GenBank

GenBank at Los Alamos

User Manual
Training Guide
and
Reference Manual

For the
ASCII AWB

GenBank
at Los Alamos

User Manual
Training Guide
and
Reference Manual

For the
ASCII AWB

This manual is produced by the GenBank Project at Los Alamos National Laboratory, Los Alamos, NM 87545. It describes software and procedures designed and implemented under funding from the National Institutes of Health.

Inquires may be sent to genbank@t10.lanl.gov
Software concerns: gb-software@t10.lanl.gov
Submission information: gb-sub@t10.lanl.gov
Submission updates: update@ncbi.nlm.nih.gov
Manual comments and suggestions: gb-manual@t10.lanl.gov

This manual is available for anonymous ftp from genome.lanl.gov in the directory pub/doc/AWB.ascii

Documentation: George C. Reese, Gifford M. Keen, Paul Gilna, and Michael J. Cinkosky

Acknowledgements: We are grateful to Patricia Reitemeier for her careful editing and to the annotation staff: Reid Hayhow, Marianne Luchini, Carol Hager, Will Fischer, Lisa Hollis, David Brown, Michelle March, Chris Skalka, and Fiona Jordan for their comments and suggestions. The AWB software is maintained by Gifford Keen, David Crowley, David Rider, Robert Sutherland, and Charles Troup.

Latest Revision: March 23, 1993

Previous Version: January 15, 1993

LA-UR#

Table of Contents

Chapter 1 General Introduction

The GenBank project at Los Alamos is funded through an interagency agreement with the National Center for Biotechnology Information at the National Library of Medicine to collect nucleotide sequence submissions from the biological research community. This work includes the processing of data received in several different forms (e.g., Authorin submissions, submission forms, and direct database access) as well as maintenance and quality control on those submissions. This manual explains the procedures involved in that work for both Los Alamos GenBank staff and off-site users.

The GenBank database stores annotated DNA sequences. This manual contains the procedures for depositing these sequences into the database. There are two ways to do this. Either the sequence arrives at GenBank as a submission and is entered by the database staff or the sequence is directly entered by an off-site user. The Annotator's WorkBench (AWB), which is a database browsing and editing tool, is used in both cases.

This manual is for GenBank staff and off-site users of the GenBank database at the Los Alamos National Laboratory. It contains an introduction and tutorials for AWB, as well as procedures for entering sequences either as submissions or as data directly deposited by an off-site user. Instructions for all of these are found in Chapters 2 through 4. The introduction to AWB is in Chapter 2. Instructions for submission handling are in Chapter 3. Instructions for entering sequence information are in Chapter 4. Off-site users should look at section 4.3 for instructions on entering a sequence.

In addition, the manual describes various in-house curatorial tasks that are part of maintaining the database, as well as the procedures and conventions for annotating sequences. The procedures for annotation and review are in Chapters 5 and 6. The description of in-house curator's tasks is in Chapter 7. The sections of Appendix A describe Annotation conventions.

There are two reference chapters, which contain specific descriptions of AWB and other utility programs. The AWB Reference (Appendix B) contains a complete list of all the forms, fields, and commands in AWB. The Utilities Reference (Appendix C) contains descriptions of other (non-AWB) software utilities used by database staff.

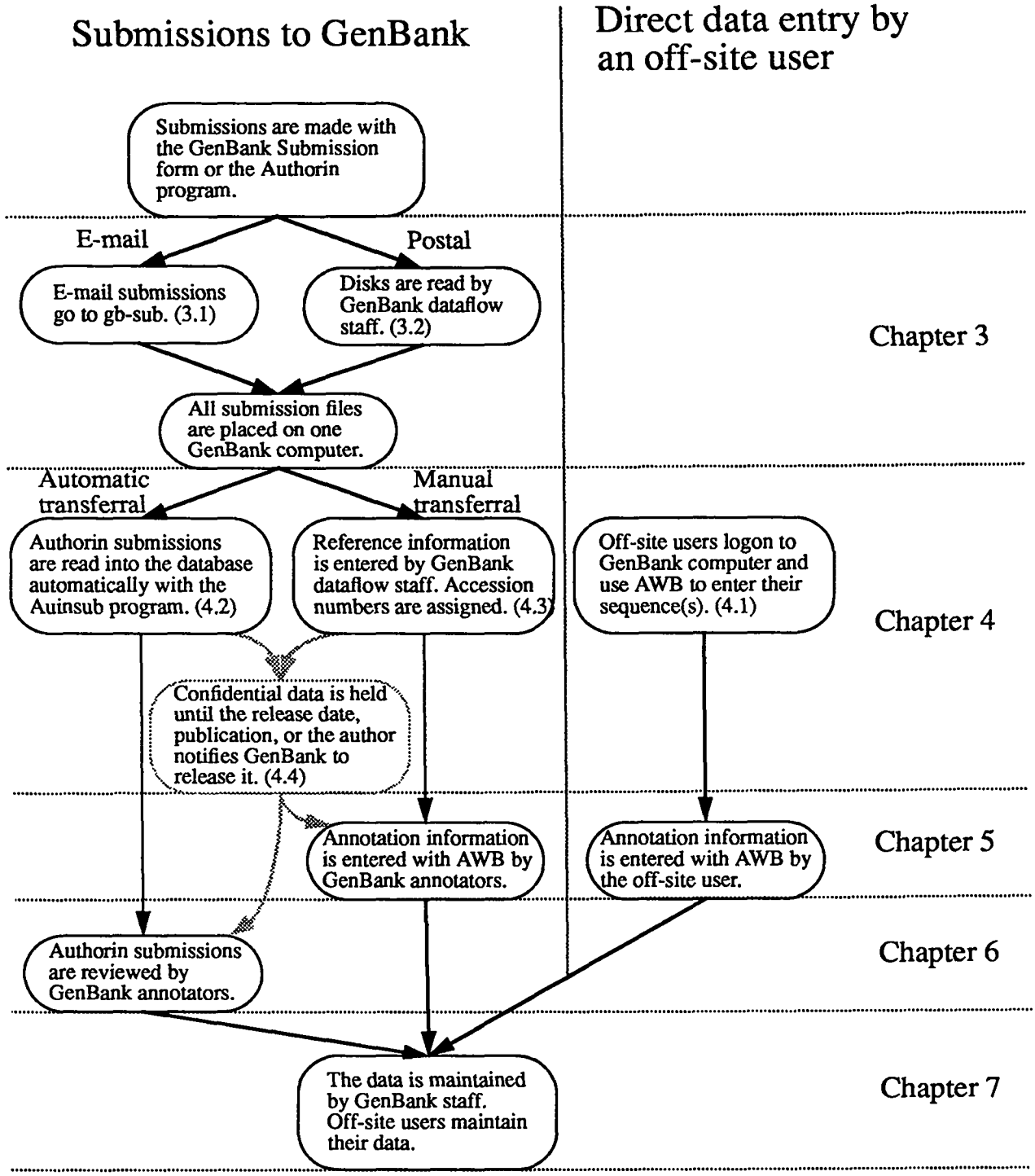

Figure 1-1 The two paths by which data enters GenBank. (Chapters 1 and 2 are introductory; the appendices, especially A and B, are referred to throughout.)

1.1 How to Use This Manual

This manual is designed to be useful to a variety of people, both in-house and off-site. Users of the manual with differing needs will find only certain chapters and sections useful.

Off-site users: A paragraph like this at the beginning of each chapter will describe it, and the paragraph may be followed by a bulleted list that explains which sections are most useful. Here is a general list of the sections most useful to off-site users entering a sequence in GenBank:

- Chapter 1 and Chapter 2 contain introductory information necessary to get started with AWB.
- Sections 4.1 and 4.4 contain instructions for entering a sequence.
- Chapter 5 contains instructions for entering the annotation information that accompanies the sequence.
- Appendix A describes annotation conventions.
- Appendix B is a reference chapter describing all AWB commands, forms, and fields.

GenBank staff: A paragraph like this at the beginning of each chapter will describe it, and the paragraph may be followed by a bulleted list that explains which sections are most useful. Here is a general list of the most useful sections for GenBank staff.
For GenBank dataflow staff:

- Chapter 1 and Chapter 2 contain introductory information necessary to get started with AWB.
- Chapter 3 contains instructions for processing all submissions.
- Sections 4.2, 4.3, 4.4, and 4.5 contain instructions for entering and updating sequence data submitted to GenBank.
- Appendix A describes annotation conventions.
- Appendix B is a reference chapter describing all AWB commands, forms, and fields.
- Appendix C is a reference chapter that contains descriptions of the non-AWB utility programs used to find information and accomplish common tasks.

For GenBank annotators:

- Chapter 1 and Chapter 2 contain introductory information necessary to get started with AWB.
- Chapter 5 contains instructions annotating a submission. (GenBank Submission form)
- Chapter 6 contains instructions for reviewing Authorin submissions.
- Appendix A describes annotation conventions.
- Chapter 7 contains information on in-house curators' tasks.
- Appendix B is a reference chapter describing all AWB commands, forms, and fields.
- Appendix C is a reference chapter that contains descriptions of the non-AWB utility programs used to find information and accomplish common tasks.

1.2 Conventions

- **Bold face** is used for the tutorial steps and for the first time that a special term is used.
- `Courier` is for specific keystrokes that you should type. For example, a typical instruction line will say, "Type `\eo` to see a list of options."
- Italics are used in the AWB reference chapter for terms being defined. Italics are also used to indicate text you must replace. For example, "Type *`filename`* hup." Paragraphs that contain special notes are also in italics.
- Return refers to the carriage return on the keyboard.
- Tab refers to the key labelled Tab on the keyboard.
- Backspace refers to the key labelled Backspace or Back Space on the keyboard.
- Esc refers to the key labelled Esc or Escape on the keyboard.
- The names of fields, forms, entities, commands, menus, and tables will be capitalized. For example, "Submission" refers to the Submission entity or the Submission form; whereas "submission" refers to a submission of data to GenBank.
- `^X` means hold down the key labelled Control and press the letter x.
- `^T` means hold down the key labelled Control and press the letter t.
- `^O` means hold down the key labelled Control and press the letter o.
- Some instructions involve a pointing device. This device is assumed to be a three-button mouse. Since mouse functions can be customized, the mouse buttons are referred to by function rather than by location. Here are the functions in terms of the default location on the mouse.

  The default location of SELECT is the left mouse button.

  The default location of ADJUST is the middle mouse button.

  The default location of MENU is the right mouse button.

  A typical instruction involving the mouse is "click SELECT on the Find button."

1.3 The Arrangement of the Tutorials

The tutorial sections are designed for repeated use. That is, you may be following the instructions several times. As you become familiar with the procedures, the detailed instructions may seem obvious. To allow for this, the tutorials are arranged with major steps of a general nature that are then broken down into action steps that explain the details. There may be elaboration following either type of step.

The instructions follow the sample arrangement below:

5. **Major steps are numbered and in bold face (as in this example). Major steps contain instructions such as, "Fill in the Person Form."**

   Sometimes there are unmarked paragraphs (such as this one) below a numbered step that contain elaboration of the step.

   - Paragraphs marked by an open square (such as this one) contain specific actions that need to be taken and that may include keystrokes or mouse instructions. An example is "Type `\eel` to call up the ENTITY list."

     Below the action steps may be unmarked paragraphs (such as this one) that state the result to expect. For example, "An entity list appears." These paragraphs may also contain elaboration of the action steps.

1.4 The Priority System

The priority system is designed to ensure that the annotation staff is on track with respect to a 10-day turn around on submissions—from arrival at GenBank to distribution.

The priority I tasks are done first, but that does not imply that they are the most important. In fact, the latter priorities are of greater importance to the database as a whole. Priorities I and II focus on individual submissions and the effort to ensure accurate entry into the database of its many aspects: sequences, features, references, etc. Priorities III and IV on the other hand, focus on a particular **attribute** in the database (e.g., Person names) and look at that attribute throughout the database in an effort to maintain database-wide accuracy.

PRIORITY I: Annotation/Review of Submissions

All submissions that are coming out of dataflow for release are queued for annotation and placed in the annotation drawer. On each day 10% of the total number of submissions are withdrawn and placed out for annotation in the priority I basket. Submissions are selected one at a time from the basket. The goals for this priority are met when the priority I basket is empty.

PRIORITY II: Database Update/Error Correction

This consists of the following tasks: handling author-supplied corrections and updates, and problem distributions. These will be culled from the current update queues and placed in the priority II basket to be handled by annotators on the next day.

PRIORITY III: Software Error Lists, Community-Supplied Error Corrections, and Entry Corrections

Whereas Priorities I and II consist of discrete units of work (papers), the tasks in this area are broader in scope and in the main consist of lists of corrections or additions, for example, a list of duplicate keywords that need to be combined or a list of feature boundaries that are incorrect.

PRIORITY IV: Annotation Curation Tasks

Specific annotators act as designated curators for assigned areas of the database, working primarily to monitor, review, and ensure consistency of new data added in their assigned domain, yet also using their speciality to improve their chosen area of the database through retroactive curation.

The primary difference between tasks in III and IV is that the tasks in III represent "enhancements" (corrections, additions, updates) that we know, or have been told about, while tasks in the Priority IV represent improvements that we actively seek out. Tasks being set include those discussed in Chapter 7.

Chapter 2 Introduction to the Annotator's WorkBench

Off-site users: Read all of this chapter for introductory information about the Annotator's WorkBench (AWB).

- Section 2.1 discusses the structure of the database and describes the AWB interface.
- Section 2.2 is a tutorial in which you enter your name in the database.
- Section 2.3 describes a typical route through the forms.

GenBank staff: This chapter contains introductory information about the Annotator's WorkBench (AWB). Read all of this chapter if you are new to GenBank.

- Section 2.1 discusses the structure of the database and describes the AWB interface.
- Section 2.2 is a tutorial in which you enter your name in the database.
- Section 2.3 describes a typical route through the forms.

2.1 The Structure of the Database and AWB

The GenBank database is maintained as a relational database, providing access to information while avoiding redundancy. It contains nucleotide sequences and other data associated with the sequences. Associated data includes information about the scientists who are submitting (or directly entering) the sequences, the biological features of the sequence, the source of the sequence, the journal in which the sequence information will be published, etc.

Entities

Primary data items are called **entities**. An entity is a distinguishable, cohesive collection of data that models a real world object (i.e., Sequence, Person, Feature, etc.). For example, each Person entity has **attributes**: First name, Last name, Address, Phone number, etc. These attributes help define the particular Person entity.

Another example of an entity is a **Feature**. Every Feature entity contains information about a biological feature of a nucleotide sequence. Attributes of a Feature entity include the type of Feature, called a **Feature Key** (e.g., a coding sequence, poly-A signal, CAAT box, etc.), the span of the Feature (e.g., base pairs #102 to #253), any product for which the Feature codes, and any special comments or qualifiers that further explain the Feature. And, as with all entities, each Feature entity has an ID number that distinguishes it from all other Feature entities.

Entities are related to each other. In AWB, these relations, or connections, are called **links**. When an Address entity is connected to a Person entity, we say the Address is "linked" to the Person.

ID Numbers

To ensure individual uniqueness within an entity set, each entity is given an ID number. The number uniquely identifies the entity from all other entities of the same type. For example, the Paper entity displayed in Figure 2-1 has the ID number 153708. That number (called the **Reference ID** or **rf_id**) distinguishes that particular Paper from all other Papers in the database.

The ID number for each Sequence in the database is called its **accession number**. The first character of an accession number is always a letter. Thus, it is easy to distinguish it from other (all numeric) IDs. An accession number is assigned automatically, by software, when a new Sequence entity is created.

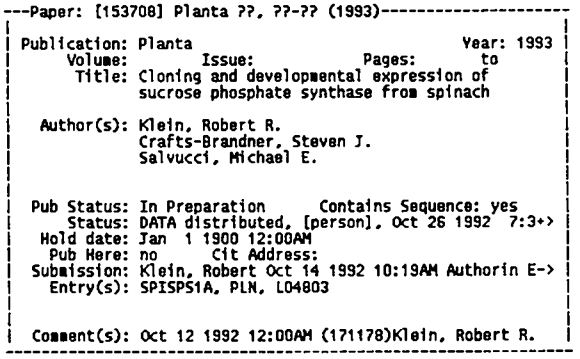

Figure 2-1 The Paper form.

2.1.1 Browsing and Editing Entities with AWB

AWB is a tool for browsing and editing entities in GenBank. It can create new entities, edit existing entities, and create links between entities.

Each type of entity in the database corresponds to a **form** in AWB. A form is a template on which to display the attributes of a particular entity. The attributes appear in the **fields** of the form. Field descriptions precede colons in the forms. There are several types of fields: single and multiple link fields, toggle fields, and text fields.

Identification of the type of field appears in the **status bar** at the lower, left corner of the display. "^X to expand" indicates that the highlighted field has a link to another entity. "^T to toggle" indicates that the highlighted field is a toggle field. Otherwise, the highlighted field is for text.

Figure 2-1 shows a Paper entity displayed in the Paper form. It contains information about the journal article (or book, or thesis, etc.) in which the sequence is referenced. The ID number of the entity appears at the top of the form.

Single and Multiple Link Fields

Single and multiple link fields contain the connections between entities. They display a summary (or summaries) of the information in a linked entity. For example, the summary of a Person is the first and last name and the middle initial. The summary for a Publication is its abbreviation. In general, the summary is information that concisely indicates the nature of the entity.

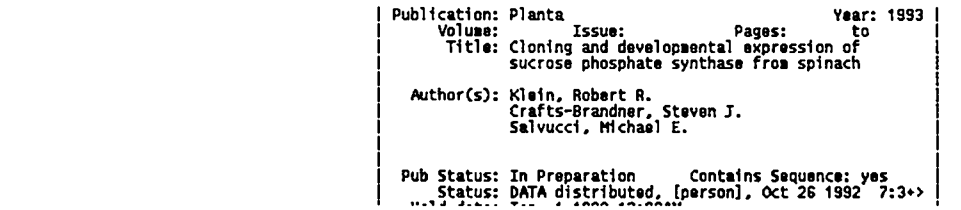

Figure 2-2 The Publication and Author(s) fields are examples of single and multiple link fields respectively. The Status field may have multiple links, even though it appears there is room for only one.

To follow the link to another field, highlight the field with the dark band and press ^X. Pressing ^X tells AWB to follow the link to another entity.

Some fields allow only one link to be made, while others may contain multiple links. Figure 2-2 shows an example of each field. The Publication field is for a single link and the Author(s) field may have multiple links.

Some fields appear to have room for only one link, when in fact, AWB will allow more than one link (e.g., the Status field in Figure 2-2). A plus (+) or minus (-) sign in the field indicates that there is more to be viewed. (Use the Tab for a + and the Backspace key for a - to see the rest of the summaries.) If a field is for single links only, AWB will generate an error message when you try to create another link.

When a single or multiple link field does not contain any links, the sign <no link> appears in the field. This means no link has been created so far. \eel (Edit: Entity: Link) is the normal command to create a link.

Toggle Fields

Some of the attributes displayed in fields come from a defined set of options. These are toggle fields. By pressing ^T you may change the contents of the field from one option to the next. \eo (Edit Options) will display the complete list. When the list is displayed, use the Tab key to highlight an option and Return to select it.

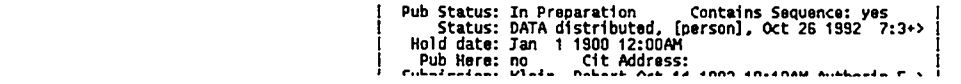

Figure 2-3 The Pub Status field is a toggle field. ^T will toggle the field to Published, Unpublished, In Preparation, In Press, or Submitted for Publication. The Contains Sequence field is also a toggle field. It has a default setting of yes, but may be toggled to no.

Text Fields

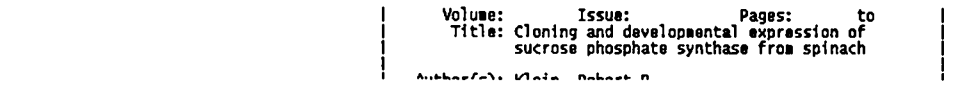

Figure 2-4 The Title field is an example of a text field. Information may be typed directly into a text field. The Volume, Issue, and Pages fields are also text fields.

Text fields allow you to directly type the information onto the form. This is necessary where the information will be connected only with that particular entity. The Title field in the Paper form is a text field. After you highlight the field, you may type text directly into it. The text to be placed in some fields may be much longer than the space in the form. (For example, a Sequence is usually much longer than the space in the form.) The command `\et` (Edit: Text editor) places you in the vi text editor where more of the text can be seen and the vi commands may be used to edit the information. Type `:q` to quit the text editor.

2.1.2 Moving through Forms and Fields

As we have previously said, entities are connected by links. Following a link moves you to another entity. For example, the Sequence form will have Source and Feature fields. If entities are linked to these fields, the user may expand one of the fields to either a Source or Feature form.

When information is entered with AWB, the various forms are filled out, and links are created between the entities. (This may be done manually or through programs, like Auinsub, that automatically enter the information.) Once the links are created, you may use AWB to follow the links from one entity to another. Figure 2-5 illustrates one route through some of the forms. A Sequence form is in the foreground. The Sequence entity has been linked to an Entry entity, which was linked to a Paper entity, which was linked to a Worksheet. From the Sequence form, the user could expand either the Source or Feature field to another form.

When the attributes of an entity are edited, the `\es` (Edit: Save) command will save the changes made. However, AWB automatically saves the changes when you quit a form. To quit a form, type `\qf` (Quit: Form). (Pressing the Esc key will also work, but not as quickly.)

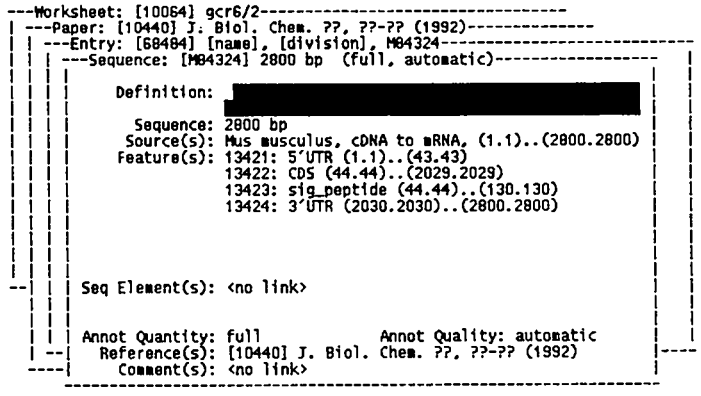

Figure 2-5 The Sequence form. This entity is linked to an Entry, which is linked to a Paper, which is linked to a Worksheet. The Source field can be expanded to the Source form, and each of the Features can be expanded to a Feature form.

To move within a form, use the Tab and Back Space keys. This makes the highlight move from one field to another. You may also use the arrow keys.

The Return will also move the highlight band forward through the form; however, the main use of Return is to select an item from a **list.**

Esc quits a form, **locator box,** list, or **menu.** If you find yourself in a form, list or locator box that you did not intend to bring up, Esc will take you back to the previous form.

\qf (Quit: Form) is the quickest way to exit a form.

2.1.3 Issuing Commands

Commands in AWB are issued through the **menu bar** at the top of the AWB display. Each word in the menu bar is the top of a pull-down menu that goes with it. To access the menu bar type \. You then execute commands by typing the first letter of the word options in the menu. For example, to see the worksheet options type \w (do not press Return). The worksheet pull-down menu will appear. Some items on the menu have an ellipsis (...) following them. Any menu item that is followed by three dots has another menu (a submenu) attached to it.

Figure 2-6 shows the Edit menu with the Entity submenu. \eel (Edit: Entity: Link) is the command to link an entity in to a field.

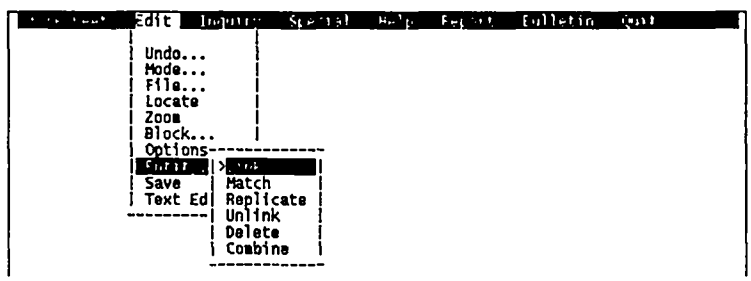

Figure 2-6 The Edit menu with the Entity submenu. The command \eel chooses Edit from the menu bar, Entity from the pull-down menu, and Link from the submenu.

