct system access].

vii. Internet Users

Ultimately (though this is not yet implemented), information from the Monitoring Database could be made available on the internet in a standardized way. Therefore, although they may not be intrinsically part of your organization, Internet Users could comprise a part of the group of users of your Monitoring Database. Internet Users would typically be guided through any information you provide through point-and-click hypertext and photo links, and would be able to learn from your data with virtually no knowledge of the Monitoring Database.

[...where to put the following...?]It is planned that a system will be implemented to allow you to easily allow internet access to selected data (probably COPIES of the data, summarized only [if desired]). The internet is an excellent medium for public or private distribution of summary or detail data to selected audiences (e.g. other selected Monitoring Database users or the general public). Internet distribution of data is particularly exciting because of the potential of Internet Users being able to transparently obtain composite information extracted from multiple Monitoring Database systems, each set of information being maintained independently by its respective data owner. (Of course, you [the data owner] will have complete control over the information [if any] which is released to the internet.)

Until a standardized system for internet distribution of Monitoring Database information is devised, there is no reason

that innovative individuals and organizations cannot publish information derived from this system on the internet. In fact, this is encouraged; please report your work (successes and heartaches) to the MSDCSD!

b. What every Monitoring Database user should know

This section [basic data entry, system navigation, etc.]... [...hierarchical relationship of tables; table lookup {entry MUST EXIST in higher-level table/etc.}]

i. General overview of the Paradox for Windows environment

Generally, all user functionality of the Monitoring Database is built in to its menu system. However, to go further with the system (e.g. to create your own queries), the user should first become somewhat familiar with the PDOXWIN environment. It is not within the scope of this document to explain this; users are referred to other sources of information about this topic.

(1) "Getting started" tips

Here are two tips before you get started learning about Paradox for Windows environment: (1) Check out the "Project Viewer." (2) Realize that, as with any powerful software product, you must WORK with it in order to learn it: "Time breeds proficiency."

(2) Where to get more information

Refer to the "Quick Start" guide ("A quick tour of the desktop"), "User's Guide," the on-line help, and other Borland-supplied documentation for additional information. A variety of third-party books are also available for Paradox for Windows users of all levels. Additionally, visit Borland's web site[30] for the latest news about PDOXWIN and other Borland products, including product information, technical information sheets, literature lists, news releases, downloadable files, and more.

ii. Rules for use of PDOXWIN

Here are a few basic tenets which, if heeded, will prevent unnecessary headaches when using PDOXWIN. Although

30. http://www.borland.com

these are geared towards system administrators, all users should at least skim these topics in order to avoid problems.

(1) Renaming/deleting files

Don't ever, **ever**, ***ever*** rename or delete *any* files outside the PDOXWIN environment[31] (e.g. DOS prompt, Windows File Manager, Explorer). This also applies to files which you don't understand or may "appear to be useless" (e.g. *.X02); these are often critical to the operation of your database and/or contain data.

(2) Use of Table Repair Utility

At least through version 5.0 of PDOXWIN, the Table Repair Utility (TRU) (which no one but a system administrator should EVER attempt to use!) does NOT support **referential integrity**. **Referential integrity** should be removed[32] from any table prior to using TRU.

iii. Rules for use of Monitoring database

Here are a few basic tenets which, if heeded, will prevent unnecessary headaches when using the Monitoring Database (including getting you in trouble with your system administrator).

*(1) Use **FORMS**, not **TABLES***

Always use **form**s provided for viewing/manipulating data; do *not* manipulate the **table**s directly. The reason for this is that some forms contain certain types of validity checks and/or perform critical calculations which will NOT be performed when using tables directly. (In fact, this is a good guideline to follow when using PDOXWIN in general. Get in the habit of creating a form for EVERY table you create, and always use

31. If you know EXACTLY what you are doing (e.g. you know that NO referential integrity is used, you rename all related files at once, you understand the implications to related forms, reports, and/or data models, AND etc.), it's POSSIBLE to do a limited amount of manipulation IF you COMPLETELY understand all the implications.

32. It might often be the case when you need to use TRU that you can't restructure the table in order to remove RI. In this case, you must manually delete the .VAL file, but be aware that you will then have to do the same for ALL related files, and therefore reconstruct not only all RI relationships, but also all validity checks for all affected files.

the form instead of directly opening the table, even if you do not add any "fancy" features to the form.)

(2) Modify only data for which you are responsible

Do not attempt to modify (add or change records) any data for which you are not directly responsible.
For example, the addition or changing of a Taxon is something which must be coordinated among various AGENCIES. If you bypass the "chain of command" in your organization, there will be no "audit trail" to show which changes were made; your organization will be out of sync with others. Another example would be addition of or change to Taxon Codes. Changes here should not only be coordinated among agencies, but can profoundly affect the analysis of data on your system. It is likely that only your organization's Database Administrator fully understands the "ripple effects" of this change.[33]

33. Although this is NOT the way v.1 is installed by default, your system may be set up so that you may READ-ONLY access to unauthorized fields; in this case, it should be impossible for you to make unauthorized changes (see section 3.a.ii.(3)). Check with your organization's Database Manager to see how your system is set up.

c. What the Monitoring Database System Designer should know

This section discusses the information with which the Monitoring Database System Designer should be familiar. An overview of the responsibilities of the Monitoring Database System Designer's is presented.

i. The Monitoring Database System Designer should know EVERYTHING!

(That was easy!) The Monitoring Database System Designer should have an excellent knowledge of both interactive Paradox for Windows and ObjectPAL, as well as in-depth understanding of the current design of the Monitoring Database. Additionally, (s)he[34] should understand (and have documentation from users regarding) the users' monitoring goals, field methods, and data entry and analysis needs.

ii. Design change decisions

Decisions to change the design of the Monitoring Database should not be taken lightly--even minor changes--because of the consideration of keeping multiple sites' versions compatible/identical. **It is critical that any site-specific changes are documented COMPLETELY in a standard format, IF SUCH CHANGES ARE MADE AT ALL.**

iii. Compatibility with future upgrades

[...]

iv. Distribution of upgrades & patches

Distribution of upgrades and patches will be done via the internet at the Hawaii Natural Resource Monitoring Database home page[35] and/or via the Hawaii Natural Resource Monitoring internet list (HIMONDB-L; see section 1.f).

34. The issue must be addressed, I suppose! While not implying sexism in any form, the generic "he" will be used from this point forward in this document to enhance flow and readability.