Some commands are complete after typing \ and the character string of first letters. Others will require you to select from a list and/or give more information in a locator box.

Entity Lists and Options

A list will display a group of entities in the database. A list appears, or may be called, whenever there are multiple options from which to choose. For example, when the command \eel is issued from the Worksheet ENTITY field, a list of entities appears. The user then highlights the desired item and hits Return. The Figure below shows the Entity List.

There are several ways to move the highlight in a list. The Tab and Back Space keys will move the highlight band down and up the list. The arrow keys will also move the band through the list. The quickest way to highlight a list item is to type the first letter, or first few letters of its name.

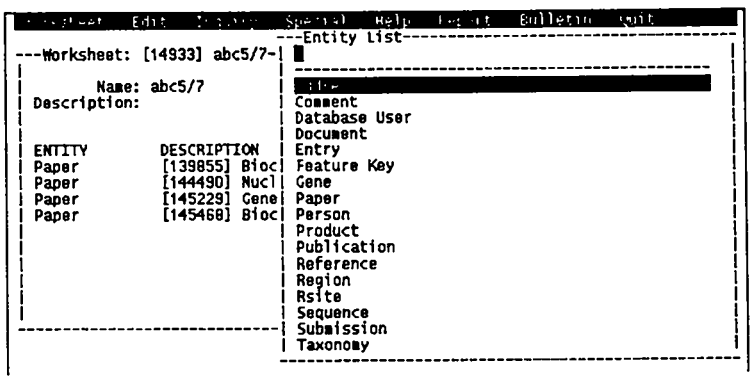

Figure 2-7 The list of entities. The command \eel (Edit: Entity: Link) brought up this list. It is a general list of entities. Use the Tab key, or type the first few letters of the word; then press Return to select an item from the list.

Locator Boxes

When you choose an entity from the Entities list, AWB presents you with a locator box. Locator boxes find specific entities in the database. For example, if you choose "Paper" from the Entity List, the Paper locator appears. The box has fields to be filled out that help specify the item you are looking for.

Enter in only as much information as seems necessary to match just one entity. When you press ^X (Control x), AWB finds entities that match the information. The more specific the information, the faster the search will go. Entering the database ID is the fastest way to find an entity. If the data you enter matches more than one entity, you will get a list of entities from which you can choose.

All text fields in locator boxes accept the asterisk (*) as a wildcard matching any string of characters. For example, in the Publication Locator box if you enter `*ell*`, AWB produces a list of GenBank publication abbreviations that have the string "ell." This includes "Cell" and "Mol. Cell. Biol."

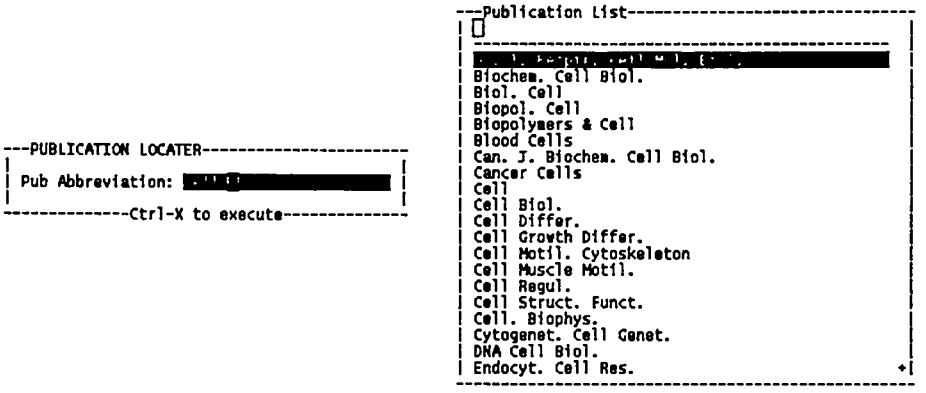

Figure 2-8 A wildcard search with the Publication locator. *ell* brought up the list of publication abbreviations on the right. Notice the plus (+) sign at the bottom right of the Publication list. It indicates that there is more of the list to be viewed.

Use wildcards often (but carefully), especially in place of hyphens and articles (items that may or may not be in the database). If the sought-after item does not appear on the first attempt, try again with different wildcards. Be aware that using wildcards in a locator without enough information may generate a huge list (e.g., *protein* in the Product locator).

> *Note: Placing only an asterisk in a field could have disastrous consequences; the database clogs while AWB creates a list of thousands of items.*

The goal is to be certain the entity is not already in the database before creating a new entity. This is particularly important with Persons, Publications, Keywords, and Products. All of these may vary in their appearance. For instance, John K. Smith may be in the database as J. K. Smith III. It is best to search for this name with J*Smi*. As another example, The University of Florida could be abbreviated in the database in a variety of ways (Univ. of Florida, University of Florida, The University of Florida, etc.). A good search with wildcards would be *U*Flor*. The disadvantage of using too many wildcards is that the list of entities may be very large. However, with experience and thoughtful use, wildcards can make more rapid searches and help keep redundant entities out of the database.

The locator boxes are case sensitive. Thus, `Nat*` will bring up a list that includes Nature; `nat*` will not find any journal abbreviation, since all of them begin with a capital letter.

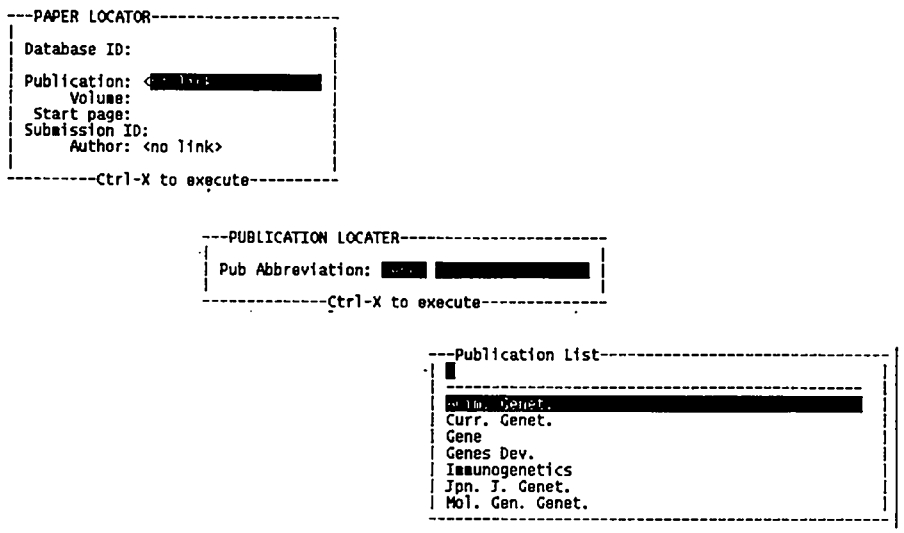

Figure 2-9 Using Locator boxes to find a Paper entity. Typing \eo with the Publication field of the Paper locator highlighted brought up the Publication locator. The Publication List was found using the wildcard search *ene* which located the journal abbreviations containing the character string ene.

The database ID number is, by itself, enough to specify a unique entity. If you know the database ID (e.g., the reference ID of a paper, the accession number of a sequence), enter just that. This is the quickest way to find an entity.

If the locator box can not find an entity to match the information, it will ask if you want to create a new entity. Type y for yes or n for no. If you type y, a new form will be linked in to the field.

If you want to link in a new, blank form, press ^X on a blank locator box. A box appears that asks if you want to create a new entity. Type y. The new form will have a unique ID number, and only the default values will appear in the fields.

Otherwise they are grayed out. Double clicking on an entity has the same effect as selecting the entity and choosing Follow Link from the pop-up menu.

2.1.4 Worksheets

A Worksheet is the usual starting place in AWB. The difference between worksheets and other entities is that Worksheets provide a place for a user to put other entities. In other words, starting with a Worksheet, one can link to it other entities on which to work.

The Entity field of the Worksheet contains a list of the types of entities linked to it with a brief description of the entity in the Description field. Press ^X on a highlighted entity to expand to its form.

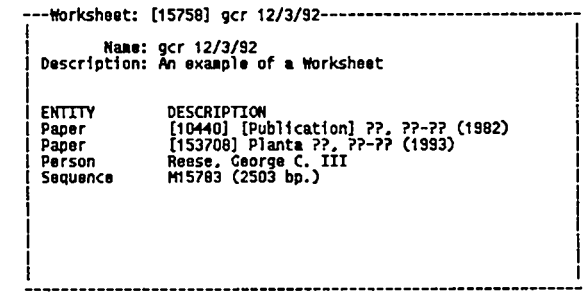

Figure 2-10 A Worksheet. Two Paper entities, a Person, a Sequence are linked to the Entity field.

You are now ready to begin working with AWB. The next section is a tutorial introduction.

2.2 Getting Started

What follows is a hands-on introduction to AWB. You should read section 2.1 before beginning this section.

In this tutorial, you will login to AWB and create a Person entity that contains your name and address. You will use a locator box to check the database to be sure that there is not already an entity with your name.

1. **Set the environment variables.**
   - Type `setenv LIBDA_DB genbank`.
   - Type `setenv DSQUERY SYBASE_SYNAPSE`.
2. **Login to AWB.**
   - From a Shell tool, type `awb` and press Return.

     Use lower case. UNIX is case sensitive.
   - Enter your login name and Return.

     Your login name and password for AWB may be different from your computer login.
   - Enter your password and Return.

     The initial display appears.

     If you enter an incorrect login name or password, you will see the a message, "Login incorrect." If this happens try to login again. If you are repeatedly unable to login, send a message to gb-software@temin.lanl.gov.

   After a preliminary message ("Starting the Annotator's WorkBench...") the AWB display takes up the whole of a window

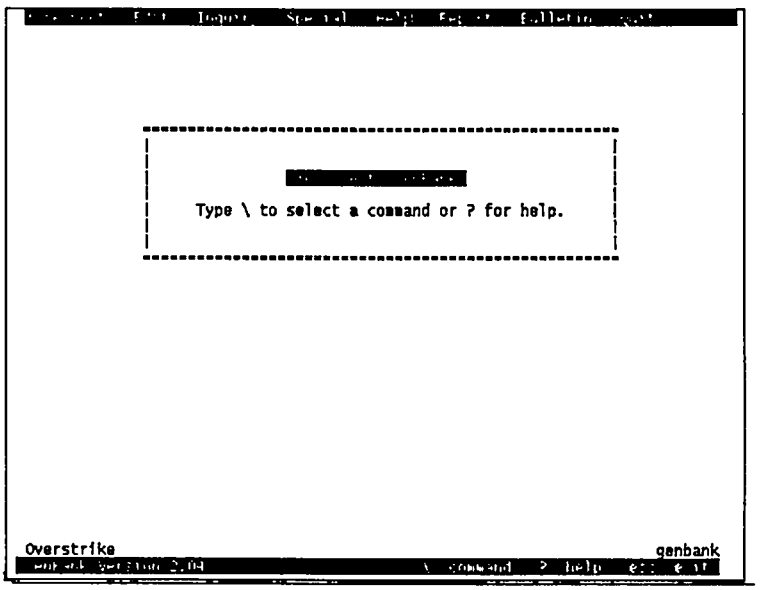

Figure 2-11 The initial display.

3. **Notice the menu bar at the top and the status bar at the bottom.**

   At the top of the window, the **menu bar** displays the commands that the workbench accepts. Each of the words represents the top of a pull-down menu, from which the details of the command are selected.

   Along the bottom of the window runs the **status bar.** At any given point during a WorkBench session, the left side of this bar will contain some information about what activity is happening or what sort of input is expected. As you move through the different forms in the database and issue commands, you should notice the contents of this field changing. It says, "GenBank Version 2.04." (AWB is always being updated; the current version will have a higher number than 2.04.)

   On the right side of the status bar are reminders of three useful keystrokes.

   - \ is the first keystroke for issuing commands. It will activate the menu bar at the top of the screen.
   - **?** may bring up some information about the current field.
   - Pressing Esc will exit the current form.

   Above the status bar on the left is the current editing mode. You may toggle through them with ^O. There are three modes:

   - **Overstrike** will type over any characters in a text field.
   - **Insert** will insert typed characters in between characters already in a text field.
   - **Readonly** will not allow the insertion of text into a text field.

4. **Open a Worksheet.**

- Type \

  This activates the menu bar . Notice that the menu bar has a highlight on the word "Worksheet"

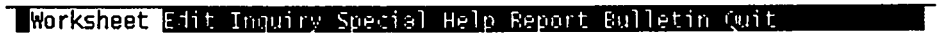

Figure 2-12 The menu bar. Typing \ has highlighted "Worksheet".

- Press Tab to move forward on the menu bar. (Or use the right arrow.)
- Press BackSpace to move backwards. (Or use the left arrow.)

  As you move along the menu, the commands become highlighted. To select the command, type its first letter. Typing the first letter of the command both highlights it and selects it. Use Esc to unselect an item.

- Type w.

  This will highlight and select the worksheet command. (Instead of typing w, you may also tab to highlight Worksheet and hit the Return key.) The Worksheet pull-down menu will appear as in the Figure below.

  The pull-down menu lists the actions that can be performed on worksheets. Typing the first letter will select that action.

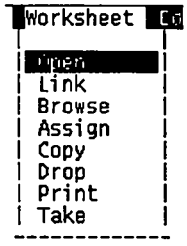

Figure 2-13 The Worksheet pull-down menu.

- Type o.

  The Worksheet list appears. It contains the list of Worksheets in your space. If this is the first time you have used the AWB, then the Worksheet list will be blank. You can type in the name of a new Worksheet to create a list. Your initials and the date would be a good Worksheet name (as this should make it unique). For example, `gcr 12/3/92`.

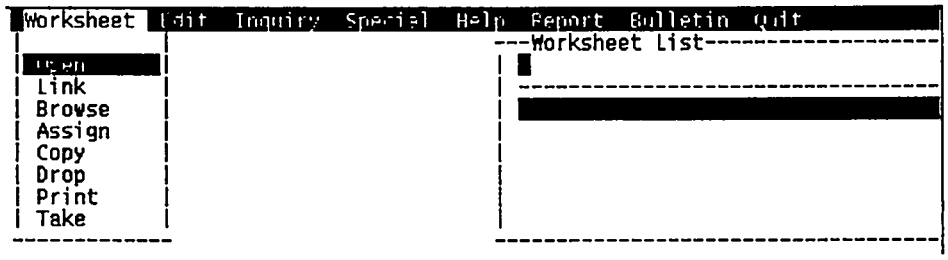

Figure 2-14 The empty Worksheet list.

- Enter the name of your Worksheet and press Return.

  A message appears, saying that there is not yet a Worksheet with that name. It asks if you want to create one.

- Press `y` (for yes) and then Return.

  A worksheet is opened. You may type in a Description if you desire. Use the Tab or arrow keys to move down to the entity field. It says, "<no link>". Since there is no link, you want to create a new one.

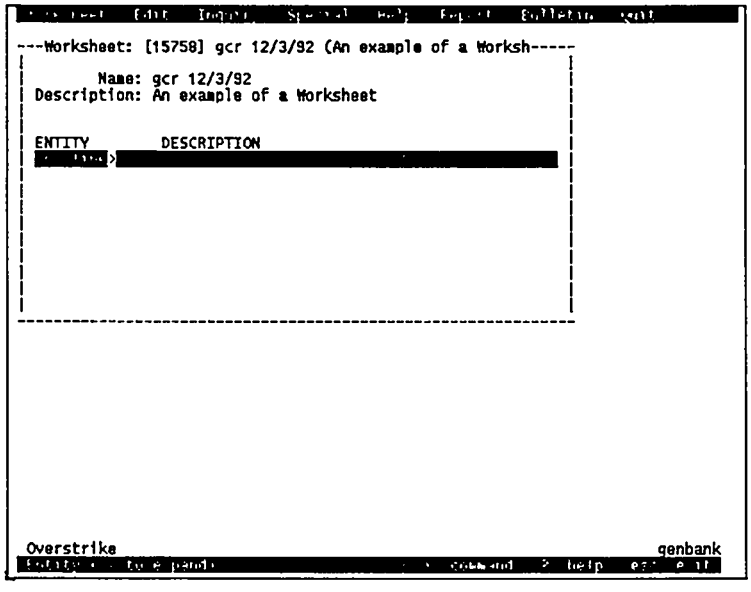

Figure 2-15 The Worksheet.

5. Link in a Person form to the Worksheet.

- Type `\eel` (Edit: Entity: Link).

  A list of entity types will be displayed. These represent the kind of entities that can be linked to a worksheet. To select an entity, type the first few letters of its name.

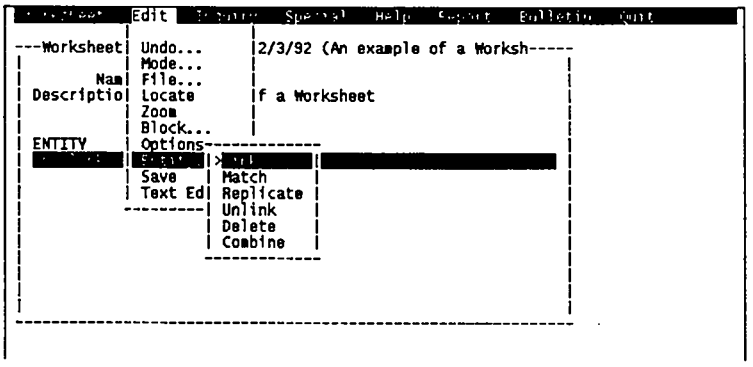

Figure 2-16 The Edit menu with the Entity submenu. The keystrokes \ee call up these menus.

- Type pe and then Return. (pe highlights Person and Return selects it.)

  A locator box appears. The box has five fields. All of them are text fields. AWB will take the information in this box and look in the database for a Person entity that matches the information in the box.

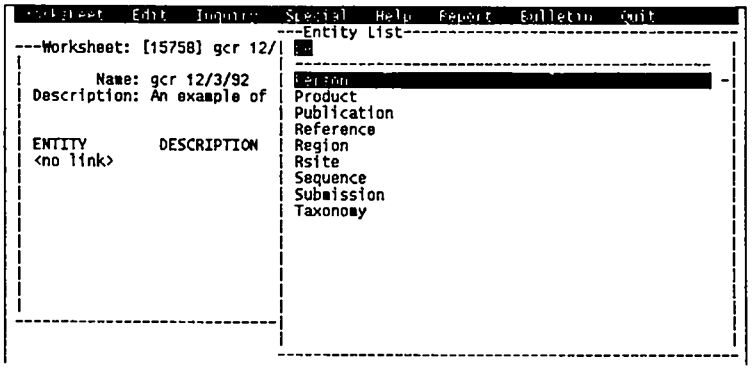

Figure 2-17 Selecting Person from the Entity list.

- Enter your last name with the first letter capitalized (e.g. Reese) and Return.
- Enter your first name with the first letter capitalized (e.g. George) and Return.

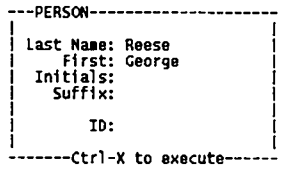

Figure 2-18 The Person locator.

- Enter an initial. Include a period after the initial.

- Enter a suffix, if appropriate.

  The suffix should be part of your last name (e.g., Jr. or III), not a title like Ph.d.

- Press ^X.

  AWB will present you with a list of people in the database with that name. If it can not find that person with these names, AWB will print the message, "No match found. Create new entry (y/n)?"

- If AWB does not find a match, type y.

  This will link in a new Person form that contains your name as you entered it.

- If AWB does find a match, then either select your name from the list or Esc back to the locator box.

  (There are two possibilities. Either your name is already in the database, or AWB found a name that matched the information you gave the locator box. If your name is already in the database, you can select your name from the list by Tabbing to highlight it and then pressing Return. If no name on the list is correct, Esc back to the locator box and enter more information.)

- With the Person linked in, ^X to expand the field to the Person form.

6. **Fill out the Person form.**

   - Enter as much information as you can into the type in fields.

     For complete field information see "The Person Form" on page B-24.

```
---Worksheet: [15758] gcr 12/3/92 (An example of a Worksh-----
| ---Person: [174956] Reese, George C. III-----------------------------
| |
| |   Last Name: Reese            First: George       Initials: C.    |
| |      Suffix: III                                                  |
| |                                                                   |
| | Sp. Address: as k710                                              |
| | Institution: <no link>                                            |
| |                                                                   |
| |  Work Phone: 665-3799                 Ext:                        |
| |  Home Phone:                                                      |
| |         FAX:                         Telex:                       |
| |      E Mail: gcr@life.lanl.gov                                    |
| |                                                                   |
| | User Number: 376          Perm:              Corr. Author:        |
| |                                                                   |
| |  Comment(s): Jan 12 1993 14:54 (182744)Reese, George C. III       |
-----------------------------------------------------------------------
```

Figure 2-19 The Person form. A link from the Worksheet was followed to this entity. Highlight the Institution field and type `\eel` to link an Institution Address entity to the Person entity.

7. **Link in the Institution Address form.**

   - In the Institution field, type `\eel`.

     The Institution (Address) locator appears.

   - Enter the institution name and ^X.

     Capitalize the first letters and use wildcards. For example, look for Los Alamos National Laboratory with *Los Alamos*. The wildcards help minimize spelling and capitalization differences that keep AWB from finding an entity.

     You want to be sure that the information is not already in the database before you create a new entity.

8. **Expand to the Address form (for the institution) and check the fields there.**

The information entered in these fields should only refer to an institution. That is, the address and phone numbers should be those of a department rather than those of an individual.

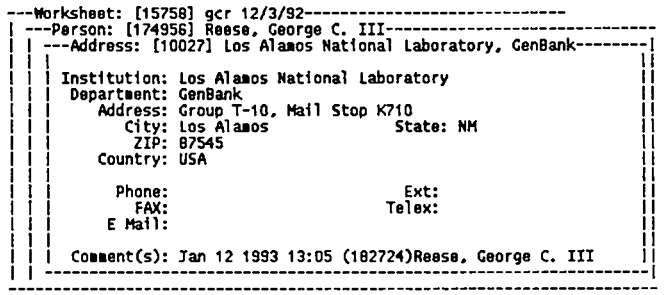

Figure 2-20 An Address entity linked to a Person entity, linked to a Worksheet.

9. **Quit the Address form and the Person form.**
   - \qf (Quit: Form) will quit the forms.

     Notice as you quit the forms that the status bar says, "Updating database, please wait..." As you exit, AWB automatically saves the information that you entered. \es (Edit: Save) can be used at any place in the form to force AWB to save the information (but not quit).
10. **At the Worksheet, try linking in a Gene, Product, or Publication.**
11. **Find the linking fields and expand them to see more forms.**

    Notice the changes in the status bar as you move through the fields. At a linking field the status bar will say, "*Fieldname* (^X to expand)."

    Be careful as you move through the fields not to change any information. (Use ^O to change the editing mode to Readonly.)
12. **Practice using wildcards in locator boxes to find entities.**
13. **When you are finished, quit the program.**
    - Use \qf to quit any forms.

      When you quit a worksheet, it will remain on the list, and can be reopened later. \wd (Worksheet Drop) is the command to remove a worksheet from the list.
    - Type \qp (Quit: Program) to quit the program.

By now you should be familiar with forms, fields, links, lists, and locator boxes. The next chapter contains information on handling GenBank submissions. Information on entering a sequence can be found in Chapter 4. Entering a sequence with AWB directly is in section 4.3.

2.3 A Typical Route Through the Forms

There are many ways to move through the database with AWB. Most entites may be displayed by the uer at any time. However, there is a general path through the forms that is commonly used by GenBank annotators. The path proceeds in the following manner:

1. **Start with a Paper.**

   Off-site users and dataflow staff create the Paper entities. Annotators reviewing submissions will display the Paper entity that has already been created.