35. http://www.botany.hawaii.edu/MonitoringDatabase

v. Tech support for Database Administrators

At the time of publication of this documentation, Database Administrators may contact the Monitoring Database Project Coordinator[36] for support regarding the Monitoring Database. For detailed questions regarding specific uses of PDOXWIN, the Database Administrator may be referred to one of Borland's technical support options (see Borland documentation for details).

Please remember that ONLY an organization's Database Administrator should contact the Monitoring Database Project Coordinator directly with PDOXWIN/Monitoring Database questions. Other users should first attempt to resolve questions from within their own organization (ask your Database Administrator or other designated in-house support person). If the problem cannot be resolved in this way, the fully-apprised Database Administrator should then contact the Monitoring Database Project Coordinator for support.

36. Philip A. Thomas, Research Corporation of the University of Hawaii, P.O. Box 369, Makawao, Maui, HI 96768; voice: (808) 572-9306 ext. (5) 5938; e-mail: thomasp@maui.com

d. What the Database Administrator should know

This section discusses the information with which the Database Administrator should be familiar. An overview of the responsibilities of the Database Administrator is presented. Addressed also are issues which should NOT be delegated to data entry personnel; information about "where to go for help" is provided here. Finally, details of the data entry person's job are presented. This section can and should be used--in conjunction with other related resources--as both a training aid and an on-the-job reference.

i. Responsibilities (overview)

[...] Understand "What the Site Manager should know"; have at least an overview of "What the Programmer should know"

(1) Sites' hardware & software

[...](operating systems, networks, etc.); backup procedures; "high points" of PDOXWIN (including referential integrity)

(2) Installation of the Monitoring Database

[...see Roy's writeup; include:]

(a) system requirements

The minimum system requirements for running PDOXWIN version 5 are as follows: 80386-based PC; 6Mb RAM (8Mb recommended); xxx Mb disk space (minimum installation) or xxx Mb disk space (full installation). You should allow a minimum of xxx Mb for initial installation of the Monitoring Database, and additional space as needed for your data. It is HIGHLY recommended that you use the fastest computer possible for the Monitoring Database.[37] PDOXWIN--as ANY full-featured database--is resource-intensive. Your minimum system requirements may vary if you use a version

37. A frequently asked question is "Would it be worth it for me to add memory (RAM) to my system?" The short answer is: for PDOXWIN, you will notice a great difference in performance from 8 Mb to 16 Mb; it's worth it! The longer answer--and the answer for other situations--requires analysis of your particular system and needs, and should be discussed with knowlegeable personnel on a case-by-case basis.

of PDOXWIN other than v.5; consult your PDOXWIN documentation for specifications for your version.

(b) create directory structure

The recommended directory structure includes two directories: (1) the directory containing the data, forms, reports, and queries; and (2) a directory containing elements (such as subroutines) in common with other databases that may be on your system. In order to maintain consistency with Monitoring Database installations in other organizations, it is strongly recommended that you use for the directory containing the data a directory named MONITRNG, directly below the PDOXWIN\WORKING subdirectory (e.g. C:\PDOXWIN\WORKING\MONITRNG). For the same reason, it is strongly recommended that you use ...\PDOXWIN\WORKING\COMMON for your "common" elements.

(c) create alias(es)

Next, one critical and one recommended alias should be created. The correct assignment of the :Common: alias is critical for the Monitoring Database to function; creation of a :Monitoring: alias is suggested for ease of PDOXWIN system use (when working with multiple databases), and for consistency with other sites. The :Common: alias should be assigned to the directory mentioned in 4.d.i.(2)(b) containing common program elements (the "Common directory"). The :Monitoring: alias should be assigned to the directory mentioned in 4.d.i.(2)(b) containing your data (the "monitoring data directory").

To create the aliases, run PDOXWIN, then choose File | Aliases, select New, then type in the alias name (don't type the colons) and its associated path. ("Driver type" [...wording?] should always be

"STANDARD.") (Note: You may also manipulate aliases via the IDAPI [BDE] configuration utility; refer to Borland documentation for details.) Refer to Borland documentation for further details regarding creation and use of aliases.

(d) copy files

Copy all files from the distribution medium in the MONITRNG directory on the distribution medium to the monitoring data directory on your system. Copy all files from the distribution medium in the COMMON directory on the distribution medium to the common directory on your system. (If the files on your distribution medium were compressed and/or combined using a compression utility [such as PKZIP], follow the recommended procedures for that utility to expand the files into the appropriate directories.)

(e) set up high-level data

[...] [...include coordination of updating of taxonomic information with Site Managers for multi-site organizations] [work with Site Managers to set up this data]

(3) Responsible for/should be consulted for high-level data changes

[...<--how is this going to be designated in TABLEDOC?]

(4) Organization-specific documentation

[...] (...how does your organization's structure relate to the Personnel Groups outlined in this documentation? QC RE: procedures, metadata entered into TMD, particularly higher levels)

It is the responsibility of each site's Database Administrator to maintain a log of ALL programming/enhancement changes to their copy of the Monitoring Database, AND to incorporate their site-specific changes into any new versions of the Monitoring

Database. (Emphasis is placed on this because even seemingly minor changes--particularly as changes evolve, and particularly to ANY data structures--almost certainly cause nontrivial headaches when attempting to upgrade to a new version.) ALL site-specific changes to the Monitoring Database should be made only after careful consideration of migration of the customizations to a future software release.

(5) Pass information to the Monitoring Database System Designer

[...consolidate users' "wish lists"/etc. & submit to Monitoring Database System Designer]

ii. What the Database Administrator SHOULDN'T have to know

[...]

iii. Nitty-gritty details

[...]

e. What the Programmer should know

This section discusses the information with which any Programmer working on the Monitoring Database should be familiar before working on the system. An overview of the responsibilities of the programmer is presented. Addressed also are issues regarding what should NOT be performed or attempted. Information about "where to go for help" is provided here. Finally, useful details of items of interest to Programmers are presented. This section can and should be used--in conjunction with other related resources--as both a training aid and an on-the-job reference.

i. Responsibilities of the Programmer

The responsibilities of the Programmer include adding functionality to the core Monitoring Database to facilitate the needs of his organization. Generally, these responsibilities would include creation of new forms and analysis tools (typically accessed by users via the "User Menu" of the Monitoring Database) and creating auxiliary tables (which may "hook in" to the data infrastructure of the Monitoring Database) for specific data needs.