2. **If this is a submission to GenBank, through E-mail or disk, check the Submission form.**

   GenBank staff only.

3. **Move to the Entry form.**

   One or more Entry entities are linked to the Paper. The Entry form has several text fields and exclusive settings. There is a single link to a sequence in each Entry, but there may be multiple links to Keywords and, possibly, Secondary Accession Numbers.

4. **When the Entry form is completed, follow or create the link to the Sequence entity.**

   The Sequence form has a scrolling text field for the Sequence and multiple link fields for the Features and Source(s).

5. **Follow/create the Source link to the Sequence.**

   There is usually only one Source for each Sequence, but there may be more.

6. **From the Source form, follow or create a link to the Taxonomy node of the source organism.**

   Off-site users can link in a Taxonomy entity, but can not edit any of the information displayed in the Taxonomy form.

7. **After the Taxonomy is linked, return to the Sequence form.**

   As you leave the Source form, check it for completeness.

8. **From the Sequence form, follow or create the links to the Features.**

   Features are the most complex aspect of the annotation. A Feature key should be linked to each Feature. Gene Occurrences, Products, and Qualifiers may also be linked. If the Feature is virtual, other component Features must be linked in.

9. **Back out of the forms.**

   Use the Exit: Save command at the bottom of the all forms. The changes will be saved as you quit the forms. If you do not want to save changes, use the Exit: Reset command.

This is only a general description of the motion through the forms. There are many specific tasks along the way that are described in Chapters 4 through 6.

You may wish to refer to the Figure below as you use those Chapters.

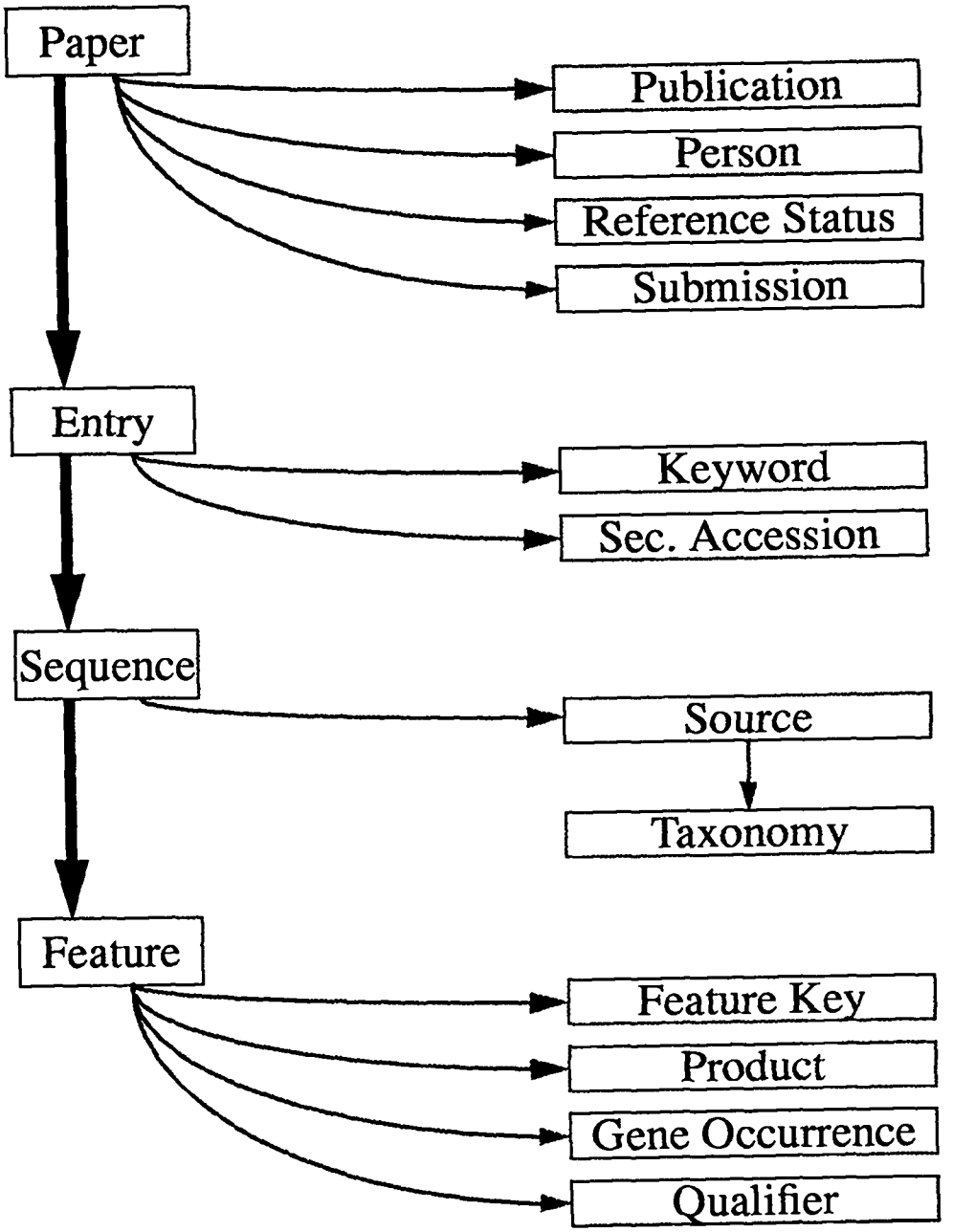

Figure 2-21 A typical route through the forms.

Chapter 3 Submission Processing

Submissions to GenBank are made by one of two means: an Authorin file or a GenBank Submission Form. Both types of submissions may arrive through either E-mail or the postal service.

This chapter contains directions for processing both types of submissions. The first section contains instructions for processing E-mail submissions. The second section contains instructions for reading submissions sent on disk.

Off-site users: This chapter is for GenBank staff who process submissions.

GenBank staff: This chapter contains instructions for dataflow procedures.

- Section 3.1 contains instructions for processing E-mail submissions.
- Section 3.2 contains instructions for disk reading.

3.1 Processing E-mail

Submissions from authors to GenBank are sent via electronic mail to gb-sub@genome.lanl.gov. These messages are sorted once a day and run through a program (Mailsplit) that prints out the submissions and sends acknowledgments to the submitters.

3.1.1 Initial E-mail Handling

The initial mail handling involves logging in as gb-sub and opening the mailtool. After the mail tool is opened, the messages are processed by eliminating all but the actual submissions. (These steps assume that you are in Open Windows.)

1. **To login as gb-sub from a shell tool follow these steps:**
   - At the Unix prompt type `/bin/su gb-sub`
   - Enter your password in the window that appears.

     You may need to click on the window to make it active.
2. **You may also login with the Workspace menu options.**
   - Right click and hold on the background to call up a Workspace menu.
   - Choose Change User from the menu and then gb-sub from the submenu.

     An x-term window appears.
   - Enter your password at the prompt in the window and Return.
3. **Open the mailtool and enlarge the window.**
   - In the x-term window type `mailtool`.

     The mailtool icon appears after a few seconds.
   - Double click on the icon to open the mailtool.
   - Double click on the header to enlarge the window.
4. **Group the messages by sender.**

   Grouping by sender will ensure that messages from the same sender are read together. This is useful if an author has made a submission in several parts and the components are scattered in the mail queue.
   - Right click and hold on the View menu button.

     The View menu appears.
   - Drag down to Sort By on the menu and choose Sender on the submenu.
5. **Display the first message.**
   - Left click on the message to select it.

     A rectangular box encloses it.
   - Double click on the message to view it.

     The View Message window appears. Use the scroll bar on the right to move through the message.

The next action is to carefully go through the messages and whittle down the mail queue to those messages that are actual submissions to GenBank. The mail will generally be in one of six categories: submissions to GenBank, copies of acknowledgments, citation updates, requests for information, or submissions with invalid format.

6. **Save submissions to GenBank.**

   Submissions will be either Authorin transactions or Submission forms. To see what these look like, see Authorin Submissions on page 4-6 and Submission Forms on page 4-10. If the message is one of these, leave it in the mail queue and go on.

7. **Remove acknowledgments.**

   These will be copies of responses sent by GenBank data flow staff to yesterday's submitters and the automatic "We're working on your mail" messages generated from running the mailsplit program the day before.

   - Click on the Delete button to remove the message.

8. **Forward update messages.**

   These are passed to the user named "update".

   - Go to the Compose menu in the menu bar of the mailtool.
   - Choose Forward from the Compose menu.

     A new Message window appears.
   - Type `update` on the To: line of the message window.
   - Click Deliver to send the message.

9. **If the request is for information about an accession number, respond immediately or print out the message and place it in the Inquiry bin.**

   - To respond immediately, click on the Reply button and enter your message in the Message window that appears.

     Click on Deliver to send the message.
   - To print out the message, choose Print from the File menu.

     Place the printed message in the Inquiry bin.
   - Delete the message from the mail queue.

10. **Respond to requests for information and messages with invalid format.**

    Requests will be handled differently depending on the type of information requested. A personal response may be necessary. Check the list of form letters in Response Templates on page 3-4 for an appropriate response.

    - Send a form letter by choosing Reply from the mail header.
    - Right click and hold on the Include menu in the Compose Message window and drag to Templates.
    - Select the appropriate letter from the Templates submenu.

      The contents of the letter will appear in the Message window.
    - Click on Deliver to send the message.
    - Delete the original message from the mail queue.

Response Templates

The name of the template appears in italics.

- *authorin.ain* responds to authors who have sent the ".AIN" file instead of the correct ".SBT" file.
- *authorin.info* is a message describing Authorin and telling where copies are available.
- *blank* responds to a message that arrived blank in the gb-sub mail queue. It explains that accession numbers cannot be assigned until the nucleotide sequence arrives.
- *get-sub-form* is a response explaining how to obtain a data submission form.
- *just-sequence* responds to authors who have sent only a sequence. It tells them how to get Authorin and contains a submission form.

3.1.2 Running the Mailsplit Program

After the mail folder has been cleared of messages that are not submissions, the Mailsplit program is run to copy the remaining messages to files and print them out. For more information about Mailsplit and its options see Mailsplit on page C-10.

1. **Save the contents of the mail file.**

   Once this step is taken, it is impossible to undelete a deleted message. Taking this step assumes that the user has gone through the mail queue with care.

   - Choose Save Changes from the File menu.

2. **Run Mailsplit.**

   - Type `mailsplit`.

     This command runs the basic version of Mailsplit. For information on the various options that can be used, see Mailsplit on page C-10. The program will ask if you want remove the mail file. Answer with a `y` or an `n`.

   - Edit and/or send each response as it appears on the screen.

     When the message appears, you will be prompted to edit it. Move through the message with the space bar. Typing `q` quits the message. You will be prompted to send it. Answer with a `y` or an `n`.

When the E-mail submissions have been printed out they will be sorted, along with the submissions read from disks, and given the appropriate coversheets. See Separating the Printouts on page 3-9.

3.2 Disk Reading

Submissions that have been mailed in via the postal service will contain disks. They may be IBM, Macintosh, or NEC disks. The disks should contain either an Authorin file or a submission form. The goal of the disk reader is to get the files onto the hard drive (c:) of the PC. Then transfer the files (FTP) to the computer "transposon", and run the "disk" script to print a copy and save it to ~gb-sub/DISKS.

Before you read the disks, get onto the PC and type `del *.out`.

This clears the drive of old files.

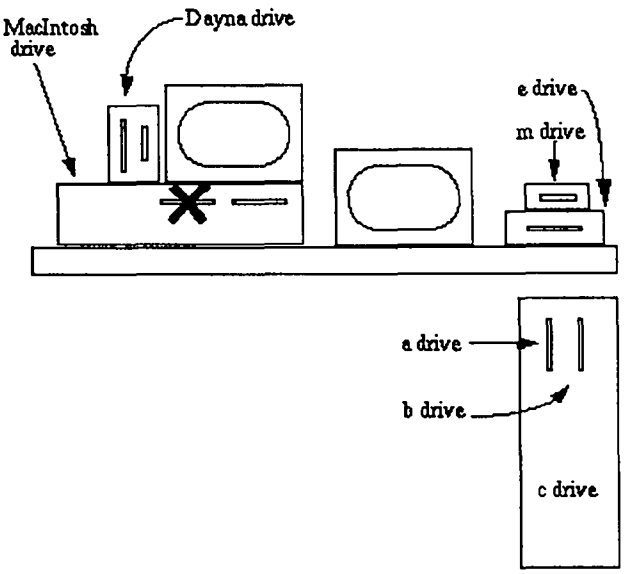

Figure 3-1 A sketch of the disk reading station.

1. **Copy the files from the IBM/PC 5 1/4" disks.**

   To copy the contents of these disks onto the c drive, follow these steps:

   - Find the 5 1/4" in disks that are IBM /PC disks.

     Insert these in the PC a drive, the one on the right.
   - Type `dir a:` to see what is there.

     A list of files should appear. Those followed by "sbt" are Authorin files.

     If the response is "sector not found error reading drive a" then go to the next numbered step. Otherwise continue here.
   - Type `type a: filename`.

     This will allow you to see what the file is like. You are looking for a sequence, an Authorin file, or a submission form.

- Type `reformln a: filename author's initials.out`.

  This step will perform the copy and rename the file.

  If there is more than one file for the same author, include a number with the author's initials, e.g., JCW1.out, JCW2.out, etc.
- Write down the original and the new file names.

  Write these on the coversheet or the letter sent with the submission
- If the disk does not "type" neatly, refer to the next two sets of steps.

  It is possible the contents of the disk are in word processor format; it could be WordPerfect or MS Word.

2. **Handle those in Word Perfect format**

   If the file is in word processor format, it may be WordPerfect:
   - Type `wp`.
   - Press Shift-F10 to include a file.
   - Enter `old filename`.
   - Press Control F5 to open a menu.
   - Press `1`, then `1` again to get the Word Perfect options.
   - Enter `author's initials.out`.
   - Press F7, then `n`, then `y`.
3. **Handle those in MS Word format**

   If the contents are MS Word, (less frequent) then follow these steps:
   - Type `word`.
   - Press `Esc` to reach the menu.
   - Select Transfer Load from the menu and select the author's old filename.
   - Press `Esc` and select Tranfer Save, then the new filename, then ↓ `tt` to select Text Only with Line Breaks.
4. **Handle the NEC 5 1/4" Disks**

   The goal here is to copy the files to a transfer disk by dragging with the mouse. Often the author will indicate that it is an NEC disk on the submission form or letter that arrives with the disk.
   - Put the NEC disk into the Dayna drive on the Mac.
   - Open the disk by double clicking on it.
   - Open the WP folder on the hard drive.

     Double click on the hard drive icon and then on the WP folder.
   - Double click on WordPerfect icon.
   - Select Open from the File menu.
   - Copy the file in the disk to the WP main folder.

     Do this by dragging the file.
   - Open the WP file "convert" to bring up the list of files you can convert.

     A window will open with options. One of the options is Save.
   - Save to the WP merge file.
   - Change the format to Text Only with Line Breaks.

This will be another option in the save window.

- Put the transfer disk into the Mac main drive and double click to open it.
- Copy the file by dragging to the transfer disk in the Mac main drive.
- Close the transfer disk and eject it.

  Close the disk by clicking in the small box in the upper left corner. When the disk is closed, eject it by dragging to the trash or by selecting Eject from the file menu.
- Read the transfer disk on the PC m drive

5. **Copy the files from the 3.5" disks using the m drive if possible.**

   These disks are usually Macintosh.

   - Put the disk in the PC m drive (the one with the apple on it).
   - Type `mdir`.

     The file will appear. (Those followed by .sbt are Authorin).

     If the message reads "Match Maker Error: Not a Macintosh diskette", then this may be a PC or high density disk. Try the e drive. If that fails, try it in the Dayna drive. If that fails, go to the Macintosh in George Bell's office. (See the sections below for the e drive and high density disks.)
   - Type `mtype m: filename`.

     This will allow you to see the file. You are looking for a sequence, an Authorin file, or a GenBank submission form.
   - Type `mcopy m: filename c: author's intials.out /t`.

     This will copy the file from the m drive to the c drive.

     If there is more than one file for the same author, include a number with the author's initials, e.g., JCW1.out, JCW2.out, etc.

     If the type only shows "~7" or "~4", it is a Mac application file and needs to be opened on the Mac. Go to step 7.

6. **Use the e drive to copy the files from some of the 3.5" disks.**

   - Insert the disks into the e drive.
   - Type `dir e:`.

     This will allow you to see what is on the disk. A list of files should appear. Those followed by "sbt" are Authorin files.

   If the response is "sector not found" or some other error, try the Dayna drive.

   - Type `type e: filename`.

     This will allow you to see what is in the file. You should be able to find the sequence.
   - Type `reformln e: filename author's initials.out`.

     If there is more than one file for the same author, include a number with the author's initials, e.g., JCW1.out, JCW2.out, etc.
   - Write down the original and the new file names.

     Write these on the submission form or letter that arrived with the disk.

7. **Read the Mac application files.**

   Mac application files should be read on the main Macintosh drive. These will be copied to a transfer disk and then read on the m drive.

   - Insert the disk in the main Macintosh drive.
   - Open the disk by double clicking on it.
   - Open the WP folder on the hard drive.

     Double click on the hard drive icon and then on the WP folder.
   - Double click on the WordPerfect icon.
   - Select "open" from the file menu.
   - Copy the file in the disk to the WP main folder.

     Do this by dragging the file.
   - Open the WP file "convert" to bring up the list of files you can convert.

     A window will open with options. One of the options is Save.
   - Save to the WP merge file.
   - Change the format to "text only with line breaks".

     This will be another option in the save window.
   - Put the transfer disk into the Mac main drive and double click to open it.
   - Copy the file by dragging to the transfer disk in the Mac main drive.
   - Close the transfer disk and eject it.

     Close the disk by clicking on the small box in the upper left corner. When the disk is closed, eject it by dragging to the trash or by selecting Eject from the File menu.
   - Read the transfer disk on the PC m drive.

8. **Transfer files from high density disks to a low density disk.**

   The high density disks must be transferred to a low density disk. If these can not be read on the Dayna drive, then use the Macintosh in George Bell's office.

   - Insert the disk into the drive.
   - Open the disk by double clicking on it with the mouse.
   - Open the hard drive by double clicking on it.
   - Open the GenBank Transfer folder.
   - Copy (by dragging) the file to the hard drive on the Macintosh
   - Eject the disk.
   - Once all the disks have been copied to the hard drive, insert the transfer disk.
   - Copy the files from the hard drive to the transfer disk.
   - Read the transfer disk on the PC m drive.

9. **Transfer the files to Transposon.**

   After all the files have been put on the c drive, transfer the files to transposon with the following steps:

   - Type `ftp -i transposon.`

     You will then be asked for a password. Enter your password
   - At the ftp prompt (ftp>) type `mput *.out`.

This step transfers all the .out files to transposon.

- After the files have been transferred, type `quit`.
  This takes you out of ftp and back to the c drive.
- At the prompt (c>) type `telnet transposon`.
- Type `ansi` for the terminal type.
- At the prompt (%) type `disk *.out`.
  This sends all the .out files to the printer and saves them to the ~gb-sub/DISKS directory.
- Type `^D` to log out of transposon.
  You will now be back in the c drive.
- Type `del *.out` to delete the files from the c drive.

3.2.1 Separating the Printouts

After the files have been printed out the next step is to separate them and assign the proper cover sheet. The printed files will be either Authorin forms or Submission forms. These must be further separated into HUP (Hold Until Published) and non-HUP.

Authorin Files.

```
entity ( address.1.10 ) {
institution_name = "University of Alabama at Birmingham";
dept = "Cell Biology";
street_address = "UAB Station";
city = "Birmingham";
state = "Alabama";
country = "USA";
zip_code = "35294";
```

Figure 3-2 Authorin files contain lines like these.

With Authorin files take the following steps:

1. **Separate the HUP files.**
   These files will have a hold date line under the reference listing. It looks like this: hold_date = "01-JUN-1992";.
2. **Attach a pink coversheet to the Authorin HUP submissions.**
3. **Put an "x" in the HUP box in the upper right hand corner of the coversheet.**
   If there is a hold date, write it in the HUP box. If there is not, write "Feb. 29, 1996."
4. **Place a yellow coversheet on the non-HUP files.**
5. **Initial and date the lines for "Received" and "Read" on all coversheets.**
6. **Place them in the Authorin tray**

Submission Forms

The GenBank Submission Form is a questionnaire which will also contain a sequence. These will also need to be separated into HUP and non-HUP.

Handle these with the following steps:

1. **Find the HUP forms.**

   Under the line where it says, "Do you agree that these data can be made available in the database before they appear in print?" there are two boxes: "yes" and "no". If it says "no", it is HUP. If it says "yes", or neither box is checked then it is not HUP.

2. **Attach a pink coversheet to the HUP submissions.**
3. **Put an "x" in the HUP box in the upper right hand corner of the coversheet.**

   If there is a hold date, write it in the HUP box. If there is not, write "Feb. 29, 1996."

4. **Attach a yellow cover sheet to the non-HUP submissions.**
5. **Initial and date the lines for "Received" and "Read."**
6. **Place the submissions in the Subform tray.**

Chapter 4 Sequence Entry

This chapter contains instructions for entering a sequence into the database. The sequence will either be entered into the database with AWB directly by an off-site user or it will be part of a submission to GenBank.

Section 4.1 contains the instructions for entering a sequence with AWB directly. The user creates the entities and enters the information with only AWB commands. Section 4.1 requires that the user have a basic familiarity with AWB. He/she must also have a file which contains the sequence to be read into the database.

Submissions to GenBank are of two sorts: Authorin or GenBank Submission Form. Sections 4.2 and 4.3 of this chapter contain the instructions for entering the sequences from these submissions. The instructions assume that the submissions have been sorted, copied to computer files, printed out, and assigned coversheets. (Chapter 3 contains the instructions for processing submissions.)

If the sequence is designated "Hold-Until-Published," it is held confidential until a certain date. When that date arrives (or if, for other reasons, the submission may be make public), the sequence is read into the database. Section 4.4 contains the instructions for updating HUP entries.

Off-site users: Follow the instructions in the first section of this chapter to enter a sequence. The information in Appendix B will be helpful as you move through the forms.

- Section 4.1 contains the instructions for entering a sequence.
- Section 4.5 contains instructions for transferring a file to the computer Linker and instructions for changing your AWB password.

GenBank staff: The last two sections contain dataflow procedures for entering submissions. The information in Appendix B will be helpful as you move through the forms.

- Section 4.2 contains instructions for entering Authorin submissions with the Auinsub program.
- Section 4.3 contains instructions for entering data from a GenBank Submission Form.
- Section 4.4 contains instructions for updating HUP submissions.

4.1 Entering Information with AWB Directly

This section contains instructions for entering a sequence and reference information using AWB directly (i.e., not from a submission). It is written with an off-site user of AWB in mind.

This section assumes some familiarity with AWB. For an introduction to AWB see Chapter 2. You should have a login name and password. You should also have a file that contains a sequence (and only a sequence) to be read into the database. Multiple sequences should be placed in separate files.

For information on transferring a file from your home space to the computer at GenBank see "Transferring a Sequence File to Linker" on page 4-16.

1. **Login to AWB.**

   If you do not know how to login to AWB, see Chapter 2 for an introduction to AWB.

2. **Open a Worksheet.**

   - At the UNIX prompt, type `awb`.
   - Type `\wo` to bring up a list of Worksheets.

     You may choose from the list or enter the name of a new worksheet.
   - Press Return to select the chosen worksheet.

3. **Link in a Paper entity to the Worksheet.**

   A Paper entity is the same as a Reference entity, except that Entry entities can only be linked to Papers.

   - `\eel` in the ENTITY field of the worksheet to bring up the entity list.
   - Choose Paper from the list.

     You may Tab to it or type `pa`.
   - When the locator box appears, enter information to find a specific paper or `^X` on a blank locator box to link in a new Paper entity.

     If you `^X` on a blank locator, AWB will ask if you want to create a new entity. Type `y` for yes of `n` for no. When a new Paper is created, the "GEN citation created" status is set automatically.

   Note: More than one Entry may be linked to a single Paper. You should only create a new Paper entity if it does not already exist. Avoid making copies of the same Paper (Reference).