It is the responsibility of the Programmer NOT to make any design changes--no matter how minor--to the core of the Monitoring Database (i.e. anything existing upon installation of the software; anything that the Programmer didn't write himself and add to the "User Menu"; ANY data structure [see in particular 3.c.ii]). Generally, ALL changes to the core of the Monitoring Database should be made ONLY by the Monitoring Database System Designer. If it is ever deemed necessary for the Programmer to make any such changes, it should be done ONLY after consulting and getting the approval of both the Database Administrator AND the Monitoring Database System Designer, THEN documenting the change fully using the standard documentation method (the NOTES table, accessible from the System Documentation Menu) (see 4.c.i).

[...HERE: put note RE: RI & security loophole...]

NOTE: It is the responsibility of each site's Database Administrator to maintain a log of ALL changes to their copy of the Monitoring

Database, AND to incorporate their site-specific changes into any new versions of the Monitoring Database (see 4.d.i.(4)). (Emphasis is placed on this because even seemingly minor changes--particularly as changes evolve, and particularly to ANY data structures--almost certainly cause nontrivial headaches when attempting to upgrade to a new version.) ALL site-specific changes to the Monitoring Database should be made only after careful consideration of migration of the customizations to a future software release.

ii. What the Programmer SHOULDN'T have to know

[...]

iii. Nitty-gritty details

[...put info here RE: how to hook into system via USERMENU.FSL...]

[...][be aware that after having been compiled in 32-bit v.7, you can't take it back to v.5; by design, on one hand, but also--because of what I consider a bug--the .FSL is evidently modified so it CANNOT even be recompiled using a previous version; related to 32-bit implementation of OLE]

f. What the Site Manager should know

This section discusses the information with which the Site Manager should be familiar. An overview of the responsibilities of the Site Manager is presented. Addressed also are issues which the Site Manager should NOT attempt to resolve without consultation with the Database Administrator. Finally, details of the Site Manager's job are presented. This section can and should be used-- in conjunction with other related resources--as both a training aid and an on-the-job reference.

i. Responsibilities (overview)

[...] [...include working with Database Administrator to set up higher-level data; should document changes AND rationale & submit to Database Administrator for incorporation into organization's site-specific documentation, since DM is responsible for maintaining this]

ii. What the Site Manager SHOULDN'T have to know

[...]

iii. Nitty-gritty details

[...]

(1) How to determine where to enter data

The key to success in creating/using the data structures provided in the Monitoring Database is ensuring that values are entered at the appropriate LEVEL in the table hierarchy. For example, a general location map of a PLOT should be entered into the MONUNITS[38] table in the record where the PLOT is defined,[39] since the info at this level--such as a map of the area--is static relative to the monitoring unit itself. However, a photo of that same plot showing specific conditions (e.g. pig damage) on a certain date should be entered into the

38. NOTE: The components of the monitoring unit comprise the KEY of the MONUNITS table.

39. [...]See section 5.b.ii.(6), 5.b.ii.(7) for details about setting up monitoring units.

DATACOLS[40] table--NOT the MONUNITS table--because the photo is relative to that plot *at a particular point in time.* (Notice that the difference in MONUNITS and DATACOLS is *temporal;* DATACOLS info is related to [i.e. contains reference to] MONUNITS info, and also includes the *date* on which the plot was visited.
All non-key information in the DATACOLS table is date-dependent [e.g. weather conditions, who performed the field work].)

(2) Setting up the system for users

[TABLEDOC form: start from top & work down]

(a) Taxonomic infrastructure

With the exception of "loc."-ranked taxa (and their associated taxonomic codes),[41] only the Database Administrator should ever authorize changes to any taxonomic infrastructure. However, the Site Manager should thoroughly understand the taxonomic infrastructure setup. (See section 5.b.iii for details.)

(b) Project definition & miscellaneous

[...]

(i) AREAS[...]

[...]

(ii) PROJECTS[...]

[...]

(iii) PRJDSTPL[...]

[...]

(iv) ELEVSRCS[...]

[...]

40. NOTE: All key fields from MONUNITS are included in the DATACOLS key, *PLUS the observation date.*

41. See section 5.b.iii for details.

(v) NESW[...]

[...]

(vi) COMTYPES[...]

[...]

(vii) GUYPLTCT[...]

[...]

g. What Data Entry personnel should know

This section discusses the information with which Data Entry Personnel working on the Monitoring Database should be familiar before working on the system. An overview of the responsibilities of Data Entry Personnel is presented. Addressed also are issues regarding what should NOT be performed or attempted; information about "where to go for help" is provided here. Finally, details of the data entry person's job are presented. This section can and should be used--in conjunction with other related resources--as both a training aid and an on-the-job reference.

i. Responsibilities

The responsibilities of Data Entry Personnel include[...]

NOTE: It is the responsibility of each Data Entry Person to [...report problems/etc. to the Site Manager...]

ii. What Data Entry Personnel SHOULDN'T have to know

[...]

iii. Nitty-gritty details

[...]

h. What Manager-users should know

This section discusses the information with which Manager-users should be familiar before working on the system. An overview of the [...needs] of Manager-users is presented. Addressed also are issues regarding what should NOT be performed or attempted; information about "where to go for help" is provided here. Finally, details of how the Manager-user can access needed information are presented. This section can and should be used--in conjunction with other related resources--as both a training aid and an on-the-job reference.

i. ???

The responsibilities of Manager-users include[...]

???

ii. What the Manager-user SHOULDN'T have to know

[...]

iii. Nitty-gritty details

[...]

i. What Internet Users should know

This section discusses the information with which [...TAILOR THE REST OF THIS FOR INTERNET USERS; MORE ON WHAT THEY CAN FIND OUT VS. WHAT THEY NEED TO KNOW; COVER HOW MATERIAL SHOULD BE PRESENTED, AND WHAT TYPES OF INFO SHOULD BE AVAILABLE] working on the Monitoring Database should be familiar before working on the system.
An overview of the responsibilities of Data Entry Personnel is presented. Addressed also are issues regarding what should NOT be performed or attempted; information about "where to go for help" is provided here. Finally, details of the data entry person's job are presented. This section can and should be used--in conjunction with other related resources--as both a training aid and an on-the-job reference.

i. Responsibilities

The responsibilities of [...INTERNET USERS]nclude[...]

ii. What Internet Users SHOULDN'T have to know

[...]

iii. Nitty-gritty details

[...]

CHAPTER 5. "HOW-TO" DETAILS

This chapter provides hands-on, "how-to" instructions for getting started and working with the Monitoring Database. Note, however, that this chapter is intentionally NOT at the beginning of the documentation. The placement of this chapter near the END of the documentation is appropriate: the FIRST step in getting started is to READ ALL RELEVANT PARTS OF THIS DOCUMENTATION. You should NOT start with this chapter. If you haven't already, please start at the beginning of the documentation and work your way up to this chapter. This chapter skims over or completely omits explanations of critical concepts which you should understand BEFORE performing any of the tasks outlined here.

a. Installation of the Monitoring Database

This section outlines installation procedures for the Monitoring Database. It is assumed that Paradox for Windows has been already installed on your system. (For instructions on how to install Paradox for Windows, refer to the documentation included with the Paradox for Windows software.)

b. Initial setup of the Monitoring Database

This section outlines the initial setup procedures for the Monitoring Database. It is assumed that the Monitoring Database has been installed in accordance with section 5.a.

Note that the setup tasks MUST be performed in a specific order (the order in which they are presented here).

i. Preparation

In preparation for initial setup, print page 1 of the TABLEDOC form. (To do this, click the SYSTEM DOCUMENTATION button on the Main Menu, then the TABLEDOC button; then click the printer icon and specify to print pages 1 through 1.) This will provide you with a hard copy of the **table relationship diagram** for the system, which will be your "roadmap" during the initial setup process. Generally, our work will progress from the TOP of the **table relationship diagram** towards the bottom.

The initial setup of the Monitoring Database includes assigning values to certain higher-level data layers. This process is important because, as you recall, higher-level values must exist prior to entry of lower-level data in order to allow validation and force consistency and integrity of data. Only your site's Database Administrator--with input from the Site Manager(s)--should perform these functions.[42]

ii. Site-specific infrastructure setup

This section covers setup of site- (or organization-) specific setup parameters. Generally, this information is contained in the tables on the left-hand side of the **table relationship diagram** in Figure 1.

(1) Setting up AREAS

The purpose of data in the AREAS table is to identify the geographic area to which your data is relevant. Ultimately, the AREA can be associated with a spatial "GIS" element to indicate general geographic areas in which data has been collected. It is suggested that broad, general areas be formulated, such as islands or island regions.[43]

To set up AREAS, from the HOUSEKEEPING menu, click the AREAS button and enter the AREAS that you will be using.

(2) Setting up PROJECTS

The purpose of data in the PROJECTS table is to identify the PROJECT with which your data is associated. A PROJECT should be defined as a set of data which is related to one another in some fashion related to the analysis of that data.

42. It cannot be overly stressed that since high-level data layers affect virtually ALL other data, it should be set up very carefully from the beginning, and should be changed or added to only after careful thought. Changes to high-level data layers profoundly affect the entire database, and may be tedious and/or time-consuming to implement.

43. Remember that each project can currently be associated with only a single area; however, areas can be associated with multiple projects.

For example, "East Maui Watershed" is a good choice for a project, since various types of data (threat monitoring, bird counts, and various methods of vegetative data collection; inside & outside of preserves and/or threat-controlled areas; collected on both public and private lands) are analyzed with respect to one another in the context of the East Maui Watershed Partnership.

Another way to approach project definition may be preserve-based: a project may comprise all data collected in/with respect to a particular preserve.

To set up PROJECTS, from the HOUSEKEEPING menu, click the PROJECTS button and enter the PROJECTS that you will be using.

(3) Setting up ELEVATION SOURCES

Some ELEVATION SOURCES (e.g. GIS, METERED, TOPO) are predefined and included with the installation version of the Monitoring Database. These are examples only; you should more specifically define ELEVATION SOURCES that you use (e.g. instead of "GIS" you might use "WGS84+-3M", defined as "WGS-84 coordinates from Trimble PF-Pro, PDOP <=4, post-corrected against NPS Kaloko-Honokahau base station, confidence ±3m").

It is highly recommended that your organization's Database Administrator be the person to add/change ELEVATION SOURCE codes, and that he consult with other organizations' Database Administrators to coordinate these codes where possible.

To set up ELEVATION SOURCES, from the HOUSEKEEPING menu, click the ELEVSRCS button and enter the ELEVATION SOURCES that you will be using.

(4) Setting up DIRECTIONAL (LOOKUP) information

No setup is necessary for the DIRECTIONAL (LOOKUP) information; the NESW table simply contains the values "N", "E", "S", and "W"[44] so that directions (e.g. for point-intercept plots) can be validated and looked up during creation of their respective monitoring units (section 5.b.ii.(6), 5.b.ii.(7)).

(5) Setting up COMMUNITY TYPES

Creation of COMMUNITY TYPES is entirely optional, since they are NOT required anywhere in the Monitoring Database. However, as user-defined community types[45] may be an interesting way to analyze data, the option to define them at the level of any monitoring unit has been provided. Any community type referenced in the MONUNITS table must have been previously defined in the COMTYPES table.

To set up COMMUNITY TYPES, from the HOUSEKEEPING menu, click the COMTYPES button and enter the COMMUNITY TYPES that you will be using (if any). (This step is optional, since COMMUNITY TYPES are not required. You may add COMMUNITY TYPES and/or incorporate them into your monitoring unit definitions at any time.)

(6) Setting up TRANSECT TYPES

TRANSECT TYPES are required for monitoring unit definition, and will ideally be defined consistently among organizations using the Monitoring Database whose data will be compared. The GENR&TMON (General-purpose resource & threat monitoring) transect is predefined (and should be the only entry in the TRANTYPS table upon initial system installation). The GENR&TMON transect is defined as a multi-purpose transect, along which are

44. Due to data validation constraints built in during table creation--this done by design--no other values can be added to the NESW table.

45. The point has been raised that the data sets themselves should reveal the true "community type" of each group of data. However, practical constraints--as well as the desire to compare data to predefined "standards"--dictate that this user-defined option be included in the Monitoring Database.

defined 10m (and therefore also 50m and 250m) stations, 5x10m and 5x50m rectangular plots (for ungulate activity monitoring), and circular and/or square plots (for point-intercept and other monitoring). Therefore, as long as the physical transects along which your data are collected conform to this multifaceted definition, this transect type is the only type which should be used for all data set types which depend on one of these layouts. (The specific portions of the transect layout relevant to the data set type you are using should be outlined in the DATASTYP [data set types] table.)