4. **In the Paper form, link in the Publication (journal name).**

   Skip this step if the sequence is not yet published or if, for any other reason, there is no journal involved.

   - `\eel` in the Publication field to call up the Publication locator.
   - Enter the abbreviation and `^X`.

     It is best to use wild cards if you do not know the GenBank journal abbreviation. For example, if you are looking The Journal of Molecular Biology, `J*Bio*` will call up a list that contains the journal.

- Select the correct item from the list found by the Publication locator.

5. **Enter the Paper's citation information.**
   - Type in the Volume, Issue, and Pages.
   - Type in the Title.

     Capitalize only the first word and special terms that require capitalization (e.g., DNA, Thermus aquaticus, fliD). For more information see "Titles" on page A-6.
6. **Link in the authors.**
   - Highlight the Author(s) field.
   - `\eel` calls up the Person Locator.
   - Type the author's name into the fields of the locator.

     Use full first and last names. Capitalize the first letter of the names. Include a period with the middle initial.
   - `^X` to search for the name.

     Before you create a new entity, be sure that the person is not already in the database.
   - Choose the author's name from the list. If no list appears, type `y` to create a new Person entity.
   - `^X` to follow the link the Person entity of the author.
7. **Fill in the Person form for each author.**
   - Type in the last and first names, middle initial, and the suffix (Jr., III).
   - Type in the specific address.

     This should be a building number, mailbox, mailstop, or other specific address.
   - In the Institution field, type `\eel`.

     The Institution (Address) locator appears.
   - Enter the institution name and `^X`.

     You may use wildcards in the Institution field. For example, `*U*Fl*` for the University of Florida.
   - Select the correct institution from the list and Return.
   - Type the person's work phone number, extension, home phone, FAX number, and telex into the appropriate fields.
   - Type a single, complete E-mail address into the E Mail field.
   - Toggle (`^T`) the Corr. Author field to yes if this is the person to whom correspondence about the entry should be sent.
   - Quit (`\qf`) back to the Paper form
   - Repeat these action steps for each author.
8. **Set the Pub Status field to the correct status.**
   - Highlight the Pub Status field.
   - `^T` to toggle the field.

     `\eo` will let you see a list of the toggle options. You may then choose from the list.

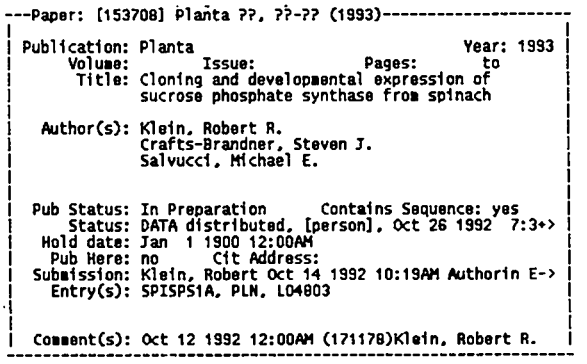

Figure 4-3 A completed Paper form.

9. **Skip the Hold date and Submissions field.**
   - For information about Submissions see section 4.2 on page 4-6 and section 4.3 on page 4-10.
10. **Link in the Entry entities.**
    - Type `\sl` (Special: Link new entries).

      AWB will ask, "how many?"
    - Enter the number (1 or more).

      This will create an Entry form and a Sequence form for each sequence. The accession number will be the last number in the field summary.
11. **Read in the sequence(s).**
    - In the Paper form, highlight the Entry field.
    - Follow the link to the Entry entity (`^X`).
    - In the Entry form, highlight the Sequence field.

      In should say, "0 bp."
    - Follow the link to the Sequence entity.
    - Highlight the Sequence field in the Sequence form.
    - `^X` to expand to the Sequence editor.

      The blank Sequence editor appears.
    - Type `\sr` (Special: Read sequence).

      A box will appear which says, "Input file name." Enter the name of the file which contains the sequence. When you press Return, the sequence should appear in the Sequence editor. Figure 4-4 illustrates this situation.

    Note: If the file with the sequence is not in the current directory, you will need to type the path as well as the filename.

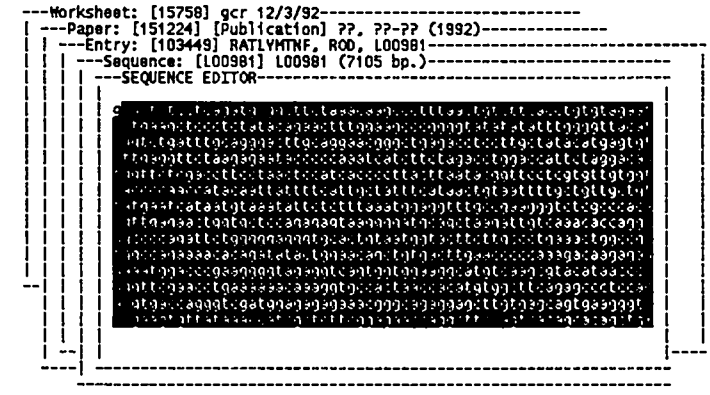

Figure 4-4 The Sequence editor. The Sequence has been read into the text field.

12. **Quit (\qf) the forms until you reach the Paper form.**
13. **If there are multiple sequences, repeat step 11. Go to the next step when all the sequences are entered.**
14. **When you are ready to annotate the sequence, go to Chapter 5.**
    - To quit the forms type \qf. AWB will update the database as it leaves the forms.
    - When you have quit the Worksheet, \qp will quit the program. The worksheet opened this time will be on the list of worksheets the next time you use the \wo command. To remove a worksheet, use \wd (Worksheet: Drop).

The Entry has now been made, and the sequence has been entered into the database. For information on annotating a sequence see Chapter 5.

4.2 Authorin Submissions

Authorin submissions can be read into the database automatically with the "auinsub" program. The program reads information from a file containing an Authorin submission. It finds the sequences, assigns an accession number to each, adds the accession numbers to the Authorin file, and stores the modified submission in a file called *filename*.rewrite.

You must then use AWB to check the information.

This section assumes that the instructions in Chapter 3 have been carried out and that the user has a copy of the Authorin submission (with a coversheet) in hand. Yellow coversheets indicate submissions for which the data is public. Pink coversheets indicate submissions for which the data is confidential.

```
dept = "Biochemistry & Molecular Biology";
street_address = "610 Oak Ridge Road";
city = "Fairview";
country = "USA";
state = "New Mexico";
zip_code = "87845";
}

}

}
} /* end entry */
entity(sequence.1.10) {
sequence = "
gtcggccgcttccaccccagttttgaagaaaacaggtctgaaacaaggtcttactcccca
gctgcctctcaacacagtgactaccagctctccagataccaagtccagctttgtccgcca
acctgtctgacatgtcgggacccgtgccaagcagagccagagtttacacagatgttaata
cacacagaccccgagagtactgggattacgagtcacatgtggtggaatggggaaatcaag
atgactaccagctggttagaaaaattaggtcggggtaaatacagtgaagtatttgaagcca
tcaacatcacaaataatgaaaaagttgttgttaaaattctcaagccagtaaagaagaaga
aaatcaagcgtgaaataaagattttggagaatttgcgaggcggtcccaacatcatcacac
tggcagacattgtaaaagaccctgtgtcacgaactcccgccttggtttttgaacacgtaa
acaacacagacttcaagcaattgtaccagacgttaacagactatgatattcgatttaca
tgtatgagattctaaaggcccttgattactgccatagcatggggattatgcacagagatg
tcaagccccataatgtcatgattgatcatgagcacagaaagctacggctaatagactggg
gtttggctgagttttaccatcctggccaagaatataatgtccgagttgcttcccgatatt
tcaaaggtcctgagctacttgtagactatcagatgtacgattatagtttggatatgtgga
```

Figure 4-1 Part of an Authorin file. The information in a file like this is entered with the Auinsub program.

Use the following steps and run Auinsub on each file.

1. **Copy the file to your home directory.**
   - In a Shell Tool type `getsub` *`filename`*.

     The program should respond with "*filename* successfully copied."
2. **Check to see if the submission is public or HUP.**

   The color of the coversheet (pink for HUP, yellow for public) reveals the nature of the submission, but it should be checked.
   - To check if the submission is HUP, type `grep` *`hold_date filename`* and Return.

     The grep command searches the file for the string "hold_date" and returns the line on which that string occurs. If the submission is HUP, the hold date will appear on the next line. A hold date looks like this:
     hold_date = "01-SEP-1993."

     The Ref program is also useful for displaying publication information. See "Ref" on page C-13.
3. **Check to be sure the journal abbreviation is in the database.**

   The Pubabbrev program checks character strings in the file for journal names and changes recognizable names to the official abbreviation.
   - Type `pubabbrev` *`filename`*.
   - Enter your password.

     If the program works, it will respond with the journal name and the abbreviation.

     If there is a problem, you should go into the file with vi (the visual text editor) and fix it. If you do not know what needs to be done, get help from the dataflow coordinator.

4. **Check the other information in the Authorin form.**
   - Type `vi filename`.

     The file will appear in the window. You are now in vi, a text editor. Use the arrow keys to move through the text.
   - If the submission is HUP, check to be sure there is a hold date.

     In vi type `/hold_date` and Return, and vi will take you to that string in the file.
   - Check the title of the paper to be sure that it fits the conventions.

     See "Titles" on page A-6 for the conventions.
   - Remove titles like Ph.D from author names.
   - If an author has included a middle name, change it to a middle initial (with a period).
5. **Run the Auinsub program.**
   - If the Submission is to be public, type `auinsub filename pub`.
   - If the Submission is HUP, type `auinsub filename hup`.

     If the program succeeds, it will say "transaction processed successfully" and report the rf_id and the accession number. For submissions with more than one sequence, write the accession number of each sequence on the portion of the printout that pertains to it.

     If the hold date in the Authorin form is later than the current date, the program will ask if you want to make it HUP, to read it in as public, or to quit. You will need to answer appropriately.

     If the program fails, it will say "syntax error" or "transaction failed". If this happens, get help and save the .rewrite file. If you suspect something is wrong with the program, send a bug report to gb-software.
6. **Complete the coversheet.**
   - Write down the rf_id, the accession number, and the citation information of the coversheet.
   - If the Submission is HUP, write the hold date in the HUP box on the coversheet.

Take the following steps in AWB to make sure the correct information has entered the database, set the appropriate statuses, and make acknowledgments.

7. **Login to AWB and open a Worksheet.**

   If you do not know how to login or open a Worksheet, see Chapter 2 for introductory information about AWB.
8. **Link in the Paper using the rf_id supplied by auinsub.**
   - Type `\eel` in the ENTITY field of the worksheet.
   - In the PAPER LOCATOR's Database ID field enter the rf_id supplied by auinsub.
   - `^X` to call up the list which should contain only one paper.
   - Select that paper.
   - `^X` to expand to the Paper form.

9. **Check the information in the Paper form and fill out the coversheet.**
   - ☐ Check to make sure the journal, page numbers, and volume are correct.
   - ☐ Write the journal information on the coversheet.
   - ☐ Check the title, making sure the capitalization is correct.

     For information on title conventions see "Titles" on page A-6.
   - ☐ Check the list of authors for correct capitalization and punctuation.
   - ☐ Write the first author's name on the coversheet.
   - ☐ Make sure the Publication Status is correct.

     Is it Published, Unpublished, In Press, etc.?
   - ☐ Check off the "Logged" box on the coversheet. Initial and date it.
10. **Check the Submission information.**
   - ☐ In the Paper form, arrow to the Submission field and expand.

     AWB follows the link to the Submission entity.
   - ☐ Check the author information (by following the link to the Person entity).
   - ☐ In the Person form, check the Address and the E-mail address.

     Double check the E-mail address against the return address in the file.
   - ☐ Follow the link in the Institution field and check the Institution Address entity for a complete postal address.

     Be sure to check the Phone Number and the Fax Number.
   - ☐ Return to the Submission entity by typing `\qf` twice.
   - ☐ In the Submission form, toggle (`^T`) the Medium field to the correct value.
   - ☐ Check for text in Text field.

     This should be the text of an Authorin submission. `\et` (Edit: Text editor) will put you in a vi file of the submission text. `:q` will quit the editor.
   - ☐ If the text of the submission is not there, type `\efr` (Edit File Read).

     A box appears that asks for the name of the input file. Enter *`filename`*`.rewrite`.
   - ☐ Quit the Submission form.
11. **Check the Entry entity to see that the correct sequence information was entered.**
   - ☐ In the Paper form, highlight the Entry field and expand to the Entry form.
   - ☐ In the Entry form, check the Sequence field.

     If the Submission is HUP, the Sequence field should say, "0 bp." Otherwise, there will be some non-zero number in this field.
   - ☐ Quit the Entry form.
12. **Send an acknowledgment to the corresponding author.**
   - ☐ In the Paper form, type `\sae` (Special: Acknowledge: Email) to acknowledge with E-mail or `\sap` to acknowledge with a printed letter to be faxed.
13. **Complete the coversheet and file the submission.**
   - ☐ If the Submission is HUP, quit to the Worksheetand file the form by accession number in the HUP drawer.

- Unlink the Paper from the Worksheet with `\eeu` (Edit: Entity: Unlink).
- If the submission is public, type `\ss` (Special: Set status).

  The list of Statuses appears. Assign the status "AN paper received in queue."
- If the Submission is public, put the Authorin form, with coversheet, in the Annotation Queue bin.

14. Repeat these steps for each Authorin submission.

The paper is now ready to be reviewed. See "Review" on page 6-1.

4.3 Submission Forms

The GenBank Submission Form is a questionnaire that is filled out by the scientist making the submission and sent to GenBank by E-mail or on a floppy disk. The submissions are sorted and copied to computer files, and copies of the submissions are printed out (see Chapter 3).

This section assumes that the instructions in Chapter 3 have been carried out, and the user has a copy of the submission (with a coversheet) in hand. Yellow coversheets indicate submissions for which the data is public. Pink coversheets indicate submissions for which the data is confidential.

```
I.  GENERAL INFORMATION
-------------------------------------------------------------------------------
Your last name  Public          first name John     middle inititials  Q
-------------------------------------------------------------------------------
Institution     Bradford University
-------------------------------------------------------------------------------
Address         P.O. Box 1999
                Bradford University
                Bradford, Minnesota
-------------------------------------------------------------------------------
Computer mail address jqp@mars.bdu.edu              Telex number
-------------------------------------------------------------------------------
Telephone  287 825-8291                          Telefax number  287 827-1032
   Direct
-------------------------------------------------------------------------------
On what medium and in what format are you sending us your sequence data?
(see instructions at the beginning of this form)
  [X] electronic mail
  [ ] diskette
        computer:              operating system:          editor:
  [ ] magnetic tape
        record length:              blocksize:            label type:
        density          [ ] 800    [ ] 1600    [ ] 6250
        character code   [ ] ASCII  [ ] EBCDIC
-------------------------------------------------------------------------------
```

Figure 4-2 Part of a GenBank Submission Form. Information in this questionnaire is entered piece by piece from a vi file.

With the submission forms, AWB will be run side-by-side with a vi file of the submission. Typically most of the information can be cut and pasted into the AWB forms. To get the vi file follow these steps.

1. Copy the file to your home directory.

- In a Shell Tool type `getsub` *`filename`*.

  The program should respond with "*filename* successfully copied."

2. **Check the information in the Submission form.**
   - Type `vi filename`.

     The file will appear in the window. You are now in vi, a text editor. Use the arrow keys to move through the text.
   - Check to see if the submission is HUP.

     This is a double check to make sure the color of the coversheet is correct. If the submission is HUP, it will have a line like this:

```
-------------------------------------------------------------------------
Do you agree that these  data can be made  available in the  database
before they appear in print?
  [ ] yes  [X] no, they should be made available only after publication.
=========================================================================
```

   - Check the title of the paper to be sure that it fits the conventions.

     See "Titles" on page A-6 for the conventions.
   - Remove titles like Ph.D from author names.
   - If an author has included a middle name, change it to a middle initial (with a period).
   - Type `:q` to quit the file. (`:wq` will write any changes to the file and then quit.)
3. **Copy the sequence(s) to a new file.**

   Give the sequence the same file name, but add ".seq" to the end. For several sequences make them .seq1, .seq2, etc. For instructions on copying the sequence to a file, see "Copying a Sequence to a File" on page 4-18.
4. **Login to AWB and open a Worksheet.**

   If you do not know how to login or open a Worksheet, see Chapter 2 for introductory information about AWB.
5. **Create a new Paper entity.**
   - Type `\eel` in the ENTITY field of the worksheet.
   - Type `pa` to select a paper from the Entity List.
   - When the PAPER LOCATOR appears, type `^X` and the `y` to call up a blank form.
   - Write the rf_id of the new entity on the top of the coversheet.
6. **In the Paper form, link in the Publication.**
   - With the Publication field highlighted, type `\eel`.

     The Publication locator appears.
   - Enter the journal abbreviation in the Publication locator and `^X`.

     A list appears.
   - Select the correct abbreviation from the list and Return.
7. **Fill in the Title and other article information fields.**
   - Fill in the Title field and the Volume, Pages, and Issue if known.

     For conventions on titles, see "Titles" on page A-6.
   - Toggle (`^T`) to the correct Pub Status.

     That is, "In Prep," "Published," etc.

8. **Link the Authors to the Paper.**
   - ☐ Highlight the Author(s) field and type `\eel`.

     A Person locator box appears. You must enter the name of an author, `^X`, and then select the Person from the list or add them to the database (if AWB cannot find the name). Repeat this step for each author.
9. **If you have created a new Author entity, fill in its Person form.**
   - ☐ `^X` on the Author's name in the Paper form.

     The link is followed to the Person entity.
   - ☐ Fill in the fields of the Person form.

     For information about the fields of the Person form, see "The Person Form" on page B-24.
   - ☐ Link in the Institution Address, or create a new Address entity if necessary.

     You may use wildcards in the locator box. For example, type `U* Fl*` for the University of Florida. Be sure the name is not already in the database before you create a new one.
   - ☐ If you create a new Address entity, you must follow the link to that entity and complete the Address form.
10. **Link in a new Submission entity.**
    - ☐ Quit back to the Paper form.
    - ☐ Highlight the Submission field and type `\eel`.

      `^X` on the blank locator box to bring up the new form.
11. **In the Submission form, link in the submitting author.**
    - ☐ Highlight the Author field, type `\eel`.
    - ☐ Enter the name of the submitting author in the locator box and `^X`.
    - ☐ Select the Author's name from the list that appears.
    - ☐ `^X` to expand to the Person form of the author.
12. **Complete the Person form of the submitting author.**
    - ☐ In the Person form, type the specific address in the Sp. Address field.

      A post office box, mail stop or building number are examples of specific addresses.
    - ☐ Type in the E-mail address and the phone numbers.
    - ☐ Toggle the Corresponding Author field.
    - ☐ Save the changes and quit the Person form.
13. **Complete the rest of the Submission form.**
    - ☐ Toggle (`^T`) the Medium field to the correct value.
    - ☐ In the Text field, type `\efr` (Edit: File: Read) and enter the name of the file which contains the complete text of the submission.

      Do not enter the file which contains only the sequence.
    - ☐ Save the changes and quit the Submission form.
14. **In the Paper Form, link in the Entries.**
    - ☐ If you are entering only one sequence, type `\eel` and `^X` on the blank locator box.

The blank Entry form will be linked in.

- If there is more than one sequence, link them all into the Entry field with `\sl` (Special: Link new entries).

  A box will appear asking, "how many?"

- Enter the number of sequences.

  This will create Entry entities for each of the Sequences. The last number in the field summary for each of the entries is the accession number. Write the accession numbers on the coversheet.

15. If the Submission is HUP, set the Status.

For the full list of statuses see "The Reference Status Form" on page B-29.

- Highlight the Status field and type `\ss`.

  The list of Statuses appears.

- Select the status "SUB hold until published."

16. If the submission is <u>not</u> HUP, enter the Sequence(s).

- `^X` on an Entry summary.
- In the Entry form, highlight the Sequence field.
- Expand to the Sequence form (`^X`).

  The link is followed to the Sequence entity. The Sequence field in the Sequence form should say, "0 bp."

- Expand on the Sequence field to the Sequence Editor.
- If the sequence is not HUP, enter it with the command `\srs` (Special: Read: Sequence).

  AWB will ask for the name of the file which contains the sequence. The file should contain only the sequence. For instructions on creating a sequence file see "Copying a Sequence to a File" on page 4-18.

  DO NOT TAKE THIS STEP IF THE SUBMISSION IS HUP!

- Save the changes and quit to the Paper form.
- Repeat these substeps for each Entry.

17. In the Paper form, send an acknowledgment to the author.

- Type `\sae` (Special: Acknowledge: Email) to acknowledge with email or `\sap` to acknowledge with a letter.

18. Set two more Statuses on the non-HUP Papers.

Skip this step if the Paper is HUP.

- If the submission is not HUP, type `\ss` (Special: Set status) and select the status "SEQUENCE read into database."
- If the submission is not HUP, type `\ss` and select another status, "AN paper queued for annotation."

19. **Save the changes and quit the Paper entity.**
20. **File the HUP printout with coversheet in the HUP drawer.**
21. **If the Submission is public (non-HUP), put the printout of the GenBank Submission form, with coversheet, in the Annotation Queue bin.**
22. **Repeat these steps for each GenBank Submission form.**

When you have worked through all the submissions, you will need to dispose of the Worksheets.

23. **Quit AWB.**
    - \qp will quit the program.

4.4 HUP Updates

When submissions are given a hold-until-published status, only reference information is put in the database. The rest of the information is entered into the database when one of three events occurs: the release data arrives, the paper in the submission is published, or the author notifies GenBank to release the data. The steps for updating will depend on whether the submission was Authorin or a submission form.

4.4.1 Authorin HUP Updates

1. **Open a worksheet on the AWB and link in the original paper.**
2. **Follow the link to the Paper entity.**
3. **Follow the link to the Submission entity.**
   - Highlight the Submission field.
   - `^x` to follow the link.
4. **Go to the Submission Text field**
   - Type `\efw` (Edit: File: Write).

     After you type this command AWB will ask for the name of the output file. Choose any name that will be easy to remember.
5. **Go to a command window and run auin sub on the new file.**
   - Type `auinsub` *`filename`* `pub`.

     If the program succeeds, it will say, "transaction processed successfully" and report the rf_id and the accession number. The accession number will be the same. The rf_id, however, may be different.
6. **If the rf_id is different from the original, continue with the next step. If the rf_ids match, go to step 9.**
7. **Quit the submission and paper forms.**
8. **Combine the rf_ids.**
   - From the worksheet form type `\eec` (Edit: Entity: Combine).

     The entity list will appear.
   - Highlight "paper" and return.
   - Enter the two rf_id's.

     AWB will come back with both rf_id's and ask if you want to keep the newest one. You will respond with `y`.
9. **Assign a status.**
   - Expand to the Paper form.
   - Type `\ss` call up the list of statuses.
   - Select the status "AN paper received in queue."
10. **Quit the program and put the paper in the Annotation Queue bin.**

4.4.2 Submission Form HUP Updates

1. **Open a worksheet on the AWB and link in the original Paper.**

2. **Copy the sequence to a file.**
   - Expand on the Paper form.
   - In the Paper form, highlight the Submission field and expand.
   - In the Submission form, highlight the text field.
   - Type \et (Edit: Text Editor).

     This will place you in the text editor.
   - Arrow to the sequence, mark it, and copy it to a file.

     If you do not know how to copy to a vi file see "Copying a Sequence to a File" on page 4-18.

     Do not use the \efw command! This would edit out carriage returns and make the file one long line.
   - Quit the text editor (`:q`).
   - Quit the Submission form.
3. **Read the sequence into the Sequence editor.**
   - In the Paper form, highlight the Entry field.
   - Expand to the Entry form.
   - Highlight the Sequence field and expand to the Sequence form.

     The Sequence field will say "0 bp."
   - In the Sequence form, highlight the Sequence field and expand to the Sequence editor.
   - Type `\sr` (Special: Read sequence).