To view TRANSECT TYPE information, from the HOUSEKEEPING menu, click the TRANTYPS button and enter the TRANSECT TYPES that you will be using. Remember, however, that unless (1) you are collecting data in a nonstandard manner, (2) you set up a transect that does not conform to the definition of the GENR&TMON transect type, or (3) you create a new data set type which does not use any feature defined along this type of transect (AND the new transect feature cannot be logically appended to the description of the GENR&TMON transect type), you should NOT add a new transect type. Your organization's Database Administrator should be the only person to add transect types, and then only after discussing the issue with the System Coordinator and other organizations' personnel.

(7) Setting up MONITORING UNITS

The setup of MONITORING UNITS is arguably the most confusing part of setting up the Monitoring Database. This is because (in one sense) MORE THAN ONE ENTITY TYPE is defined within the MONUNITS table, e.g. transects, plots, and stations. Note, however, that each of these entities is, in fact, a MONITORING UNIT.

Consider for a moment the example of a TRANSECT along which there are a number of 5m x 50m PLOTS (sound familiar?). Additionally, consider that there are

STATIONS marked along that transect at 10m intervals. In order to enter data into the Monitoring Database from any of these MONITORING UNITS (the TRANSECT, PLOT, and STATION), you must "tell" the system that these specific entities exist. The way this is done is by entering them into the MONUNITS table. If our hypothetical transect is 500m long, it will contain 10 PLOTS and 50 STATIONS.[46] All of these entities--the transect, plots, and stations--are defined in

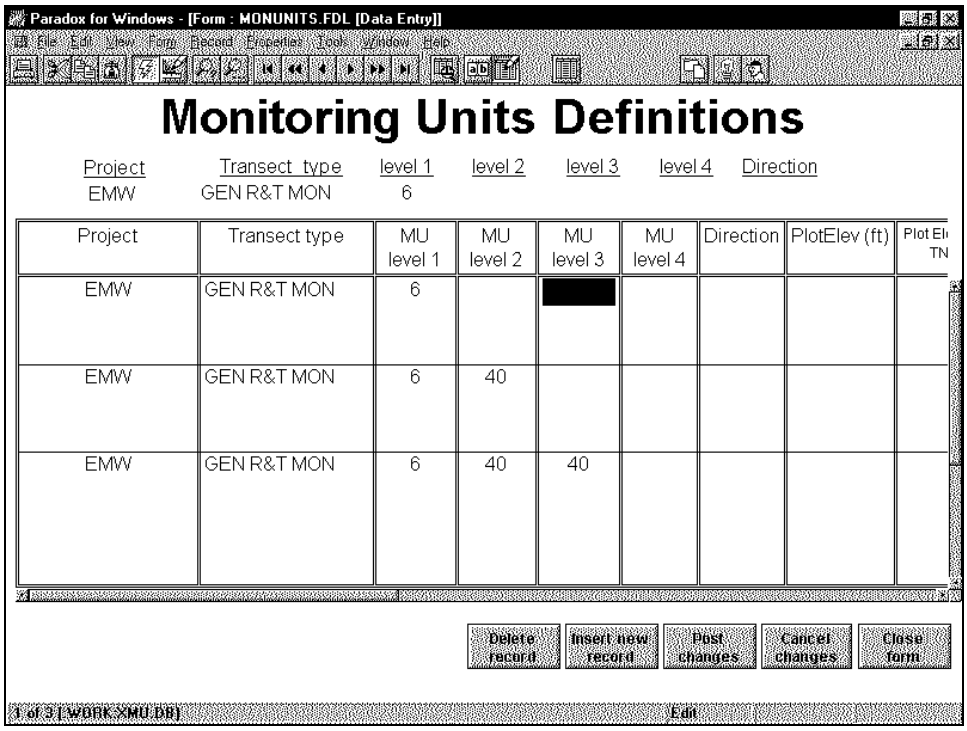

Figure 2

the TRANSECT is the "top level" entity, TRANSECT will correspond to "Monitoring unit level 1" in the MONUNITS table. Thus, to define the transect, you create a **record** with only the first monitoring unit **field** ("MU level 1") filled in (leaving "MU level 2," "MU level 3," and

46. Though in reality, there are 51 points at 10m intervals along a 500m transect, it is logical to use only 50. To illustrate this point, imagine STATIONS being numbered starting with their BEGINNING meter mark (e.g. station 0, station 10, station 20, etc.) with five stations associated with each PLOT for analysis purposes (e.g. plot 0 is associated with the five stations 0, 10, 20, 30, and 40). Ask yourself, "With which plot would station 500 be associated?" (The answer I'm searching for is, "Gee, NONE, otherwise plot 450 would have SIX stations associated with it!")

It is critical that your experimental design take this kind of thing into consideration. It is MUCH easier to "start on the right foot" than to try to correct problems later. For your own sake, please, *please* consult with others who have faced similar situations before you design new methods!

"MU level 4" blank). Since PLOT is the next level down, to define a PLOT you would create a **record** filling in *both* the "MU level 1" *and* the "MU level 2" fields (where "MU level 1" is [still] the transect [with which the plot is associated], and "MU level 2" is the plot number). Likewise, since STATION is one level down from PLOT (thus it's level 3), to define the station, create a new **record** with "MU level 1," "MU level 2," *and* "MU level 3" filled in (where "MU level 1" is [still] the transect, "MU level 2" is [still] the plot with which the station is associated, and "MU level 3" is the station number).[47]

Note that in these examples that STATIONS are "nested" under PLOTS, although the stations don't necessarily have anything to do with the plots. This hierarchical method, however, allows easy and flexible comparison of data of various types collected at various levels of resolution. It may save hours of grief if you discuss your monitoring design setup with someone with experience in this arena BEFORE you commit to a setup design.

To set up MONITORING UNITS, from the HOUSEKEEPING menu, click the MONUNITS button and enter the MONITORING UNITS that you will be using.

iii. Taxonomic infrastructure setup

This section covers setup of taxonomic parameters. Generally, this information is contained in the tables on the right-hand side of the **table relationship diagram**

47. Note that in Figure 2 the station number ("MU level 3" in this case) is the same as plot number ("MU level 2" in this case). This does not have to be the case--station numbers can begin at 0, 1, or any other number WITHIN PLOTS--but in the case of the GENR&TMON transects, it may sense to identify both the plot and the station by the meter mark from the beginning of the transect. Alternately, it may make more sense in your case to have the station number be the number of meters from the PLOT, in which case 6/40/40 would be 6/40/0, etc.

in Figure 1. *With two notable exceptions,[48] no one but your organization's Database Administrator should ever update this information (and then only in cooperation with the System Coordinator).*

Taxonomic nomenclature by its very nature is a hierarchical system, and is therefore easy to conceptualize in relational database format. Taxon definition is no exception to the "top-down" rule for setting up data: higher-level tables must be populated before data is added to lower-level (dependent) tables. The highest level of the taxonomic hierarchy in the Monitoring Database is KINGDOMS (see section 5.b.iii.(1)). Entries for this table must be entered first, then (in order) Phyla/Divisions, Classes, Orders, Families, Genera, Taxa, and Taxon Codes (see respective sections). Again, all updates to taxonomic information should be coordinated with the organization's Database Administrator.

[...Standards are important; why they are important.] [...Standards for the Monitoring Database for taxonomic nomenclature are: Bishop; taxon code list maintained by [???...]] [...] [...define standards [taxa [from the Manual, updates from Bishop {"most recently published"}]; taxon codes]; discuss importance of conformity to standards, using USFWS E/T updates as an example, & comparison of data; cite discussion of "alt" infrasp. code]

48. One exception to the Database Manager being the only person to change taxonomic information is that the Site Manager may be authorized to add "loc."-ranked taxa and their corresponding taxon codes. This should be coordinated with the Database Manager. Anyone who adds "loc." taxa should be thoroughly familiar with the procedure and its implications. The other exception is that the Site Manager may be authorized to add questionable or unknown taxonomic groups (e.g. Families, Orders, etc.) *as long as the first character of each added entity is a question mark ("?")*; the question mark as the first character distinguishes the entry as a user-defined entry (as well as indicating the uncertainty of the group assignment).

(1) Setting up KINGDOMS, PHYLA/DIVISIONS, CLASSES, and ORDERS

The original distribution of the Monitoring Database does NOT include accurate information above the FAMILY level. Information provided at the ORDERS level in the original distribution allows very broad categories of FAMILIES to be grouped together, not necessarily following rules of mutual exclusion enforced in a true taxonomic hierarchy. The KINGDOMS, PHYLDIVS, CLASSES, and ORDERS tables were last-minute additions to the Monitoring Database; the information provided in the ORDERS table is a holdover from the previous highest level of taxonomic classifications, "Family Groups." It was thought important to provide the infrastructure to allow full treatment of the taxonomic hierarchy so as not to create stumbling blocks for those who wished higher-level analysis of species groups (particularly entomologists, etc., who may not always collect species-level information).

Included at these high taxonomic levels may be generic categories such as "Non-biotic" in order to account for entities such as "bare ground" which may appear in field data. The values entered in all these tables should be standard among all organizations using the Monitoring Database. If you have special data analysis needs, of course, feel free to enter additional values, but be sure you understand the implications of doing so (e.g. your data may not be comparable to that of others).

Currently, the only ways to access these higher taxonomic levels are to (1) enter new information into the Order field in the FAMILIES form [Main menu, Housekeeping, Families], etc., so the system will "bounce" you up the hierarchy; or (2) open the forms directly via the PDOXWIN menu system [via the Project Viewer or Form | Open, then select the appropriate form]. Eventually, pushbuttons will be added to the Monitoring Database menu system to directly access each of these levels.

At the time of publication of this documentation, higher-level information (above Family) has not been entered at any Monitoring Database site. If there are any volunteers who wish to embark on this task (even if only for a specific group or groups), please inform the System Coordinator of your efforts so we can make your work even more valuable by distributing it to others.

(2) Setting up FAMILIES

The family of organism should be entered into the FAMILIES table, and associated with an ORDER (see previous section). The family should be entered as mixed case (*not* ALL UPPERCASE). If the family is not known (as often may be the case with insects, for example), you may enter a reference to some higher level (e.g. "?Odonata") or a nonscientific description (e.g. "?Bird family," as has been used somewhat whimsically in the past).[49] Note that if the family is changed in this table, the family for EVERY RELATED GENUS will change to the new family name. (To change the family associated with *only* a particular genus, change the value of that genus' Family field in the GENERA table [see 5.b.iii.(2)].)

(3) Setting up GENERA

The genus of organism should be entered into the GENERA table, and associated with a family (see previous section). Data which also may be incorporated at this level includes a genus description, page number on which this genus may be found in the standard reference work for this group of organisms, and a genus description source reference.

49. By convention, the first character of an unknown family should be a question mark (see previous examples); this indicates at a glance that the proper family name is unknown, allows the "unknowns" to be sorted together, and enables effective querying and/or filtering of data [see PDOXWIN documnetation for more info on filtering].

The genus should be entered as mixed case (*not* ALL UPPERCASE). The genus description can be of virtually unlimited length. The page number is with respect to the "standard" manual[50] for that organism type (e.g. the "standard" manual for vascular plants is assumed to be Wagner *et al.* 1990). The genus description source should be used to document the reference from which the genus description was extracted.

If the genus is not known (as often may be the case with insects, for example), you may enter some higher level (e.g. "?Odonata") or a nonscientific description (e.g. "?Dragonfly").[51] Note that if the genus is changed in this table, the genus for EVERY RELATED TAXON will change to the new genus name. (To change the genus associated with *only* a particular taxon, change the value of that taxon's Genus field in the TAXON table [see section [...] {"Setting up TAXA"}].)

(4) Setting up the INANAREA table

The purpose of the INANAREA table is to allow the user to select from a list of valid options (and force consistency of data entry) for the answer to the question "Does this taxon occur in [some particular area]?" Information of this type is tracked by specific fields in the TAXA table (e.g. "In HALE?" [Does this taxon occur in Haleakala National Park?] and "In HAVO?" [Does this taxon occur in Hawaii Volcanoes National Park?]).

There is no menu access to this table/form. All entries which are likely to be used are already included in

50. A "housekeeping" table entitled MANUALS may be added in the future with a tie-in to the taxonomic infrastructure in order to have this information directly available in the Monitoring Database.