     A box will appear that asks for the input file. You will enter the name of the file that contains the sequence.
4. **Set a status.**
   - Quit the forms until you are in the Paper form.
   - Type `\ss` (Special: Set status).
   - Select "AN paper queued for annotation."
5. **Quit the worksheet and put the paper with coversheet in the Annotation Queue bin.**

4.5 Special Situations

4.5.1 Transferring a Sequence File to Linker

Off-site users will login to the computer named linker. The following steps are for transferring a file from the user's home space to linker at Los Alamos.

1. **Begin ftp (File Transfer Program).**
   - Type `ftp linker.lanl.gov`.

     The computer should return "Remote User Name".
   - Enter your system login name and Return.

The computer will say "Enter Password."

- Type in your password and Return.

  The ftp> prompt appears.

2. **Move the file to linker.**
   - At the ftp prompt type `put` *`filename`*.

     This puts the file into linker's space.
   - Type `quit` to leave ftp.

4.5.2 Changing an AWB Password

1. **From a Shell tool or Command window enter SQL.**
   - At the Unix prompt type isql.
     You are then asked for your password.
   - Enter you current AWB password.
     You will now see a numbered prompt (1>).
2. **Change your password.**
   - At the prompt type `sp_password` *`oldpassword, newpassword`* and Return.
     Put a comma and a space after the old password.
   - At the prompt type `go` and Return.
     The password is changed.
   - Type `quit` and Return to leave sql.

4.5.3 Copying a Sequence to a File

Follow these steps to write a sequence to a vi file. The sequence is part of the text of an Authorin submission or a Submission Form. The text of both types of submission can be found in the Text field of the Submission form. Follow these steps:

1. **Open AWB.**
2. **Expand to the Submission form.**
3. **In the Text field type `\et` (Edit: Text editor).**
   You are now in a vi file which contains the text of the submission.
4. **Arrow to the first letter of the sequence.**
5. **Type `ma`.**
   This "marks" a point "a."
6. **Go to the last letter of the sequence and type `mb`.**
   This "marks" the point "b."
7. **Type `:'a,'b w` *`newfilename`*.**
   This command writes the text between "a" and "b" to a new file. You may choose any name for the newfilename that is easy to remember.
8. **Quit the editor with the command `:q`.**

Chapter 5 Annotation

This chapter contains the steps for annotating a typical sequence. Additional information about virtual features is included in "Special Situations" on page 5-8.

The instructions in section 5.1 assume that the sequence and reference information have already been entered into the database. For information on entering a sequence, see Chapter 4.

Off-site users: Use the instructions in 5.1 to annotate sequences. The information in Appendix B will be helpful as you move through the forms.

- Section 5.1 contains instructions for adding annotation information to a sequence.

GenBank staff: Read all of this chapter. The information in Appendix B will be helpful as you move through the forms.

- Section 5.1 contains instructions for adding annotation information to a sequence.
- Section 5.2 describes the conventions for annotating virtual features.

5.1 Annotating a Typical Sequence

1. **Login to AWB and open a Worksheet.**

   If you do not know how to login or open a worksheet, see Chapter 2 for introductory information about AWB.

2. **Link in the Paper.**

   - In the Entity field of the worksheet type `\eel`.
   - Select Paper from the Entity List.
   - Enter the rf_id (Reference ID) or as much other information as possible and `^X`.

     The Reference ID is the quickest way to bring up the paper.
   - Select the correct paper from the list.
   - When the paper appears in the ENTITY field, `^X` to expand to the Paper form.

3. **Set the Statuses.**

   For a description of the fields in the Paper form see "Special Commands" on page B-22.

   - Type `\ss` (Special: Set status).

     The list of statuses appears.
   - Select "AN annotation begun"
   - Toggle the Pub Status field to the correct value.

     `\eo` will display the list of options.

4. **If you are annotating a submission, follow the link to the Submission entity and check the information there.**

   Off-site users skip this step.

   - From the Paper form, highlight the Submission field and type `^X`.
   - In the Submission form, toggle the medium to the correct value (`^T`).
   - Check to be sure the text of the Submission is in the Text field.

     You may Tab through the text of the Submission, or `\et` places you in the text editor.

     If the Submission is not there, `use \efr` (Edit: File: Read) to read in the text of the Submission. A box will appear which asks for the name of the input file. Enter the name of the file that contains the text of the Submission.
   - `\qf` to return to the Paper entity.

5. **Check to be sure all the Entry entities are linked.**

   It is possible that an Entry was accidentally unlinked. If this has happened, it can be found by using the accession number of the Sequencer in the Entry locator to find it, and then relinking it to the Paper.

   - In the Paper form, highlight the Entry field.
   - Tab through the Entries.

     If the list is longer that the display area an plus (+) or minus (-) sign will indicate that there are more entities to see. Use Tab to move forward and BackSpace to move backward through the list.

- If an Entry is missing, type `\eel`.

  An Entry locator appears.
- Type the accession number into the Accession # field and `^X`.
- Highlight the Entry on the list that appears and Return.

  The Entry is linked.

6. **Follow the link to the Entry entity.**

   If the Entry field of the Paper form has no link, then an Entry entity has not been linked to the Paper. See "Entering Information with AWB Directly" on page 4-3.

   - In the Paper form, highlight the Entry field and `^X`.

     The Entry form appears.

7. **Fill in the text fields of the Entry form.**

   For information on the fields of the Entry form, see "Special Commands" on page B-10.

   Skip the Locus and Division fields. You will return to them after you have linked in the Source and Taxonomy forms.

   - The Origin, Organism, and Old Source fields are seldom used.

     See "Special Commands" on page B-10 for more information about these fields.
   - Set the Mol Type and Topology fields to the correct values.

     `\eo` will display the list of options. The molecule type is usually ds-DNA (double stranded DNA) or ss-RNA (single stranded RNA). The topology can be either Linear or Circular. Circular means a complete, circular genome.

8. **Link in the keywords.**

   These are usually products or gene names (only in the case of bacteria and yeasts). They should be words that give information to help locate this entry. See "Keywords" on page A-3.

   - Highlight the Keyword field and type `\eel`.

     The Keyword locator appears.
   - Enter the keyword and `^X`.

     The keyword appears in the list. Use wildcards if there is no match on the first try, especially in place of any punctuation or capitalization.

9. **Link in any secondary accession numbers.**

   These are not frequent.

10. **Link in the primary reference.**

    For GenBank annotators working on a GenBank Submission Form, the primary Reference has usually been linked in by the dataflow staff.

    - Highlight the Reference field and type `\eel`.

      The Reference locator appears.
    - Enter the rf_id of the Paper and `^X`.

      The summary appears in the list.

- Choose the Reference from the list.

  The Reference is linked to the Entry.

11. Fill in the Sequence information.

For information on the fields of the Sequence form, see "The Sequence Form" on page B-31.

The number at the top of the Sequence form is the accession number for this sequence.

- Type in the definition.

  The definition should proceed from general to specific in this form: *Organism product (gene name)* "gene" or "m-RNA", *fraction of gene or mRNA*. An example is, "Human tropomyosin 2 beta (TPM2) gene, complete cds." The figure below shows another example. For more detailed information, see "Sequence Definitions" on page A-5.

```
Definition: Rattus norvegicus lymphotoxin (TNF-beta)
            gene, complete cds, tumor necrosis factor    +
```

Figure 5-1 A Definition line. The + sign means there is more of the definition.

12. Link in the Source entity.

In general, a new Source entity is linked to each Sequence. On some occasions, more than one Source is linked to a single sequence. In either case, the Source refers to a particular span of the Sequence.

- In the Sequence form, highlight the Sequence field.
- Type \eel.

  A new Source entity is created.

13. From the Source form, link in a Taxonomy entity.

- Highlight the Lowest Tax Node field and type `\eel`.

  The Taxonomy locator appears.

- Enter as much information as possible into the fields of the locator and use wildcards wherever possible.

*Note: The effective use of wildcards is important here. Use an asterisk in place of an upper case letters and special characters. For example, a search for Sprague-Dawley should use *prague*awley*.*

- `^X` to call up the list of items that satisfy the information entered.
- Highlight the correct item and Return.

  The taxonomy is linked in to the Lowest Tax Node field.

14. Fill out the remaining fields in the Source form.

For descriptions of these fields, see "The Source Form" on page B-33.

- Type in the Cell Line, Cell Type, Developmental Stage, Tissue Type, Library, Haplotype, and Sex/Mating Type if applicable.
- Set the Macronucleus, Provirus, Germline, and Complete fields to yes or no as applicable.
- Link in the Specific and Lab Hosts if applicable.

  Taxonomy nodes are linked into these fields. The Specific Host will be a particular individual or organism. Link in a Lab Host if the laboratory host is different from the Natural Host (found in the Taxonomy form). Avoid linking the same node into more than one field. For details and examples, see "The Source Form" on page B-33.

15. Follow the link to the Taxonomy entity.

- Highlight the Keyword field and type `\eel`.

  The Keyword locator appears.

- Enter the keyword and `^X`.

  The keyword appears in the list. Use wildcards if there is no match on the first try, especially in place of any punctuation or capitalization.

16. Make note of the Division code and the Abbreviation.

The three-letter Locus and Division codes are in the Taxonomy entity. (These codes are used in the Entry entity.) If either or both are missing, they can be found by moving up the Taxonomy tree.

- To move up the Taxonomy tree, highlight the Parent Node field and `^X`.

  The link is followed to a new Taxonomy entity.

- Continue to move up the tree until both the Division code and the Abbreviation are known. Figure 5-2 and Figure 5-3 illustrate this process. .

Note: Users other than GenBank staff will not be able to change any information in the Taxonomy form. GenBank Annotators will be able link in a Natural Host and, in some cases, add to the Taxonomy tree. If you need to create a new Taxonomy node contact mia@genome.lanl.gov .

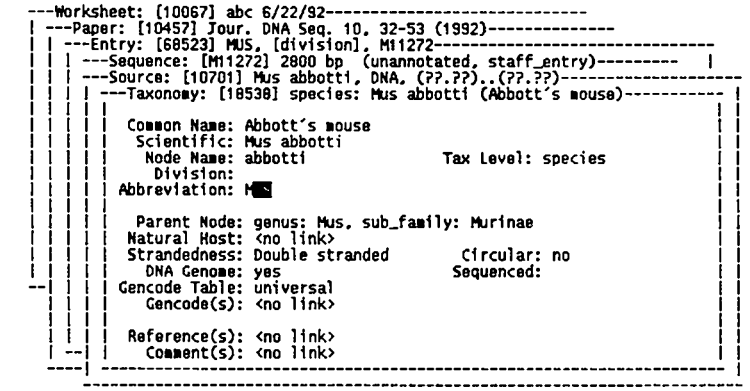

Figure 5-2 A Taxonomy form. The Abbreviation field is highlighed. To find the Division code, you must expand on the Parent node field to move up the Taxonomy tree.

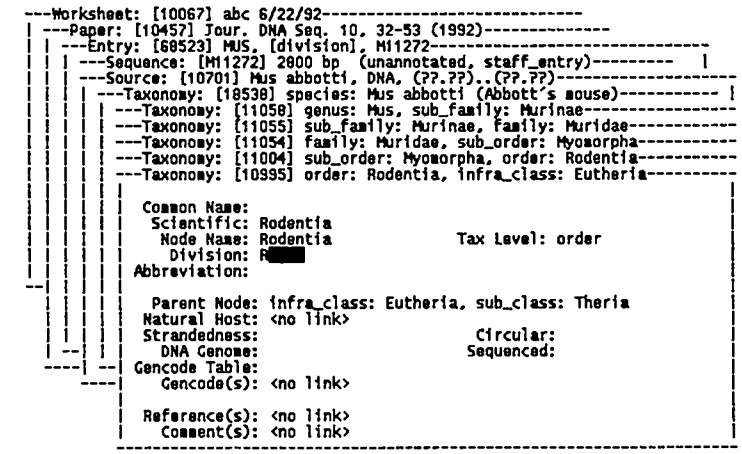

Figure 5-3 Moving up the Taxonomy tree. By expanding on the Parent Node field of each Taxonomy form, you will eventually come to one that contains the code you seek.

17. Quit the Taxonomy and Source forms. (Return to the Sequence form.)

If AWB does not let you leave, then the information in the form has been changed. Choose Reset from the Exit Form menu, and AWB will return the contents to their original state before it quits the form.

18. In the Sequence form, link in the Features.

- In the Features field type `\eel`.

  This will bring up a blank Feature form (not a locator box).

19. Fill in the Feature form.

For more information about the Feature form, see "The Feature Form" on page B-14.

- Link in the Feature Key with `\eel`.

  The list of Feature Keys will appear. Use Tab to move through the list or type the first few letters of the Feature Key to highlight it.

- Link in the Product if applicable.

  In the Product locator, use wildcards in place of questionable spelling or capitalization.

- Link in the Gene if applicable.

  In the Gene locator, use wildcards in place of questionable spelling or capitalization.

- Type in a Note if applicable.

  The Note field is used only when there is information that cannot be included as a Qualifier.

- Link any Qualifiers.

- Enter the Start and End ranges for the Feature span.

  If this is a virtual feature, there will not be a span, but components must be linked. See "Virtual Features" on page 5-8.

- Toggle the 5' and 3' complete fields.
- Type in the Replace String if applicable.

  See "Replace Strings" on page A-5.
- Toggle the Complement field to yes if the Feature is on the complementary strand.
- Toggle Exp Determined the Fits Consensus and fields.
- Link in any Comments or References.

20. If this is a Virtual Feature, set the operator and link in the component Features.

More information about virtual features can be found in "Virtual Features" on page 5-8.

```
---Feature: [321746] 321746: exon (1634.1634)..(1738.1738)-----------
|                                                                    |
|            Key: exon                                               |
|        Product: <no link>                                          |
|           Gene: TNF-beta, species: Rattus norvegicus (brown rat;>  |
|         Region: <no link>                                          |
|           Note:                                                    |
|       Sequence: L00981                                             |
|          Start:(1634      .1634      ) End:(1738      .1738      )|
| 5' end complete: yes        5' symbol:     Frame: 1                |
| 3' end complete: yes        3' Symbol:     Label:                  |
|        Operator:                                                   |
|  Replace String:                                                   |
|    Comp Feat(s): <no link>                                         |
|                                                                    |
|      Complement: no         Qualifier(s): /number = 2              |
|  Exp Determined:            Fits consensus:                        |
|    Reference(s): <no link>                                         |
|      Comment(s): <no link>                                         |
--------------------------------------------------------------------
```

Figure 5-4 The completed Feature form. The Feature Key is exon. No Product is linked to this Feature.

21. After the features are complete, back out of the forms.

As you quit the forms, the database is updated. Notice the status bar says, "Updating database, please wait..." As you go through, check each form to make sure all the fields are filled in correctly. Is the sequence definition complete?

AWB has various checks which run as you close the forms. These checks will produce an error message if they find a problem. The error message will give some indication of where the problem lies.

- Use the \qf (Quit: Form) command to exit the forms.

22. At the Entry form enter the Locus name and set the Division field using the three-letter abbreviations found in the Taxonomy form(s).

For more information about Locus name conventions, see "Locus Names" on page A-3.

23. When you reach the Paper form, set the Paper to Distribute.

- Type \sd (Special: Distribute).

  This is the command that sends out the new data; it is then available to everyone who uses the database.
- In the box that appears, set the correct Annotation Quality and Quantity.

24. Respond to the author, if appropriate.

For an off-site user this step is unnecessary.

- Type `\sre` or `\srp` to respond to the author email or through a letter to be faxed or mailed.

25. **To see the flatfile, type `\sm` (Special: Make flatfile).**
26. **Quit the Paper entity.**
27. **Quit the program.**
    - If you want to drop the worksheet from you list, type `\wd`.
    - `\qp` will quit the program.

5.2 Special Situations

5.2.1 Virtual Features

A virtual feature is any feature composed of other features. A complete list of the circumstances and types is impossible, but a few general examples can provide guidelines for their construction.

A Typical Virtual Feature

A very common virtual feature is the virtual mRNA. It is constructed by joining the exons from a sequence containing both exons and introns. The construction is illustrated in Figure 5-5.

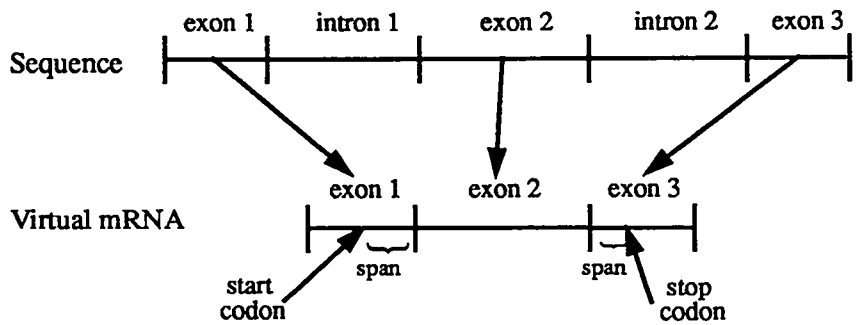

Figure 5-5 Constructing a virtual mRNA. The exons from the original sequence are joined end to end to make the virtual mRNA. The virtual cds, on the other hand, begins with a span from the first exon (from the start codon forward) and ends with a span of the last exon (until the stop codon).

Steps for Constructing a Virtual mRNA

Link in all the other features before using the following procedure. Also make a note of the database IDs of the component features.

These directions are for a virtual mRNA, but they may be used to make any virtual feature (with only minor adjustments).

1. **Call up a blank Feature form.**

   Make a note of the ID numbers of the component features before creating the new Feature entity.

   - Highlight the Feature field of the Sequence form and type `\eel`.

     The new Feature entity is created.

2. **Link in the component Features.**

   - Highlight the Comp Feature field.
   - Type `\eel`.

     The locator box appears.
   - Enter the database ID of the first component feature (exon 1) and `^X`.
   - Repeat these actions until all the components are linked.

3. **Choose the operator.**

   The operator will be "join" for this example. This means the feature spans should be placed end to end. (For more information about the various operators, see "The Feature Form" on page B-14.)

   - Highlight the operator field.
   - `^T` to toggle the field to the correct value.

4. **Complete the rest of the Feature form.**

   - Link in the Feature Key (mRNA for this example).
   - Link in the Product (if applicable, e.g., cds and mat_peptide).
   - Link in the Gene (if applicable).
   - Toggle the 5' and 3' Complete fields.
   - Set the Reading Frame.
   - Enter any applicable Note.

     Enter only those Notes that cannot go in the Qualifier field or anywhere else.
   - Link in any applicable Qualifiers.
   - Leave the Start and End fields blank.
   - Type \sa1 (Special: Amino Acid Translation: 1 letter code) to check the amino acid translation.

5. **Quit the Feature entity.**

   - Type `\qf`.

     Notice the new feature has the word "virtual" is in parentheses after it.

```
---Feature: [321753] 321753: mRNA (virtual)-----------------------------
|                                                                       |
|            Key: mRNA                                                  |
|        Product: <no link>                                             |
|           Gene: TNF-alpha, species: Rattus norvegicus (brown rat>     |
|         Region: <no link>                                             |
|           Note:                                                       |
|       Sequence: L00981                                                |
|          Start:(          .          ) End:(          .          )    |
| 5' end complete: yes        5' symbol:     Frame: 1                   |
| 3' end complete: yes        3' Symbol:     Label:                     |
|        Operator: join                                                 |
|  Replace String:                                                      |
|    Comp Feat(s): 321740: exon (4270.4270)..(4617.4617)                |
|                  321742: exon (5098.5098)..(5152.5152)              + |
|      Complement: no          Qualifier(s): <no link>                  |
|  Exp Determined:             Fits consensus:                          |
|    Reference(s): <no link>                                            |
|      Comment(s): <no link>                                            |
-----------------------------------------------------------------------
```

Figure 5-6 The completed Feature form for a virtual mRNA.

A virtual cds is constructed in the same manner as the virtual mRNA. The first component feature, however, is the span of an exon. This span begins with a start codon. At the end of the virtual cds is another span from an exon, this time ending with a stop codon. Use the join operator, just as with the mRNA.

A virtual intron is constructed when the 3' and 5' ends are given but a range of the intron in between is missing. Link in the two ends as component features and choose the order operator. Use the order operator, because the order is known, but the spans cannot be placed end to end.

The virtual intron provides a rule of thumb for constructing other virtual features; a virtual feature is required whenever the 5' end and 3' end are given but a section in between is missing.

An application of this rule is a LINE repeat (a long sequence that is repeated in many places). Often, only the 5' and 3' ends are given. A section (usually a large section) in between is not included with the sequence. The two ends should be linked in as component features of a virtual LINE repeat. Again, the operator to use is order.

5.2.2 Sequence Updates: Adding Sequence

It frequently happens that an author sends an update containing additional sequence to be added to a previously submitted sequence. The following instructions are for the case in which additional sequence is appended to the original sequence.

1. **Copy the sequence to be added to a file.**

   There are a number of ways to do this. One way is described in "Copying a Sequence to a File" on page 4-18. Pick up with step #4.

2. **Login to AWB and call up the Sequence entity.**
3. **Write the sequence text to a file.**
   - Use the \sw (Special: Write Sequence) command.
   -
4. **Merge the sequences.**

You can do this by concatenating the two files and sending the output to a third file. That is, file1 has the old sequence, file2 the additional sequence, and file3 contains the combined sequences.

- Type `cat file1 file2 > file3` and Return.

  Check the third file to be sure the additional sequence was added correctly.

5. **Delete the sequence text in the Sequence form.**
6. **Read in the new sequence.**
   - Use the `\sr` (Special: Read Sequence) command.
7. **Run any checks on the sequence.**

   For example, check the translation of a cds.
8. **Change the annotation accordingly.**

   You may need to add a Feature and/or change a Feature span.
9. **Link in the status "AN revision" and include a Comment.**

   This status must be added along with the Comment. Include the author's update message in the Comment text.

Chapter 6 Review

When a paper arrives in the annotation queue it has a complete citation, with all required entities properly linked, including one or more entries with a sequence linked to each. The precise steps taken to annotate these submissions differ from paper to paper. The following guide presents the basic procedures for a typical Authorin submission.

This section assumes the user has a copy of the submission in hand with a cover sheet that contains the rf_id and other dataflow information.

Off-site users: This chapter is for GenBank staff reviewing an Authorin submission.

GenBank staff: Use this chapter to review an Auhorin submission. The information in Appendix B will be helpful as you move through the forms.

- Section 6.1 contains the instructions for reviewing a typical submission.

6.1 Reviewing a Typical Submission

1. **Login to AWB and open a Worksheet.**

   If you do not know how to login to AWB, see Chapter 2.

2. **Link in the Paper.**

   - Type `\eel` in the Entity field of the Worksheet.
   - Choose Paper from the Entity List.
   - Enter the rf_id of the paper in hand in the Database ID field of the Paper locator.

     A list that contains the Paper appears.
   - Choose the paper from the list.

3. **Check the information in the Paper entity.**

   - Check the Publication, Volume, Issue, and Pages.
   - Make sure the contents of the Title field fit the convention.

     See "Titles" on page A-6. The first letter of the title should be capitalized. All other letters should be lower case except those that are capitalized by convention, e.g., DNA, Mus musculus, lacZ.
   - Make sure the Hold Date field has a date that has expired.

     If the submission was HUP (Hold-Until-Published) the release date should have expired.
   - Check the Pub Status (Publication Status).

     Look for inconsistencies. For example, there should not be a Publication linked to the Paper if the Pub Status is Unpublished. There must be a Publication linked if the Pub Status is Published.

4. **Expand to the Person entity of the corresponding author and check the information there.**

   It may be necessary to go through the Author list one at a time to find the corresponding author.

   - With the author's name highlighted, `^x` to expand to the Person form.

     The corresponding author should have a complete address, including a complete phone number and E-mail address.
   - Type `\qf` to return to the Paper form.

   Note: When the Respond to Author command is given, AWB first tries to respond to the Person linked to the Submission, then to the Person who has the Corresponding Author field set. Finally, it looks at the first Person on the Authors list in the Paper entity. For an E-mail response, if it cannot find an E-mail address, the command fails.

5. **Set the Status and Pub Status fields in the Paper entity.**

   The Statuses (Reference Statuses) give a history of a submission, from its arrival to its distribution. Each Status includes the name of the Person who set the Status and the date the Status was set.