51. By convention, the first character of an unknown genus should be a question mark (see previous examples); this indicates at a glance that the proper genus name is unknown, allows the "unknowns" to be sorted together, and enables effective querying and/or filtering of data [see PDOXWIN documnetation for more info on filtering].

the INANAREA table in the distribution of the Monitoring Database. These codes should remain standard among all Monitoring Database installations, and are considered to be static. In the event that you *must* add a value (please *don't* change the existing ones or their meainings!), enter the new value into the TAXA table in one of the linked fields; you will be prompted to add it to the INANAREA table. *Only the Database Administrator should do add INANAREA values, and then only in cooperation with the System Coordinator.*

(5) Federal E/T status (FEDSTAT table)

The FEDSTAT table contains standard codes for valid Federal endangered/threatened species statuses. [...rank] [...USFWSRAR table; why it's defunct]

Easy integration with the U.S. Fish & Wildlife Threatened & Endangered (E/T) Species list from the Honolulu office will be forthcoming. This will allow you to easily update the Federal E/T statuses of all taxa in your database based directly on information from that office. This feature will be distributed to all Monitoring Database sites when it becomes available. (Note: This will REQUIRE 100% compatibility of taxon codes and/or taxon names among all Monitoring Database sites who wish to use this feature. This is a prime example of the importance of conformity to proposed standards.[52])

(6) USFWSRAR

[omit this from **database (1)** altogether! mention its existence in the FEDSTAT description in case someone sees it in an older version & wonders what & where it is!]

(7) Critical habitat codes (CHCODES table)

The purpose of this table is to allow the user to select from a list of valid options (and force consistency of data entry) for the answer to the question "Is critical habitat

52. Refer to section [...] for information regarding proposed taxonomic nomenclature and taxon code assignment standards.

defined for this taxon (by the Federal government for E/T species)?" Standard codes include "?" (unresearched) and "NA" (not applicable, e.g. if the taxon has no Federal status), as well as "Y" (yes) and "N" (no).

(8) Origins (of taxa) (ORIGINS table)

"Origin" in this context refers to the origin of the taxon in Hawaii (e.g. endemic, Polynesian introduction, post-Cook introduction). The values in this table as they are provided with the original installation of the Monitoring Database are based on the convention established by Wagner *et al.* 1990.[53]

Note that general categories such as "Native" and "Nonnative" should NOT be used.[54] The set of categories defined should be *unambiguous* and *mutually exclusive*.

(9) Infraspecific designators (INFRASP table)

This is a "lookup table" for infraspecific designators (e.g. "var.", "f.", "ssp."). Its sole purpose is to ensure consistency in entry of valid data.

There are two special infraspecific designators included as part of the data distributed in the INFRASP table which are unique to the Monitoring Database: "loc." (location [of an unidentified taxon]) and "alt." (alternate [in-house] nomenclature). These designators have special significance to Monitoring Database users because ANY TAXON ADDED TO THE MONITORING DATABASE which is specific to your organization/site (i.e. "nonstandard" with respect to the entire Monitoring Database users's community) should ALWAYS use one of these two designators. (Refer to section 5.b.iii.(10) for details.)

53. Wagner[...]. Manual of Flowering Plants of Hawaii[...]

54. NOTE: At this time, there is NO security set up to exclude inappropriate entries.

(10) Taxa (TAXA table)

The taxon of each organism should be entered at this level, along with information corresponding to that taxon. [...put this somewhere: Although it is (obviously) not important for every species to have corresponding infraspecific entries, each INFRASPECIFIC taxon should ALWAYS have a separate entry at the species level.] [...put this in field-level descriptions: The page number on which this taxon may be found in the standard reference work for this group of organisms...].

The genus should be entered as mixed case (not ALL UPPERCASE); specific and infraspecific epithets should be lowercase. [...include in description of each memo field: ...can be of virtually unlimited length. The page number is with respect to the "standard" manual[55] for the respective organism type (e.g. the "standard" manual for vascular plants is assumed to be Wagner *et al.* 1990). The TAXON[...] description source should be used to document the reference from which the TAXON[...] description was extracted.

If the genus is unknown or unidentifiable (as often may be the case with insects, for example), you may enter some higher level (e.g. "?Odonata") or a nonscientific description (e.g. "?Dragonfly") as the genus.
By convention, the first character of an unknown TAXON[...] should be a question mark (see previous examples); this indicates at a glance that the proper TAXON[...] name is unknown, allows the "unknowns" to be sorted together, and enables effective querying. ALWAYS FOLLOW THIS CONVENTION so it will be obvious which genera in your database are NONSTANDARD (and therefore will NOT NECESSARILY BE COMPATIBLE with anyone else's data).

55. A "housekeeping" table entitled MANUALS may be added in the future with a tie-in to the taxonomic infrastructure in order to have this information directly available in the Monitoring Database.

If the species is unknown or unidentifiable, [...use the "loc." infraspecific identifier].

To make life easier for everyone, full taxonomic designations are *not* used for data entry of taxa. There are several reasons for this: shorter codes are easier to use in the field and for data entry, and (a technical point) single-field keys (e.g. taxon codes) are easier to manipulate than multiple-field keys (e.g. full taxonomic names) in the context of database structure and function. See section 5.b.iii.(10) for further details regarding taxon codes.

To change the genus associated with a particular taxon (one taxon only), change the Genus field of that record in the TAXA table. (To change the name of the genus itself--affecting all species in that genus--change the Genus field in the GENERA table [see section 5.b.iii.(2)].)

GENERALLY, YOU SHOULD *NEVER* CHANGE STANDARD TAXON NAMES (species and/or infraspecific) IN THE TAXA TABLE; doing so will make your data incompatible with others' data (e.g. USFWS) and will compromise the integrity of data exchange. The following information is provided in case you need to change information about "loc." or "alt." taxa, or if you are changing information per instructions from the System Coordinator. In any case, ONLY YOUR ORGANIZATION'S DATABASE MANAGER (or Site Manager, under the supervision of the Database Manager) should ever do this. With all that said, to change a taxon name, change the SPECIES, RANK, and/or INFRASPECIFIC EPITHET field(s) in the TAXA table. NOTE: Changing a taxon name will automatically change references to that taxon name in all **child tables**.[56] NOTE: Changing a SPECIES name in one

56. In the initial distribution of the Monitoring Database, only the TAXONCOD table is a **child table** to the TAXA table.

record does NOT change the corresponding species name in (taxonomically) related records (e.g. infraspecific taxa of that species); these must be changed MANUALLY.

(11) TAXONCOD

[...Note that if the TAXON[...] is changed in this table, the TAXON[...] for EVERY RELATED taxon will change to the new TAXON[...] name.]

GENERALLY, YOU SHOULD *NEVER* CHANGE STANDARD TAXON CODES IN THE TAXONCOD TABLE; doing so will make your data incompatible with others' data (e.g. USFWS) and will compromise the integrity of data exchange. The following information is provided in case you need to change information about "loc." or "alt." taxa, or if you are changing information per instructions from the System Coordinator. In any case, ONLY YOUR ORGANIZATION'S DATABASE MANAGER (or Site Manager, under the supervision of the Database Manager) should ever do this. With all that said, to change a taxon code, change the TAXON CODE field in the TAXONCOD table.

c. Collecting field data

This section outlines the recommended procedures for collecting field data for entry into the Monitoring Database.

This may seem to overlap with the realm of responsibilities of the field biologist. Indeed, it does: it is the managing field biologist who is ultimately responsible for the quality of the results based on data in the Monitoring Database. This section provides practical guidelines to ensure that after the field data is collected, the data entry *and analysis* phases also go smoothly. Ultimately, this contributes to operational efficiency and quality data.

i. Before you go into the field...

[...]

(1) [...stuff to put in this section:]

have field sheets set up properly; REVIEW PROCEDURES, including data entry sheets--what TO do, what NOT to do; education of field crew RE: database [show it to them, etc.; review THEIR data set types prior to going into the field]

(2) Review the Taxon Codes "duplicates" list

Before going into the field, each crew member should review the Taxon Code "duplicates" list (the list of taxa for whom the 3+3 method of creating the taxon codes produces duplicate codes). (This list may be obtained by clicking the "DUPLICATES" REPORT button on the TAXONCOD form.) With these taxa fresh on your mind, it will be easier to remember that while collecting data you should distinguish *on the field data collection sheet* between the "duplicate" taxa, *even if you will encounter only one of the alternates* in that data set. (Believe me, this will make your life easier when you begin to enter data!)

(3) Review previous data sets

"Prep sheets" can be produced from data which can be stored in the Monitoring Database, including maps of locations that will be visited; notes about terrain, individual plots/stations, etc.; rare plants found there or nearby previously; species lists from prior data collection efforts; etc. For example, a composite species list from all prior years at a site list could be reviewed and items such as difficult taxonomic distinctions, etc., could be discussed with all crew members (especially new personnel) prior to leaving for the field.

(4) Use properly-prepared field data sheets

Ensure that you have properly-prepared field sheets, i.e. that ALL blanks on the sheets correspond to a field in the Monitoring Database (so all info collected can be stored in the database), and that there is a blank for every piece of information required by the database. (Your Database Administrator and Site Manager should

both review the field sheets to ensure that these criteria are met.) Additionally, if you will be collecting data at known permanent locations, you may wish to create forms in advance with known information already entered (e.g. transect, plot, station info). The field data sheets themselves may be prepared directly from the Monitoring Database by creating reports from existing data (including lists of taxa expected to be found at each plot/station, if this is deemed desireable).

(5) Review data collection procedures

Ensure that the field crew understands what are and what are NOT acceptable entries on the field sheets, and that they understand how to resolve problems which may arise in the field and to forestall data entry problems.

ii. While in the field...

[...] [use of data entry sheets; methods {data set types} created "on-the-fly";...]

(1) Code data collection sheets properly

Be sure to use ONLY the standard codes for collection of field data. If you feel that other information is necessary to accurately record your observations, please record this information as NOTES. Also, be sure to indicate whether each NOTE on the data collection sheet refers to only one taxon, the entire plot, the entire station, etc.

Be sure, also, to follow "exception procedures" properly. For example, it is probably not reasonable to have the field crew remember and enter the exact taxon code for the "duplicate" taxa (e.g. CyrGra020 for *Cyrtandra grayana*). However, if the field crew realizes that this is a taxon which will cause a coding conflict later (even though they may not see *Cyrtandra grandiflora* or *C. grayi*--which create the conflict--during their trip), it would be good to indicate unambigously which taxon is being recorded. (This indication should occur on EACH

FIELD SHEET for which this is a factor [remember that data entry occurs one sheet at a time, and may not occur in the same order in which the taxa were encountered in the field].) In other words, "CyrGra" should be AVOIDED as an entry on a field sheet UNLESS the interpretation is provided ON THAT FIELD SHEET. (Translation to the correct taxon code will be done after returning from the field [see section 5.c.iii.(2)], but the person who analyzes the data sheet must be able to determine which taxon was recorded.)

iii. When you return from the field...

[...]

(1) [...stuff to put in this section:]

; add to DATACOLS as needed; create user-defined tables/etc. for *ad hoc* methods devised in the field;

(2) Prepare field data sheets for data entry

review taxon codes dups list, then REVIEW & CODE field sheets (creating new *loc.* taxon codes as necessary) [...]

(3) [...]

d. Entering field data

This section outlines the recommended procedures for entering field data into the Monitoring Database. (It is assumed that the Monitoring Database has been set up for your site/organization in accordance with section 5.b, and that the procedures in sections 5.c.i, 5.c.ii, and 5.c.iii were followed.[57])

i. [...stuff to put in this section:]

set up DATACOLS; verify all TAXON CODES (incl. review of taxon code "dups"); data-checking routines (xref sums by column, row)

57. Yeah, right!

ii. [...]

iii. [...]

iv. [...]

CHAPTER 6. Other stuff to include

Somewhere, include:

- system requirements
- installation procedures
- where to go for tech support (as a separate section?)
- how to use table lookup
- how to use auto-add (tricks: to get to an auto-add form to change infor instead of just the read-only "lookup," enter a dummy value (e.g. "XXX") into field, then say "yes, add", then cxl changes (e.g. Alt-Bkspc), then edit; when returning to original form, press <Esc> to return the original value in the field you changed [to "XXX" (the dummy value)] ot its original value.)
- You can also create external cross-reference tables (see 3.b.ii.(3)) to which can link and therefore analyze by certain groups of existing family groups (e.g. you could combine Plants, Lichens, Fungi, Bacteria).
- when RI lookup fails: (1) technical reason [combined--i.e. >1 field--"lookup" fails]; (2) how to deal with it: you'll get a msg such as...; then: a button along the bottom edge of each form is available in each case that this may occur.[58] By clicking one of these buttons, you will be able to directly access any data upon which the data in the current form is dependent.[59]
- I refer to the **table relationship diagram** of the system as the "magic chart," because questions about data relationships may be quickly resolved by reference to it (e.g. when creating queries or entering data which is dependent upon entries in other tables).

58. [list all these forms here...]

59. NOTE: It is no coincidence that the table to which each of these buttons refers will be exactly one level above that of the current form's table in the **table relationship diagram**.

INDEX