   - Highlight the Status field.

- Type `\ss` (Special: Set status) and choose "AN annotation begun" from the list.

6. **Check the information in the Submission entity.**
   - From the Paper form, highlight the Submission field and type `^X`.
   - In the Submission form, toggle the medium to the correct value (`^T`).
   - Check to be sure the text of the Submission is in the Text field.

     You may Tab through the text of the Submission, or `\et` places you in the text editor.

     If the Submission is not there, use `\efr` (Edit: File: Read) to read in the text of the Submission. A box will appear which asks for the name of the input file. Enter the name of the file that contains the text of the Submission.
   - Make sure a Person, with a complete address, is linked to the Author field.
   - `\qf` to return to the Paper form.
7. **Check the information in the Entry entity.**

   The steps from here on will need to be repeated for each item in the Entry field.

   For information about the Entry form, see "Special Commands" on page B-10.
   - Highlight the first summary in the Entry field and `^X`.
   - Skip the Locus and Div fields.

     These are more easily done after the correct Source and Taxonomy forms have been linked.
   - Set the Mol Type and Topology fields to the correct values.

     The Topology should be set to Circular only if the sequence is complete and circular.
   - For segmented entries, enter the correct numbers in the Segment field.

     For example, if thiis Entry is for the second of four segments, the field should have 2 of 4.
   - In the Sec Acc(s) field, link in any Secondary Accession numbers.
   - To link in a Keyword, click MENU and choose Selector from the Keyword pop-up menu.

     The Keyword selector appears.
   - Type `\eel` in the keyword(s) field to link in the keywords.

     These are usually products and gene names. You should use wildcards in the Keyword locator that appears. Do not capitalize the first letter of product names.
   - `^X` after entering the Keyword name (with wildcards).
   - Select the keyword from the list the produced by the locator box.

     If the locator box fails to find the keyword, and you are sure the database does not contain the word you seek, you may create a new one by entering the word in the locator box and typing `y` when it asks if you want to create a new one.

   You will return to the Entry form as you back out of the forms with AWB.
8. **Link in the Source and Taxonomy entities.**
   - Highlight the Sequence field of the Entry form and `^X`.

- If the Source is linked in, highlight the field and `^x`.

  The link is followed to the Source entity.
- If no Source is linked in, highlight the Source field and type `\eel`.

  A new Source form appears.
- If a Taxonomy is linked to the Source, highlight the Lowest Tax Node field and `^x`.

  The link is followed to the Taxonomy entity.
- If there is no Taxonomy link, highlight the Lowest Tax Node field and type `\eel`.

  The Taxonmy locator box appears.
- Enter the scientific name of the lowest taxonomy node in the Scientific Name field of the locator box.

  Wildcards are recommended, and you should capitalize the first letter (for Genus and above).
- Enter any other useful information into the other fields of the locator box and `^x`.

  For example, it the lowest taxonomy node is species, you might enter "Bos*" in the Scientific Name field and "cow" in the Common Name field. `^x` calls up a list that contains "species: Bos bovis (cow)."
- Select the correct item from the list produce by the locator box.

  The taxonomy is now linked in.
- `^x` to expand to the Taxonomy form.
- In the Taxonomy form, make a note of the three-letter abbreviations in the Division and Abbreviation fields.

  You will use these in the Locus and Division fields of the Entry form.

  If either or both of these abbreviations is missing, they may be found by moving up the Taxonomy tree. This is done by following the link in the Parent Node field to a new Taxonomy entity. Continue following the links in this field until an entity is found that contains abbreviations.
- `\qf` to quit the original Taxonomy form(s) and enter any Gencode exceptions.

  For more information about these translation exceptions, see "Gencode Exceptions" on page A-2.
- `\qf` to return to the Source form.

9. **Check the information in the Source entity.**

   For a list of the fields, with descriptions, see "The Source Form" on page B-33.
   - Link in the Specific and Lab Hosts if necessary.

     Taxonomy nodes are linked into these fields. The Specific Host will be a particular individual or organism. Link in a Lab Host if the laboratory host is different from the natural host (the node linked to the Nat Host field found in the Taxonomy entity). Avoid linking the same node into more than one field.
   - Type `\qf` to quit the Source form.

     The Sequence entity returns.
10. **Complete the information in the Sequence entity.**

- Type the sequence definition into the Definition field.

  The definition should proceed from general to specific in this form: *Organism product (gene name)* "gene" or "mRNA", *fraction of gene or mRNA*.The gene name is in parentheses if there is a product. An example is, "Rattus rattus K+ channel protein (KSHIIIA3) mRNA, complete cds." The figure below shows another example. For more detailed information, see "Sequence Definitions" on page A-5.

```
Definition: Rattus norvegicus lymphotoxin (TNF-beta)
            gene, complete cds, tumor necrosis factor    +
```

Figure 6-1 Part of a completed Definition line.

- If this is a virtual sequence, link the sequence elements to the Seq Element(s) field.
- Toggle the Annot Quality and Annot Quality fields to the correct values.

  The Annotation Quantity is full if the submission has been thoroughly reviewed. The Annotation Quality is staff_review for new annotators whose work is reviewed and is staff_entry for experienced annotators.

11. Check the Features that are already linked in.

Check the fields of Feature form against the text of the submission to be sure the information has made in correctly into the database.

The authoritative document for Feature table syntax is "The DDBJ/EMBL/GenBank Feature Table: Definition." Refer to this document for detailed feature information.

For information on the fields of the Feature form, see "The Feature Form" on page B-14.

- Highlight the Feature and `^X`.

  The link is followed to the Feature entity.
- Check the Start and End fields.
- Check the Product.

  Is it the correct Product?
- Check the Gene link.
- Check the 5' and 3' Complete settings.
- Check the Qualifier and Note fields.

  Be sure that a Note is not something that could be linked as a Qualifier.
- Check the translation of a CDS, mat_peptide, or sig_peptide using the `\sa` (Special: Amino acid translation) command.

  You may choose a 1 or 3-letter code. If the translation fails, try adjusting the Reading Frame, or check the span to be sure the start codon is included for the CDS or sig_peptide. The start and stop codons should be included for a CDS, but the mat_peptide should not include either. Use the special Locate Span and Show Sequence commands to check the sequence.

12. Enter any new Features.

These will be Virtual Features or other Features that have not been entered automatically.

The authoritative document for Feature table syntax is "The DDBJ/EMBL/GenBank Feature Table: Definition." Refer to this document for detailed feature information.

For information on the fields of the Feature form, see "The Feature Form" on page B-14.

Note: Be careful when unlinking features. They will not be retrievable. If you attempt to unlink a Feature, but it remains, then it is probably linked to a Virtual Feature somewhere.

- `\eel` in the Feature field creates a new Feature entity.
- Highlight the Key field and type `\eel`.

  The Feature Key List appears. Select the correct key.
- If there is a product for this feature span, highlight the Product field and type `\eel`.

  The Product locator appears.
- Enter the Product name in the locator box and `^X`.

  As usual, wildcards are recommended, especially in place of hyphens and articles.
- Select the Product from the list.
- If this feature span is associated with a gene, highlight the Gene field and type `\eel`.

  The Gene Occurence locator appears.
- Enter the gene name and/or the organism scientific name and `^X`.
- Select the Gene from the list.

  If the Gene is not in the database, you will need to create a new entity.
- Enter the Start and End ranges for the feature span.

  The two fields for each allow the sequence to be labelled with fuzzy ends. For more information see "The Feature Form" on page B-14.
- Set the 5' and 3' complete fields.

  The options are yes and no.
- Set the Frame field for the number of the base pair that is the start of the first codon.
- Enter the Qualifiers and Notes.

  The DDBJ/EMBL/GenBank Feature Table: Definition document contains specific information about qualifiers.
- Type in any Replace String information.

  See "Replace Strings" on page A-5.
- If this is a Virtual Feature, link in the component Features to the Comp Feat(s) field and toggle the operator to the correct value.

  For more information about virtual features see "Virtual Features" on page 5-8.
- Toggle the Complement, Fits consensus, and Exp Determined fields.

- Check the translation of a CDS, mat_peptide, or sig_peptide using the `\sa` (Special: Amino acid translation) command.

  You may choose a 1 or 3-letter code. If the translation fails, try adjusting the Reading Frame, or check the span to be sure the start codon is included for the CDS or sig_peptide. The start and stop codons should be included for a CDS, but the mat_peptide should not include either. Use the special Locate Span and Show Sequence commands to check the sequence.
- Type `\qf` to return to the Sequence form.
- Repeat these steps for each feature.

13. Fill in the remaining fields of the Sequence and Entry forms.

- At the Sequence form, check again to be sure the Definition line is complete.
- Quit to the Entry form.
- At the Entry entity, type the Locus name into the Locus field and set the Div field to the correct value.

  The first three letters of the locus name will be the three letter abbreviation found in the Abbreviation field of the Taxonomy form. The next three letters should indicate the product. The remaining letters, up to a total of ten, are for uniqueness. See "Locus Names" on page A-3 for more information.

  The Division code is the three-letter code found in the Division field of the Taxonomy form.
- Make sure all the Keywords have been linked in.
- Quit to the Paper entity.

14. Repeat steps 7 through 12 for each item in the Entry field.

The `\se` (Special: Entry template) command may be very useful for multiple Entries. This command will copy the contents of the forms linked to the highlighted Entry. That is, the entities just completed will be a template for the entities linked to the other Entries. You will need to check each of the entities and make some changes. (The Locus field will not be copied.)

Use the Entry template command when the contents of the Entries are very similar.

- When all the Entries are complete, quit to the Paper entity.

15. Respond to the author.

- Type `\sre` or `\srp` to respond to the author email or through a letter to be faxed or mailed.

  AWB will generate a standard letter with a flatfile report, and place you in the text editor were you may inspect and edit the letter before printing or mailing it.

16. Set the information to Distribute.

- Type `\sd` (Special: Distribute).

  This is the command that sends out the new data; it is then available to everyone who uses the database.
- Set the Annot Quantity and Annot Quality fields (if they are not already correct) and `^X`.

17. When the review is complete, quit the Paper and unlink the Paper from the Worksheet.

(An option to this step is to drop the entire worksheet at the end of the day, and then it is not necessary to unlink each paper. \wd is the command to drop a worksheet.)

- Quit to the Worksheet.
- With the paper highlighted, type \eeu (Edit: Entity: Unlink).
  The paper disappears from the ENTITY list.
- To quit the program, type \qp.

Chapter 7 In-House Curation

Off-site users: This chapter describes in-house GenBank tasks for maintaining the database.

GenBank staff: This chapter contains the description of the curation tasks given to individual members of the annotation and dataflow staff.

- Section 7.3 describes the task of updating gene names.
- Section 7.4 contains the procedures for maintaining the Person table.
- Section 7.5 contains the procedures for maintaining the Reference table.
- Section 7.6 contains the procedures for maintaining the Taxonomy table.

7.1 Overview

The tasks described in this section involve an orthogonal view of curation. That is, the person responsible for a particular area should be looking across the database to be sure that area is in order. This is accomplished by using the regularly generated reports as a springboard for further investigation, either alone or with the help of software.

For example, the person in charge of miscellaneous features will examine the regular reports of the features assigned the misc_feature key. She/he peruses the list to be certain that standard conventions are being maintained. If an error is found, the curator corrects it and also looks through the database for other similar errors. The in-house curator should also communicate with the GenBank software department regarding possible software checks for errors.

7.2 GDB Links

7.2.1 Summary

A daily report of all submissions that have a human locus name is sent to the person charged with this task. The person compares the products, genes, and map locations with the information in the hugene.dc2 document. If the locus of the GDB name differs from the gene name of the GenBank entry, the gene name in the GenBank entry is changed.

Use the humgene program to search humgene.dc2 for the products, map locations, or gene names in the entry.

Keep a list of the products which do not have a corresponding gene name in humgene.dc2. When the list grows large, send it in an email message to Michael Chipperfield at chipper@welchgate.jhu.edu.

```
Locus: HUMCD441
  Keyword(s): CD44 gene,
  Origin:
  Old Source: Homo sapiens RNA.
Sequence:
  Definition: H.sapiens CD44R mRNA
  Features:
    ID: 303711
    Key: mat_peptide
    Note:
    Gene: CD44R1
    ID: 303712
    Key: sig_peptide
    Note:
    ID: 303710
    Key: CDS
    Note:
    Gene: CD44R1
```

Figure 7-1 Part of the daily HUM report. The gene name CD44R1 will be changed to CD44 to match the name in the humgene.dc2 document.

7.2.2 Procedures

1. **Check the product names on the report.**
   - Run each name through the Humgene program by typing `humgene` *`productname`* and Return. (You may use gene name or map location as well as products.)

     The program will print lines that contain the product name. Gene names will correspond with products. That is, the product will be listed in humgene.dc2 with the official GDB gene name. The gene name is the first item on the line, the product name is the last. If you search for a product that has more than one word, put the words in quotation marks.
   - If the product is not in the humgene file, keep a record of it in a file.

     Include the author's gene name, map location, and/or keywords, if present. When the file begins to get large, send it to Johns Hopkins.
   - For all products that are in the file, check the locus name (the first item in the row) and compare it with the gene name.

2. **Change gene names to correspond with the locus names in the humgene document.**

   Use AWB.

   - Use Inquiry: Locate to pull up the feature which has a gene name to be changed or added.
   - Unlink an incorrect gene name by highlighting the Gene field and typing `/eeu`.
   - Type `\eel` to link in the new gene.

     The Gene locator appears.
   - Enter the correct gene name and `^x`.
   - Select the gene from the list and Return.

7.3 The Person Table

The Person Table contains the information about individuals in the database. These include author

7.3.1 Summary

A daily report of the new additions to the Person table is sent to the person charged with this task. The list should be checked for common errors, such as incorrect suffixes, initials without periods, all capitals, and repetition of names. The report also contains a list of possible errors which should also be inspected.

```
[102708]  Rock              Daniel          L
[102723]  Ruiz-Larrea       Fernanda
[102713]  Savard            Louise
[102686]  Sechi             Leonardo        A
[102688]  Sechi             Leonardo        A.
[102683]  Sherr             Elliott         H
[102691]  Silver            Pamela          A.
[102712]  Skalka            Christian       E.
[102748]  Skorupa           Grzegorz
[102695]  Sneider           Judith          M.      Ph.d.
[102755]  Stec              Wojciech        J.
[102749]  Szczepanek        Andrzej
[102674]  Tang              Chieh-Ju C.
[102678]  Tang              Tang K.
[102724]  Thompson          Andrew
[102725]  Totty             Nicholas        F.
[102696]  Trent             Dennis          W.
```

Figure 7-2 Part of the daily Person table report. The fourth and fifth names on the list are identical and should be combined (unless the addresses are different. The Ph.d. suffix on the tenth name should be removed.

7.3.2 Procedures

The following procedures assume familiarity with AWB.

Before beginning these procedures, get a copy of the daily Person table report and logon to AWB.

1. **Combine all duplicate names.**

   The names should be exact copies, but with separate ID numbers.

   - Open a Worksheet in AWB and link in the two Person entities.

     Use the ID numbers from the Person table report.
   - Expand the Institution field to check the addresses of each person.

     If the addresses are the same, then the entities need to be combined. If they are different, then assume they are separate people.
   - Use the Edit: Entity: Combine command to combine repetitions.

     Do this from the Worksheet level. When the command is issued, AWB will ask for the ID of the retained person and the ID of the replaced person.

2. **Remove all incorrect suffixes**

   Suffixes should be part of the name, like Jr. or III, not Ph.d.

   - Open a Worksheet in AWB and link in the Person entity.
   - Expand to the Person form.

- Use the space bar in the Overstrike mode to remove the suffix.
- Exit the form.

3. **Add periods to all initials that do not have them, and change other punctuation and capitalization as needed.**

   If a name is written in all capitals, for example, it should be changed to have only the first letter capitalized.

   - Open a Worksheet in AWB and link in the Person entity.
   - Expand to the Person form.
   - In either the Insert or Overstrike editing modes, add the period to the initial, and change other punctuation as needed.
   - Exit the form.

7.4 The Reference Table

7.4.1 Summary

Each day a list of the reference information entered into the database is sent to the person charged with curating the Publication table. The list includes the reference ID, publication abbreviation with volume and page span, the year of publication, and the publication status of the paper.

The curator checks the list for inconsistences. For example, a paper may have the status "In Preparation" but also have a volume and page span. Or it may be listed as "Published" but not have a page span. Perhaps the page span goes from a higher number to a lower number. Checking and correcting these errors may require using AWB to view the contents of the submission.

```
-
NEW ROWS TO REFERENCE TABLE:
-
-
RF ID    Pub Abbrev                          Volume Page Span   Year   Pub Stat
-------- ----------------------------------- ------ ----------- ------ ---------
[150849] DNA                                 []     -           (1992) In Prepar
[150852] Gene                                []     -           (1992) In Prepar
[150873] Nucleic Acids Res.                  []     -           (1992) In Prepar
[150851] Plant Physiol.                      []     -           (1992) In Prepar
[150854] Proc. Natl. Acad. Sci. U.S.A.       []     -           (1992) In Prepar
[150874] Proc. Natl. Acad. Sci. U.S.A.       []     -           (1992) In Press
[150857] Gene                                [79]   9-20        (1989) Published
[150883] Nature                              [358]  80-86       (1992) Published
[150868] Nucleic Acids Res.                  []     -           (1992) Published
[150863] Proc. Natl. Acad. Sci. U.S.A.       [89]   6716-6720   (1992) Published
[150864] Proc. Natl. Acad. Sci. U.S.A.       [89]   6731-6735   (1992) Published
[150867] Proc. Natl. Acad. Sci. U.S.A.       [89]   6765-6769   (1992) Published
[150869] Proc. Natl. Acad. Sci. U.S.A.       [89]   6770-6774   (1992) Published
[150870] Proc. Natl. Acad. Sci. U.S.A.       [89]   6798-6802   (1992) Published
[150886] Proc. Natl. Acad. Sci. U.S.A.       [89]   7060-7064   (1992) Published
[150887] Proc. Natl. Acad. Sci. U.S.A.       [89]   7080-7084   (1992) Published
[150888] Proc. Natl. Acad. Sci. U.S.A.       [89]   7085-7089   (1992) Published
[150855] Cell                                []     -           (1992) Submitted
[150866] Infect. Immun.                      []     -           (1992) Submitted
[150889] Proc. Natl. Acad. Sci. U.S.A.       []     -           (1992) Submitted
[150871] Virology                            []     -           (1992) Submitted
-
```

Figure 7-3 A typical Reference table report. Notice the ninth row from the top has the status "Published", yet there is no volume or page span. This is an item that should be check.

7.4.2 Procedures

The following are the steps for checking the Reference table.

1. **Check for consistency based on the Pub Status.**
   - The journal, volume and page span should be present if the status in "Published."

     In the event the status is "Published" but some of the fields are not complete, check the submission information using AWB. It may be necessary to send a message to the author to clear inconsistencies.
   - The volume and pages spans should be blank if the status is "Unpublished."

     If some of the fields are filled out, perhaps the status needs to be changed. Again, it may be necessary to send a message to the author.

- If the status is "In Press" but there is no journal listed, the status should be changed to "In Preparation."

2. **Check the journal abbreviation.**

   Is the abbreviation correct? A list of journal abbreviations is in ~genbank/jourabbs.

3. **Check to see that the volume is correct.**

   Does the volume of the journal agree with the volume numbers for the year of publication?

4. **Check the page spans.**
   - Be sure the numbers do not go from higher to lower.
   - Be sure the page spans are reasonable.

     A 200 page paper, for example, is cause for question.

7.5 Taxonomy

The Taxonomy tables contain the taxonomic classification information about organisms associated with sequences and their hosts. They are continually updated.

This section contains an overview of the task of maintaining the Taxonomy tables, the location of the lists of resources for classifying organisms, and step-by-step procedures for the most frequent tasks of Taxonomy maintenance.

7.5.1 Overview

The Taxonomy table contains information that allows for specific and clear identification of the organisms associated with sequences in the GenBank database. The information is maintained as a Taxonomy tree, with each node of the tree as a Taxonomy entity. To identify an organism associated with a Sequence, a Taxonomy entity is linked to the Source entity. This Taxonomy entity is the lowest Taxonomy node for the organism. It is necessary to identify the organism associated with the sequence in the entry, as well as the biological hosts of that organism, if applicable.

The genetic code (Gencode) varies among kingdoms. For example, the stop codon (tga) in the Universal Gencode is read as tryptophan in the Mitochondrial Gencode. When a Gencode is linked to a Taxonomy node, that Gencode applies to all the nodes below it on the Taxonomy tree. The correct placement of these different genetic codes is also the responsibility of the Taxonomy curator.

The Taxonomy of the Organism and Its Hosts

When a sequence is entered in the database, a Source is linked to it. To identify the Source, the lowest Taxonomy node of the organism is linked to the Source entity. If the organism (for example, rotavirus) is normally found in another organism (e.g., a human), an entity corresponding to the Genus/species or strain of the host organism is linked to the Taxonomy entity as a Natural Host. If a more specific organism (e.g., a particular patient who has the virus) is known, it is linked to the Source entity as a Specific Host. Finally, the lowest Taxonomy node of an organism, cell line, or tissue used in the laboratory as a host for propagating the sequence is linked to the Source entity as a Laboratory Host. More than one of these Natural, Specific, or Lab Hosts may be associated with a particular Sequence.

Changes in the Taxonomy Tree

The Taxonomy tree is not static. It is continually revised and updated by the GenBank Taxonomy curator. New Taxonomy nodes are added daily, and changes in the classification of organisms may require a major revision of the Taxonomy tree. A new node is added if, for example, a sequence is taken from an individual isolate of an organism that is in the database at only the Genus/species level. The individual/isolate is then added below the Genus/species node.

The continual fluidity of Taxonomy calls for regular changes in large subsections of the taxonomy tree, which affects the database as a whole. An example of the type of a global change that is made to the database when the taxonomy tree changes is the

addition of a new Genus to the database when a species is found to be different enough to be classified separately. For example, the Genus Aspergillus has a variety of species below it: asculatus, flavus, niger, etc. One of these species, Aspergillus nidulans, was found to be different enough to merit a new Genus. So a new Genus, Emericella, was added, and all the Aspergillus nidulans were changed to Emericella nidulans. This change required that every entity to which the Aspergillus nidulans had been linked must now be changed to the Emericella nidulans.

Note: The old name (Aspergillus nidulans) is stored as a Common Name so that it can be found, even though the official name was changed.

7.5.2 Resources

A list of persons and organizations to contact with regard to Taxonomic information is in the file /usr/gb/lookup/TaxResources. The file also contains a list of books and journal resources.

7.5.3 Procedures

A computer generated report of all the Taxonomy entities that were added to the database the previous day is sent to the Taxonomy curator. Since the taxonomies were, for the most part, added by the curator, they should be familiar. If an unknown organism is on the list, check it with the procedures below.

Daily Taxonomy Check

A new group of papers is queued for annotation every day. The Taxonomy curator takes the papers and checks through them for new Taxonomies. It is her/his responsibility to identify the unknown organisms and add new Taxonomy entities to the database when necessary.

1. **Check the papers for new Taxonomies.**

   The appearance of the taxonomies differs between Authorin submissions and GenBank Submission Forms. The lines from both forms appear in the figures below. If the Taxonomy information is unknown, the new information needs to be added to the database.

```
entity ( taxonomy.1.10 ) {
node_name = "pRTX, pRTSI, pRTSII";
entity ( parent_taxonomy.1.10 ) {
node_name = "Lambda DASH II from Stratagene";
entity ( parent_taxonomy.1.10 ) {
node_name = "Sprague Dawley";
entity ( parent_taxonomy.1.10 ) {
sci_name = "Rattus norvegicus";
common_name = "Rat";
is_dna = "yes";
strandedness = "double stranded";
is_circle = "no";
entity ( natural_host_taxonomy.1.10 ) {
is_circle = "no";
}/* end natural_host_taxonomy entity */
entity ( parent_taxonomy.1.10 ) {
node_name = "Muridae";
entity ( parent_taxonomy.1.10 ) {
node_name = "Rodentia";
entity ( parent_taxonomy.1.10 ) {
node_name = "Mammalia";
entity ( parent_taxonomy.1.10 ) {
node_name = "Chordata";
entity ( parent_taxonomy.1.10 ) {
node_name = "Animalia";
```

Figure 7-4 The Taxonomy information in an Authorin file.

```
-------------------------------------------------------------------------
The following items refer to the original source of the molecule you have sequen
ced.
  organism (species) name (e.g., Escherichia coli; Mus musculus)
Escherichia coli
-------------------------------------------------------------------------
  sub-species NA                          strain (e.g., K12; BALB/c) K12
-------------------------------------------------------------------------
  name/number of individual/isolate (e.g., patient 123; influenza virus
  A/PR/8/34) DB6430
-------------------------------------------------------------------------
```

Figure 7-5 The Taxonomy information in the GenBank Submission Form.

2. **For unknown organisms, check in the database with AWB.**
   - Login to AWB.
   - Open a Worksheet.
   - Type `\eel` to call up the Entities list.
   - Choose Taxonomy from the Entities list.

     The Taxonomy locator appears.
   - Enter the Taxonomy information in the fields of the locator and `^X`.
   - If there is no match, try again using wildcards (*) where the spelling is questionable.
   - If a number of items appear on the list, look at each of them to see if one contains the correct Taxonomy information, and check for duplicates with almost identical spelling.

     Duplicates need to be combined. Use the Edit: Combine command.

Researching Unknown Organisms

Some organisms will be completely unfamiliar. Use the resources available to determine the full Taxonomy of the organism.

3. **Check with the author for more information.**

This is the surest method for nailing down the organism taxonomy. There are occasions when the author is not certain about the taxonomy of the organism, and an agreement must be reached as to how to proceed.

4. **Check the professional and print resources.**

   If, for any other reason, more research is required, use the list of resources in /usr/gb/lookup/TaxResources. This list includes experts in many fields, as well as general resource information. The contents of the book "Synopsis and Classification of Living Organisms" are organized in the file /usr/gb/lookup/master.tax.orig.

   Experts are called under three circumstances: 1.) the author of the submission is in a foreign country and hard to reach, 2.) the author does not know the taxonomy of the organism, or 3.) there is a conflict between the author's information and the information currently in the database.

Adding New Taxonomies to the Database

If the Taxonomy cannot be found in the database, and all the necessary Taxonomic information is known (e.g., parent nodes and Taxonomy levels), use AWB to add the new node to the Taxonomy tree.

5. **Create a new Taxonomy entity.**
   - In AWB open a Worksheet and type `\eel`.
   - Choose Taxonomy from the Entities list.

     The Taxonomy locator appears.
   - Enter the information into the fields of the locator and `^X`.

     A new Taxonomy entity appears.
   - Complete the Scientific Name, Common Name (if known), and Node name fields.

     These are all text fields. The Node Name is the same as the Scientific Name except at the Genus/species level the Node Name is the species.
6. **Link in the Tax Level.**
   - Highlight the Tax Level field and type `\eo`.

     The Tax Level list appears.
   - Choose the correct level from the list.

     The Tax Level is linked.
7. **Link in the Parent Node.**
   - Highlight the Parent Node field and type `\eel`.

     The Taxonomy locator appears. (The Parent Node is, of course, another Taxonomy node.)
   - Enter the Taxonomy information into the fields and `^X`.

     A list of entities should appear. If there is no match, try again with more information, and use wildcards. In general, there must be a Parent Node for every Taxonomy node. If it is not in the database, the Parent Node needs to be added.
8. **Complete the remaining fields of the Taxonomy form.**

- Link in the Natural Host, if applicable.

  This is the Taxonomy node of the organism which functions as a natural host to the current organism. For example, Homo sapiens (human, man) is the Natural host for HIV-1.
- Type in the Abbreviation.
- Set the Gencode Table.

  See Gencode Exceptions on page A-2 for more information about Gencodes. Also see The Taxonomy Form on page B-37 for specific field information.
- Set the Division if the organism is a mitochondrion, chloroplast, or bacteriophage.

  In most cases the Division is set at a higher level and does not need to be set at the lower levels.
- Link in any References or Comments.
- Link in specific Gencode exceptions if applicable.

  Most of the these should be handled using the transl_except as a Feature Qualifier in the coding region. See Gencode Exceptions on page A-2.
- Set the DNA Genome, Circular, and Strandedness fields.
- When the form is complete, type `\qf` to exit the form.

Making Global Changes to the Taxonomy Tree

It is sometimes necessary to make changes to the database when new information is revealed or official decisions are made which change the classification of a large section of the Taxonomy tree.

For example, a family may have five genii below it on the Taxonomy tree. If it is found that two of them belong to a new family, a change in the tree is required. Follow these steps to implement such changes.

1. **Bring up the Taxonomy entity.**
   - Login to AWB.
   - Open a Worksheet and type `\eel`.
   - Choose Taxonomy from the Entities list.

     The Taxonomy locator appears.
   - Enter the information to call up the Taxonomy entity to be changed and `^X`.

     Choose the entity from the list.
2. **Change the Taxonomy entity to reflect the new information.**
   - Change the old name to the new name and enter the old name in the Common Name field.
   - If necessary, unlink the old Tax Level and link in a new one.
   - If necessary, unlink the Parent Node and link in a new one.

     It may be necessary to create a new Parent Node. In that case, highlight the Parent Node field, type `\eel` , `^X` and then `y` (yes, create a new entity). When the new entity appears, fill in the appropriate information.
3. **Link in new entities until you reach a level already in the database.**

Appendix A Conventions

The following sections contain guidelines for some of the common conventions employed by GenBank annotators when annotating or reviewing a submission. The list is not exhaustive and more information may be found in other documents. The overview found in the /usr/gb/doc directory lists other useful documents and their directory locations.

Off-site users: Use this appendix to apply GenBank conventions to your annotation. Topics of this chapter include:

- the Gencode setting in the Taxonomy form and Gencodes linked to Taxonomy and Feature entities,
- Keyword conventions,
- homology conventions,
- conventions regarding the Locus field of the Entry form,
- conventions for linking Reference entities,
- conventions for setting the Replace String and Operator fields in the Feature form,
- conventions for filling in the Definition line of the Sequence form,
- conventions for filling in the Title field of the Paper and Reference forms.

GenBank staff: GenBank annotators should refer to the sections in this appendix as needed for your annotation.

A.1 Gencode Exceptions

The "universal" gencode is the coding pattern that applies to most higher organisms. When an exception to the coding pattern occurs, it is handled in one of three ways:

(1) Patterns which fit the known patterns (universal, mitochondria, artificial, chloroplast, genetic element, and prokaryota) are entered in the Taxonomy form. The Gencode Table field can be toggled to one of the six options listed above. That code then applies to all organisms below it on the taxonomy tree. For example, Mitochondrion Drosophila melanogaster has its Gencode Table field toggled to mitochondria. This coding pattern then applies to the Bretagne strain of Mitochondrion Drosophila melanogaster.

(2) A translation exception which occurs throughout the coding sequence of a particular organism, but does not fall under one of the six categories in the Gencode Table is linked to the Gencode(s) field of the Taxonomy form of the lowest Taxonomy node for the organism. This coding exception will again apply to all organisms below it on the Taxonomy tree.

(3) A translation exception that pertains only to a particular site on a particular sequence is linked in at the Feature table as a qualifier. The Featqual is transl_except and the value is the location of the codon that is an exception. For example, on rare occasions, tga codes for Seleno Cysteine in eukaryotes. (tga is normally a stop codon.) For this example the Featqual is transl_except; the value is (pos: `bp..bp`, aa: other), including the parentheses, where the bp's are the numbers of the base pairs that start and end the exception. The name of the acid coded for (Seleno Cysteine, in this example) is placed in a note.

A.2 Homologies

GenBank conventions distinguish two types of homologies: a strong and a week type. The weak type is used when a sequence has limited regions of homology and the person annotating the sequence wants to make note of the similarity without making a claim of equality. For the weak case there is no Product linked to the Feature and no mention is made of the homology in the Sequence Definition. A /note Qualifier is added that begins with the work "homology" and may or may not include a number indicating the degree of similarity. In the second, stronger case, the homology is extensive enough to assert that the proteins are closely related or identical in function. In this case a product is linked to the Feature and the product is mentioned in the definition. The word "homologue" should be the first word of the /note Qualifier in that goes with the Feature.

An example of the weaker case let us say that a coding region has a 50% similarity to a region that codes for a zinc-finger protein. An appropriate Qualifier would look like this:

/note = "homology in base pairs 100-200 to X. laevis zinc-finger protein TFIIIA zinc-binding domain (50%).

The first word of the note should be "homology" and it may or may not include a number indicating the degree of the similarity. In this case, no product is linked to the Feature nor would one be mentioned in the sequence definition.

As an example of the stronger case, let us say that a given coding region was similar enough to an E. coli galactosidase that annotator could conclude that the region codes for galactosidase, then the product galactosidase should be linked to the feature and mentioned in the definition. The note for such a homology would look like this:

/note = "homologue to E. coli galactosidase gene (90%)"

If the rest of the information is unavailable, the note may be simpler.

/note = "homologue"

The stronger word "homologue" should be the first word of the note. The word "homologue" should not appear in the product name that is linked to the Feature, i.e., the product name should just be the name of the protein.

A.3 Keywords

Keywords are words that help identify a specific Entry. The keywords are linked into the Keyword(s) field of the Entry form. These words include products (for a coding sequence) and gene names (only in the case of a bacteria or yeast).

Any other words that contain significant information should also be linked to the Keyword(s) field. For example, germline, processed gene, and pseudogene may be linked to the Keyword(s) field of an immunoglobulin entry.

Some sequences have specific keywords associated with them. Structural information, products, regulatory mechanisms, repeat regions, and binding sites are examples of the type of information that go into the Keyword(s) field. Check the documentation in the usr/gb/doc/annot directory.

Note: Proceed with caution when changing the contents of the Keyword form. Changes made to a keyword affect the appearance of that word everywhere it is used. In other words, a single keyword may be linked to hundreds of entities, and any change to that word appears in all of the entities to which it is linked.

A.4 Locus Names

The Locus field in the Entry form contains a name that is used to identify the entry. The first three letters of the name will be the organism code found in the Abbreviation field of the Taxonomy entity linked to the Source of a Sequence.

The next two are an indication of the product or one of the following.

IG	if the sequence is part of an immunoglobulin,
MH	if the sequence is part of a major histocompatibility complex,
CG	if the sequence is the complete genome,
TG	if it is part of a transfer RNA gene,
TR	if the sequence is part of a transfer RNA,
MT	if the sequence is from a mitochondrion,
CP	if it is from a chloroplast,
RG	if the sequence is part of a ribosomal RNA gene,
RR	if it is part of a mature ribosomal RNA,

More specific information about annotating these sequences can be found in the following files in the /usr/gb/doc/annot directory:

Note: New annotators should take the time to examine these files.

flu.dc4	contains information for annotating influenza sequences,
globin.dc4	contains information for annotating globin entries,
ig.dc4	contains information for annotating immunoglobulins,
mhc.dc4	contains information for annotating major histocompatibility complex genes,
rrna.dc4	contains information for annotating ribosomal RNA sequences,
trna.dc4	contains information for annotating transfer RNA sequences,

If the entry is segmented, the last letter should be the number of the segment. That is, the locus name should end with 01, 02, 03,... (or 001, 002, 003,... if necessary).

A.5 Reference Links

Almost every form has a field to which a reference may be linked. However, particular references are linked to specific forms depending on the type of reference, as follows:

A primary reference is one that shows an original sequence for the first time.

A "sites" reference is one that reports new features of a sequence already in the database. (Sites, rather than cites, because the reference is to a site on the sequence.)

The primary reference is linked to the Entry, Sequence, and Feature entities.

A sites reference is linked to the Entry entity and, possibly, to the Feature entity, but to no others.

The reference linked to the Sequence form will appear in the flatfile as the primary reference. Other references linked to the Entry form will appear as sites references in the flatfile report. Other references linked to the Feature form will appear as "/ references" in the flatfile report.

A.6 Replace Strings

In the Feature form, when the Replace operator is chosen, a replace string must be included. The Replace String field will contain the letters of bases that represent the replacement. This may involve only the replacement of some bases with others, or it may involve the insertion or deletion of some bases.

Take, for example, the series of bases: attacggt numbered 1 through 8. To replace bases 3 through 5 with the string aaa, the Start and End fields will have the numbers 3 and 5, and the Replace String field will have the letters aaa.

To insert bases, put the numbers of the two bases that bound the insertion in the Replace string field. For example, to insert the bases aaa between bases 5 and 6, put the numbers 5 and 6 in the Start and End fields, respectively. The Replace String field has the letters caaag. Notice, when an insertion is made, the boundaries of the insertion (cg) are included in the replace string.

To delete bases, put the numbers of the two bases that bound the deletion in the Start and End fields, and the letters of the two boundary bases in the Replace String field. For example, to delete bases 3 through 7 in the series above, put the numbers 2 and 8 in the Start and End fields respectively. The bases in the Replace String field are tt, representing only bases 2 and 8.

A.7 Sequence Definitions

The sequence definition, which is typed into the Definition field of the Sequence form, contains seven elements.

1. First enter the genus species name of the organism sequenced, e.g., Homo sapiens, Ovis aries, or Bos bovis. For common organisms it is permissible to a common name, e.g., Human, Sheep, or Bovine.

2. Next, type the name of the product (if known).

3. The gene name (if different from the product) should be typed in parentheses after the product. If there is no product, then the parentheses are unnecessary.

4. Next, type the word gene if the sequence is DNA, or mRNA if it is a messenger RNA.

5. A comma should follow "gene" or "mRNA."

6. Enter "complete cds", "5' end", "3' end", or "partial cds" after the comma (whichever is most appropriate). If there is no cds in the sequence, then describe a feature of the sequence. An example is, "Homo sapiens Alu repeat sequence."

7. Finally, end the definition with a period.

The definition may be very short if the sequence does not have a product, gene name, or features associated with it. For example, "Gallus gallus DNA sequence." is a possible definition line.

Below are three examples.

Lactococcus lactis aminopeptidase N (pepN) gene, 5' end.

Homo Sapiens aromatic decarboxylase gene, exon 4.

Drosophila virilis mastermind protein gene, complete cds.

A.8 Titles

The Title field in the Paper form contains the title of the journal article (or other document) that contains a sequence.

Capitalize the first letter of a word following a colon. Do not include a period at the end of a title.

Here are two examples:

The cadC gene product of alkaliphilic Bacillus firmus OF4 partially restores Na+-resistance to an Escherichia coli strain lacking Na+/H+

The sulfatase gene family: Cross-species PCR cloning using MOPAC technique

Appendix B ASCII AWB: Reference

Off-site users: Use this appendix as you move through the database with AWB.

- Section B.1 describes the AWB commands.
- Section B.2 describes all the forms and fields in AWB with the special commands that are specific to a particular form.

GenBank staff: Use this appendix as you move through AWB to become familiar with the commands, forms, and fields.

- Section B.1 describes the AWB commands.
- Section B.2 describes all the forms and fields in AWB with the special commands that are specific to a particular form.

B.1 AWB Commands

All commands begin with a \. Next, type the first letter of the desired item on the menu bar. For example, \e pulls down the Edit menu. Continue typing the first letter of the desired items as they appear on the menus. \eel chooses Edit, then Entity, and finally Link from the menus that appear.

B.1.1 The Bulletin Menu

The Bulletin menu has only one option: Software Report.

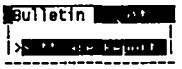

Figure B-1 The Bulletin Menu

Software Report calls up a window to type messages that are then sent to the GenBank Software department.

Software reports are usually sent when a user has some problem with AWB. Here are some guidelines for making a thorough software report:

Include a specific description of what you were doing when the problem occurred. The accession numbers or reference IDs of the sequences or papers involved should also be included.

If you were given any error messages, include those in the software report.

If AWB did not respond in a way that you expected, explain the difference between what you expected and the response you received.

If you tried to correct the problem yourself, explain what you did and what the results were.

B.1.2 The Edit Menu

The Edit menu contains options for modifying text, linking and unlinking entities, reading and writing text from files, combining, searching, copying, and saving entities.

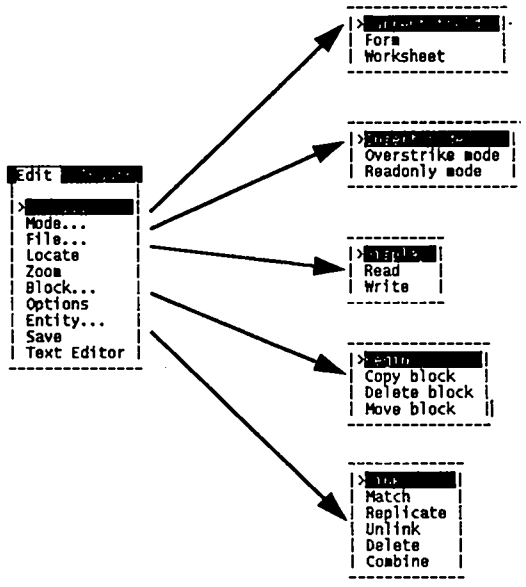

Figure B-2 The Edit Menu with submenus.

Undo: Current Field changes the content of the current toggle or text field to what it was before any editing was done.

Undo: Form changes the contents of all toggle or text fields in a form to what they were before any editing was done. This command does not apply after the contents of the form have been saved.

Undo: Worksheet undoes changes in the Worksheet text fields.

Mode: Insert changes the editing mode to insert text without removing any text that was already there. ^O will also change the mode.

Mode: Overstrike changes the editing mode to write over any text in a text field with the new text. ^O will also change the mode.

Mode: Readonly changes the editing mode to prohibit changed in text fields. ^O will also change the mode.

File: Display is not currently implemented.

File: Read reads the contents of a file into a text field. AWB prompts the user to enter the name of the input file. This command is often used to read the contents of a submission into the Submission form Text field and also to read a sequence into the Sequence TEXT EDITOR.

File: Write writes the contents of a text field to file. AWB prompts the user to enter the name of the output file.

Locate searches the current field for a string of characters entered by the user and places the cursor at the start of the string. After the command is issued, enter the character string in the box that appears.

Zoom puts the contents of a highlighted field at the top of a blank screen.

The *Block* command is not currently implemented.

Options will display a list of options for any toggle field. It is also used for linking to a locator box.

Entity: Link initiates the steps to link an entity into the field of another entity. A list of entities, locator box, or new form could follow this command.

Entity: Match searches for papers that are similar to the current paper on the screen. It provides a ranked list of papers in the database that share authors. This command is only implemented in the paper form.

Entity: Replicate creates a copy of the highlighted entity and enters it in the current field. This command works only for the Feature and Source forms.

Entity: Unlink removes the link that connects an entity to another. In a form, this command will remove an entity that is linked to a field. Be careful when using this command with features. Features that are not linked in somewhere else are not retrievable after they have been unlinked. If the command fails to unlink the feature, then the feature is linked in somewhere else to a virtual feature.

Entity: Delete removes the selected entity from the database. If an entity is unlinked, it still exists as an entity in the database. If the entity is deleted it is removed from the database completely. Certain entities, which have been linked to other entities, cannot be deleted. Use this command with caution.

Entity: Combine calls up a list of entities. After an entity is chosen, a COMBINE Command box appears in which the ID of the entity to be retained and the ID of the entity to be replaced are entered. ^X to perform the combine.

Save forces AWB to save the information in a form. AWB automatically saves when exiting a form, but this command can be used at any time.

Text Editor places the user in the vi text editor where the contents of the current, text field may be edited. This command applies only to text fields.

B.1.3 The Help Menu

The Help menu offers any available information about the current highlighted field, or displays a list of special keystrokes and what they do.

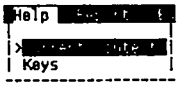

Figure B-3 The Help menu

Current Context displays a message explaining the contents of the highlighted field. This command is the same as typing ?.

Key displays a list of the command keys.

B.1.4 The Inquiry Menu

The Locate command in the Inquiry menu allows the user to find entities in the database without following or creating links.

Figure B-4 The Inquiry menu.

Locate searches the database for a particular entity. When the command is issued, a list of entities appears. The user selects from the list and then enters information in a locator box to bring up the specific entity.

Query-by-example is not currently implemented.

Tree Browser is not currently implemented.

B.1.5 The Quit Menu

The commands in the Quit menu are used to exit forms and the program.

Figure B-5 The Quit menu.

Form exits the current form. AWB automatically saves the information in the form. Notice the status bar says, "Updating database, please wait..."

Worksheet exits the worksheet and all the open forms linked to it. This command allows the user to quit several forms at the same time. The information in all the forms is saved automatically.

Program exits AWB from any stage. The information in the forms is updated before the program is terminated.

B.1.6 The Report Menu

The commands in the Report menu generate files which have information about journals scanned, a list of taxonomy nodes, or performance information.

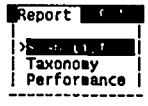

Figure B-6 The Report menu.

Scan List generates a list of recent issues for each journal that GenBank scans and prints the information to a file.

Taxonomy generates a file that contains the taxonomy tree beneath a given node. When the command is issued, a box will appear which contains the default output file name and a blank field for the taxonomy node. Enter the node, beneath which you want to see information. If you want to see the entire tree, leave the node name blank. The report will contain all nodes down to the species level.

Performance generates a report that contains statistics regarding papers entered in the last seven day period.

B.1.7 The Special Menu

The Special menu contains commands that are form specific. Refer to the Special Commands section of the particular form for an explanation of these commands.

B.1.8 The Worksheet Menu

Commands for manipulating Worksheets are in this menu.

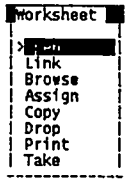

Figure B-7 The Worksheet menu.

Open calls up a list of the user's Worksheets and opens the one selected from the list.

Link puts another user's Worksheet on the current user's list.

Browse allows the user to look at the worksheets of other database users. A list of database users is presented. Select from the list and a list of their worksheets appears.

Assign changes the owner of a worksheet to from the person issuing the command to a user on the Database User List.

Copy creates a Worksheet in the user's space that has the same contents as a Worksheet belonging to another user.

Print writes the contents of a worksheet to a file.

Take allows a user to take of worksheet from a defined list of public users.

This chapter contains two major sections: AWB Commands and AWB Forms. AWB Commands contains the menu bar headings in alphabetical order, and describes the common commands with each menu item. AWB Forms contains a complete list of forms, in alphabetical order, with descriptions of the fields in each.

The commands on the Special menu are form specific. These commands will be explained in the Special Commands section with each form.

B.2 The Forms

For all selector boxes the database ID alone is enough to specify a particular entity. Other then the database ID., the fields in the selector box expect the same contents as the corresponding fields in the base window.

B.2.1 The Address Form

This form contains the address information that is linked in to the Person and Publication entities. The information in this form should pertain to an institution, not just an individual who happens to work at that institution. The phone number, for example, should be that of a department secretary or a switchboard, rather than the number of a particular person.

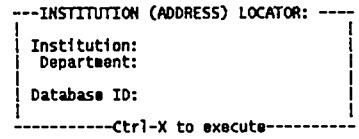

Figure B-8 The Address locator.

Institution is the name of the university or business. Use of wildcard characters is recommended.

Database ID is the unique database identifier for a particular Address entity. If the database ID is known, it alone is enough to call up a specific entity.

Department is the specific department. Do not include the words, "Department of."

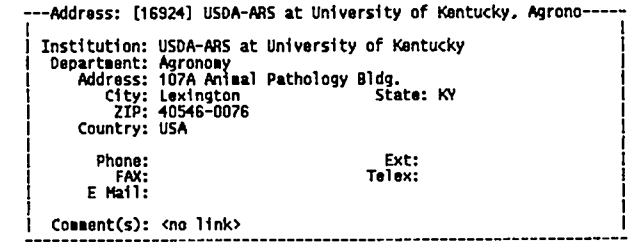

Figure B-9 The Address form.

Institution is a text field for the name of the institution.

Department is a text field for the name of the department, but not the words "Department of." For example, with "Department of Microbiology" just enter "Microbiology."

Address is a text field for the mailing address of the department.

City is a text field for the city name.

State is a text field for the name of the state or province. Ontario, PA, and Washington D.C. are examples. Use the two letter abbreviation for a state.

ZIP is a text field for the zip code or postal code.

Country is a text field for the country name (for example, U.S. A.).

Phone Number is a text field for the phone number, including area code, of the institution. This should be the number of the institution (for example, the number of the department secretary), not the number of an individual who happens to work at the institution. The phone number belonging to a person should be linked to the person form.

Ext is a text field for the extension, if available, of the department.

FAX Number is a text field. for the complete FAX number, if available, of the department.

Telex Number is a text field for the telex number, if available, of the department.

E-Mail Address is a text field. The electronic mail address of the department (if available), not just a person who works at the institution.

Comments contains the multiple links to Comments.

Special Commands

There are no Special commands for this form.

B.2.2 The Comment Form

Comments may be linked to most entities in the database. The name fields in the form and in the selector boxes refer to the name of the author of the Comment.

Figure B-10 The Comment locator.

ID is the unique identifier of the comment.

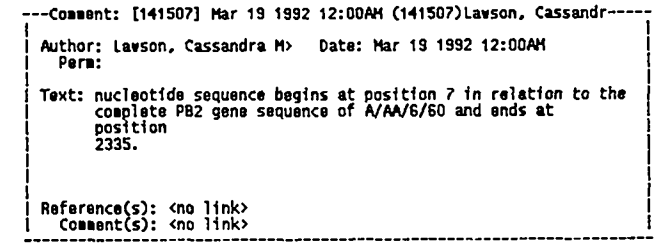

Figure B-11 The Comment form.

Author contains the single link to the author of the Comment. It expands to the Person entity.

Date is the date the Comment was created. This is put in automatically.

Perm (Permission) *Level* is an exclusive setting with two options: public and private. Public means any user will see the comment. Private comments will be seen only by database staff.

Text is a text field for the actual text of the comment. You may type the comment directly into this field.

Refs contains the multiple links to References for this comment.

Comments(s) contains the multiple links to Comments appropriate to this form.

Special Commands

There are no Special commands for this form.

B.2.3 The Document Form

This form is rarely used.

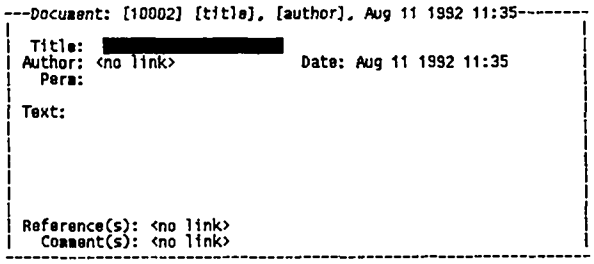

Figure B-12 The Document form.

Title is a text field that contains the title of the document.

Author is an expandable field that contains the link to the Person form of the author of the document.

Date is a text field that contains the date the document was created or last altered. This date should be entered automatically.

Perm is a text field that contains information about who can read or edit this document.

Text is a text field that contains the text of the document.

Reference(s) is an expandable field that contains the link to a Reference form.

Comment(s) is an expandable field that contains the link to a Comment form.

Special Commands

There are no special commands for this form.

B.2.4 The Database User Form

This form has fields for information about a database user.

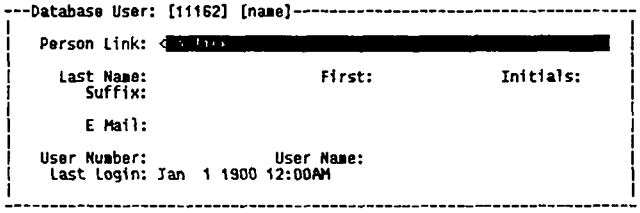

Figure B-13 The Database User form.

Person Link is an expandable field that contains the link to the Person form of the database user.

Last Name is a text field that contains the last name of the database user.

First is a text field that contains the first name of the database user.

Initials is a text field that contains the middle initial(s) of the database user.

Suffix is a text field that contains a suffix that is part of a name (e.g. Jr. or III). This field is not for titles like Ph.D.

E Mail is a text field that contains the electronic mail address of the database user. This field is primarily for remote users.

User Number is a text field that contains the system ID number in the local account.

User Name is a text field that contains the login name of the database user.

Last Login is a text field that contains the date and time of the last login. Updating of this field is not currently implemented.

Special Commands

These special commands are used only by gb-software.

Set Permissions-Super User

Set Permissions-Annotator

Set Permissions-Data Entry

B.2.5 The Entry Form

The Entry form contains information that will help database users locate a particular Sequence that is of interest to them. It also contains information used to generate the flatfile report that is sent to the corresponding author of a submission. Thus, the Entry form provides a context in which the sequence is presented.

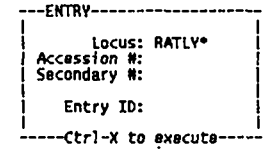

Figure B-14 The Entry locator.

Locus is the locus name of the entry.

Accession Number is the accession number of the Sequence linked to the entry.

Secondary Accession is the number of any secondary accession number linked to the entity.

Entry ID is the unique identifier for an Entry entity.

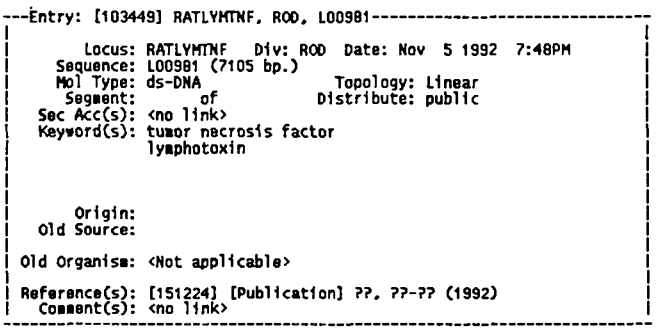

Figure B-15 The Entry form.

Locus is a text field for the locus name for this entry. The first three letters of this field are found in the Abbreviation field of a Taxonomy form linked to the sequence. They indicate the organism. The next three letters, in most cases, indicate the product. The remaining letters (up to 10 total) are for uniqueness. For information on GenBank conventions regarding locus names see "Locus Names" on page A-3. Locus names may change from release to release.

Div: (Division) A toggle field for the three letter code describing the source of the sequence. This code is found in the Div field of a Taxonomy form linked to the sequence. There are thirteen divisions.

Date is a type in field containing the date the entry was created.

Sequence contains the single link to a Sequence entity. The field may be expanded to the Sequence entity. The summary that appears in the field should indicate the number of base pairs in the sequence.

Mol Type is a toggle field for the molecule type of the sequence. This is usually ds-DNA or mRNA, though there are other, less frequent, options. ds-DNA is the default value. There are thirteen options.

Topology is a toggle field. The default value is Linear. Change this field to Circular if the sequence represents the complete circular molecule.

Segment is a text field containing the number of a segmented entry.

Distribute is an exclusive setting that is set depending upon whether the data is confidential or public.

Sec Acc(s) contains the multiple links to any secondary accession numbers. The secondary accession numbers can occur for a variety of reasons; here is one example. A segmented entry is later joined to make one sequence. Each of the accession numbers of the original segments will become secondary accession numbers for the new sequence. Those numbers should then be linked to this field as secondary accession numbers of the new

Keyword(s) contains the multiple links to keywords. Product names are usually linked in as keywords. `\eel` will bring up the locator box for keywords. You may use wildcards to find keywords. For example, if the product is "aldo-keto reductase" you may type ald*ket* and the locator will search for keywords that begin with "ald" and contain "ket" in the name. If the locator finds the pattern it will produce a list from which to choose. (If you call up the Keyword form, be very careful; changes made to a keyword will affect the appearance of that keyword everywhere in the database.)

Origin is a text field for information about restriction sites or the position of sequences relative to one another. Two examples are, "100 bp upstream of NotI site" or "1.5kb downstream of segment 1." The location referred to is the first base of the sequence.

Old Source is no longer used.

Refs contains the multiple links to References for this entity. A paper reporting results of secondary sequence analysis (secondary reference), rather than initial reporting of sequence data, should be linked to this field in the Entry form. Papers containing the initial reports of a sequence (primary reference) should be linked to the Refs field in both the Entry and Sequence forms.

Comment contains the multiple links to Comments appropriate for this entry. This is the only Comment field that will appear in the flatfile report.

Special Commands

Make Flatfile will creates a flatfile report. The report is displayed to the user.

B.2.6 The Feature Form

The Feature form contains information that provides a functional context for a span of the sequence. "The DDBJ/EMBL/GenBank Feature Table: Definition" is the authoritative document regarding features.

```
---FEATOCC----------
|                  |
| Database ID: 321753 |
|                  |
-----Ctrl-X to execute----
```

Figure B-16 The Feature locator.

Database ID is the unique identifier of the Feature entity.

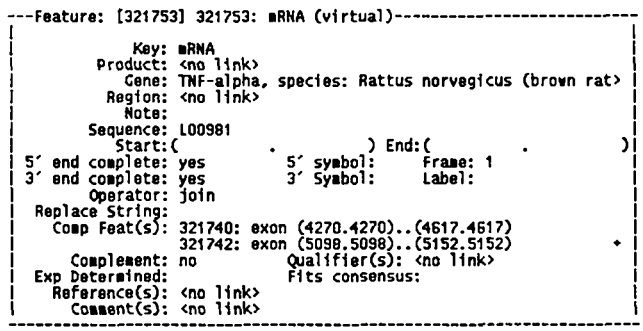

Figure B-17 The Feature form.

Key contains a single link to a name describing the function of this section of the sequence. CDS, exon, 5' UTR are examples.

Product contains the single link to the Product associated with the feature.

Gene contains a single link to the gene associated with the feature. This field expands to the Gene Occurrence form. The Gene Occurrence form has a Gene field which expands the Gene form. The same gene may be linked to multiple gene occurrences.

Region contains the single link to the Region of the genome on which this feature is found.

Note is a text field that contains any descriptive information about the feature that cannot be linked as a different qualifier. The contents of this field will appear in the flatfile report as a /note qualifier. In other words, this field is for the first of the /note Qualifiers. Other notes may be linked to the Feature in the Qualifier field.

Sequence: This field contains the accession number of the Sequence that is associated with this Feature.

Start and *End* are text fields that contain the numbers of the base pairs in the sequence that begin and end the feature. There are two fields for each to allow for fuzzy ends, i.e. a range of numbers that is known to include the start of the feature and/or a range for the end of the feature. Leaving the first number of the Start field blank means that the start of the feature is known to occur before the second number in the start field. In the same way, if the second part of the End field is left blank, it means the end of the feature occurs at some number greater than the first value in the end field. For example, Start:(. 1) End:(350 .) means the start of the feature occurs before 1 and the end occurs after 350.

5' end complete: Is the 5' end complete? Toggle this field to yes or no.

3' end complete: Is the 3' end complete? Toggle this field to yes or no.

5' symbol is a toggle field. It is not in use and should not be edited.

3' symbol is a toggle field. It is not in use and should not be edited.

Frame can be set to 1,2, or 3. The value defines the first base in a codon. If the frame is 1, the first base in the feature is the start of the codon. If the frame is 2, the second base of the feature is the start of the codon. Use the Translation button at the bottom of the form to check the amino acid translation of the feature.

Label is not often used.

Operator is a toggle field for the operator that connects the components of a virtual feature. For example, the features could be "joined," meaning end to end, or the operator could be "order," meaning the order of the features is given, but there may be areas between the features that are not known.

Replace String is used when the Replace operator is chosen. The replace strings are the letters of the of the bases that will replace the span identified in the Start and End fields. The replacement may involve the insertion or deletion of bases. For more information on replace strings, see "Replace Strings" on page A-5.

Comp Feat(s) contains the multiple links to the components of a virtual feature. Expanding on any of the components will bring up the Feature form of that component.

Complement is a toggle field indicating whether the feature pertains to the sequence as presented or its complement. Yes means it pertains to its complement.

Qualifier(s) contains the multiple links to the Feature Qualifiers.

Exp Determined: Was the feature experimentally determined? Toggle this field to yes or no.

Fits Consensus: Does this feature fit the accepted consensus for such features? For example, does an intron satisfy Chambon's Rule? Toggle this field to yes or no.

Comments contains the multiple links to Comments for the Feature.

References contains the multiple links to any References for this Feature.

Special Commands

Locate span searches the span of the feature for a given pattern of bases. If, for example, you issue this command input "atg", AWB will create a list of the places in the span where the pattern "atg" occurs. If the string is unique, AWB will enter the number of the first base pair (5' end) in the Start field.

Amino Acid Translation produces an amino acid translation of the span. Choose either the 1 or 3 letter code.

Check feature runs defined routines to check the feature for errors.

Show Sequence calls up the span defined by the Start and End fields.

B.2.7 The Feature Key Form

This form contains the official name of a feature key. The Feature key is linked to the first field of the Feature form. Currently, 3/15/93, there are 71 feature keys.

```
---FEATURE KEY LOCATER----------------------
|                                           |
| Feature Key: CDS                          |
|                                           |
-------------Ctrl-X to execute--------------
```

Figure B-18 The Feature Key locator.

Feature Key is one of the list GenBank feature names. Examples are CDS, intron, and TATA_signal.

```
---Feature Key: [10210] CDS-----------------------------------------
|                                                                   |
|         Name: CDS                                                 |
|  GenBank Key: pept                                                |
|     EMBL Key: CDS                                                 |
|                                                                   |
|   Definition: Coding sequence; sequence of nucleotides that <or>  |
|                                                                   |
| Reference(s): <no link>                                           |
|   Comment(s): <no link>                                           |
---------------------------------------------------------------------
```

Figure B-19 The Feature Key form.

Name is the current GenBank name for the feature.

GenBank Key is the old-style GenBank Feature Key name.

EMBL Key is the EMBL version of the name.

Definition is an explanation of the Feature Key.

Comments contains the multiple links to Comments appropriate to this Feature Key.

Refs contains the multiple links to References for to this Feature Key.

B.2.8 The Feature Qualifier Form

Feature Qualifiers are linked to the Qualifier field of the Feature form.

```
---FEATQUAL LOCATER-----------------
|                                  |
| Qualifier: EC*                   |
|                                  |
----------Ctrl-X to execute---------
```

Figure B-20 The Featqual locator.

Qualifier is one of the GenBank feature qualifier names. Examples are anticodon, codon_start, EC_number, and rpt_family.

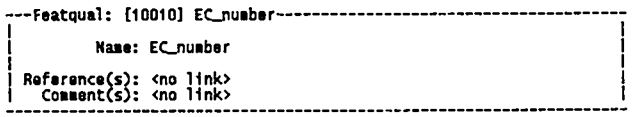

Figure B-21 The Featqual form.

Name is a text field for the name of the qualifier.

Refs contains any References linked to this Feature Qualifier.

Comments contains any Comments linked tot this Feature Qualifier.

B.2.9 The Gencode Form

The Gencode Exception form contains general translation exceptions that are applied to a given organism. It is linked to the Taxonomy form.

Note: Any exception entered in this form will apply to all taxonomy nodes beneath the one to which the Gencode Exception entity is linked. Specific translation exceptions are entered in the Feature form (of a cds) in the Qualifier field. The Featqual is transl_except.

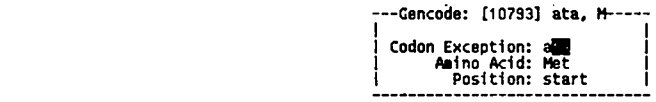

Figure B-22 The Gencode form.

Codon Exception is a text field for the letters of the codon exception.

Amino Acid is a toggle field for the amino acid for which the codon translates.

Position is a toggle field for the position of the codon exception.

Special Commands

There are no Special commands for this form.

B.2.10 The Gene Form

Gene entities are linked to Gene Occurrences which is in turn linked to a Feature.

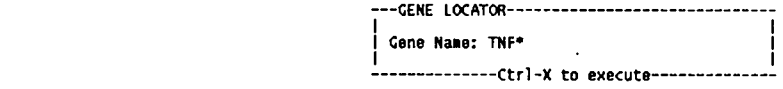

Figure B-23 The Gene locator.

Gene Name is the text field for the official name of the gene; TNF- alpha, lacZ and fliD are examples.

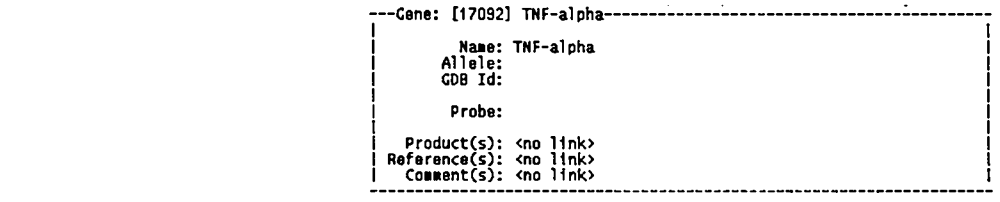

Figure B-24 The Gene form.

Name is a text field for the official name of the gene; lacZ and fliD are examples.

Allele is the GDB name of the allele, if known.

GDB ID is the GDB ID of the gene, if known.

Probe is a text field for the names of probes used to isolate the gene.

Product(s) contains the multiple links to the Products encoded by this gene.

Refs contains the multiple links to References for this gene.

Comments contains the multiple links to Comments for this gene.

Special Commands

There are no Special commands for this form.

B.2.11 The Gene Occurrence Form

Gene Occurrence entities are linked to features. The link in the Gene field of the Feature form may be followed to a Gene Occurrence form.

```
---GENE OCCURRENCE LOCATOR-------------------------------------
|                                                             |
| Gene Name: TNF*                                             |
| Organism Scientific Name: Ratt*                             |
|                                                             |
----------------------Ctrl-X to execute-----------------------
```

Figure B-25 The Gene Occurrence locator.

Gene Name is the GenBank name for the gene.

Organism Scientific Name is the scientific name of the organism. Some examples are Homo sapiens (genus and species), Scotobacteria (division), and Prokaryota (kingdom).

```
---Gene Occurrence: [17046] TNF-alpha, species: Rattus norvegicus-----
|                                                                    |
|             Gene: TNF-alpha                                        |
| Number of Exons:                                                   |
|    Map Position:                                                   |
|       Sequenced:                                                   |
|      Product(s): tumor necrosis factor alpha                       |
|       Region(s): <no link>                                         |
|        Taxonomy: species: Rattus norvegicus (brown rat; common ra> |
|                                                                    |
|    Reference(s): <no link>                                         |
|      Comment(s): <no link>                                         |
---------------------------------------------------------------------
```

Figure B-26 The Gene Occurrence form.

Gene contains the single link to the Gene. If the gene is identified only by a product, this field should be left empty. This field expands to the Gene form.

Taxonomy contains the single link to the species Taxonomy node of the organism containing this gene.

Map Position is the map location of the gene; 12q24 is an example.

Number of Exons is a text field for the number of exons in the complete gene (whether or not all of them appear in the presented sequence).

Sequenced is a toggle field. Is there a sequence in the database for any portion of this gene? The default setting is yes.

Products contains the multiple links to the Products of the gene.

Region contains the multiple links to the Regions of the genome on which this gene is found.

Refs is contains any References linked to this Gene Occurrence. This field is not often used.

Comments contains any Comments linked to this Gene Occurrence. This field is not often used.

B.2.12 The Keyword Form

Keywords are linked to Entry entities.

Note: Changes made to a keyword in this form affect the appearance of that word everywhere it is used. In other words, a single keyword may be linked to hundreds of entities, and any change to that word appears in all of the entities to which it is linked.

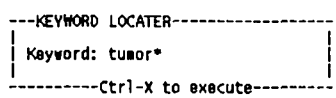

Figure B-27 The Keyword locator.

Keyword Name is the name of the keyword as it appears on the keyword list. It is particularly advantageous to use wildcards here to minimize typing errors or capitalization differences.

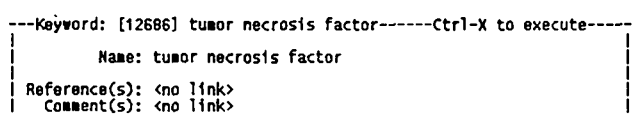

Figure B-28 The Keyword form.

Name contains the keyword. Changing the keyword in this field will change its name throughout the database.

Refs contains any References linked to this Keyword. This field is not often used.

Comments contains any Comments linked to this Keyword. This field is not often used.

Special Commands

There are no Special commands with this form.

B.2.13 The Paper Form

A journal article containing a sequence is the typical source for information that goes into the database. The Paper form contains the reference information surrounding a journal article or other published document. Unlike the Reference form, the Paper form contains a link to the Entry form.

```
---PAPER LOCATOR---------------------
|                                    |
| Database ID:                       |
|                                    |
| Publication: <no link>             |
|      Volume:                       |
|  Start page:                       |
| Submission ID:                     |
|      Author: Benveniste, Etty N    |
|                                    |
----------Ctrl-X to execute----------
```

Figure B-29 The Paper locator.

Database ID is the unique identifier for the paper.

Volume and *Start Page* are journal volume and beginning page of the article.

Publication is the abbreviation of the publication as it appears in the database. This field is the same as the *Abbreviation* field in the Publication sub-selector.

Submission ID is the unique identifier for the particular submission entity.

Author is for the single link to an author of the paper.

```
---Paper: [153708] Planta ??, ??-?? (1993)----------------------
|                                                                |
| Publication: Planta                              Year: 1993    |
|      Volume:        Issue:            Pages:       to         |
|       Title: Cloning and developmental expression of           |
|              sucrose phosphate synthase from spinach           |
|                                                                |
|   Author(s): Klein, Robert R.                                  |
|              Crafts-Brandner, Steven J.                        |
|              Salvucci, Michael E.                              |
|                                                                |
|                                                                |
|  Pub Status: In Preparation      Contains Sequence: yes        |
|     Status: DATA distributed, [person], Oct 26 1992  7:3+>     |
|   Hold date: Jan  1 1900 12:00AM                               |
|    Pub Here: no      Cit Address:                              |
| Submission: Klein, Robert Oct 14 1992 10:19AM Authorin E->     |
|   Entry(s): SPISPS1A, PLN, L04803                              |
|                                                                |
|                                                                |
|  Comment(s): Oct 12 1992 12:00AM (171178)Klein, Robert R.      |
----------------------------------------------------------------
```

Figure B-30 The Paper form.

Publication contains the single link to the Publication. The Publication abbreviation appears as the summary in the field.

Volume, Issue, Pages, and Year are text fields that contain the specific reference information for this paper.

Title is a text field for the title of the paper. The title should be in lower case except for the first letter and any letters that, by convention, are always capitalized (e.g., DNA, fliS, Thermus acquaticus). The first letter after a colon is capitalized, and there is no period at the end of the title. For more information about title conventions, see "Titles" on page A-6.

Authors contains the multiple links to the Person entities of the authors. You may expand any name in this field to see the Person form .

Pub Status is a toggle field that contains the publication status of the paper (i.e. published, unpublished, in press, etc.). `\eo` will call up a list of options.

Status contains the multiple links to the reference statuses applied to GenBank submissions. `\ss` used to set a status. > on the right side of the field indicates there is more information to the right. < indicates more information to the left. Use the arrow keys to move through the statuses.

Hold date is a text field for the date until which a sequence is to be held confidential. There must be a date for all HUP (Hold-Until-Published) papers. Information in the paper form cannot be distributed prior to this expiration date.

Pub Here is a toggle field. Is the published paper in the GenBank library. This field is for GenBank staff.

Submission contains the single link to the Submission entity.

Entry(s) contains the multiple links to the Entry entities for this paper. Each link may be followed to an Entry entity.

Comments contains the links to any Comments for this Paper entity.

Special Commands

Set status calls up a list of statuses which are entered in the Status field. Selecting from the list sets a reference status.

Distribute will set the data to be distributed that evening. When this command is issued, a box appears that contains three toggle field. Set the Distribute field to distribute, public, or private. Set the Annot Quality field to simple, full, unannotated, or leave it blank. Set the Annot Quality field to staff_entry, staff_review, automatic, or leave it blank. If the last two fields are left blank, the values set (for these same fields) in the Sequence form are used.

Entry template copies most of the contents of a highlighted entry to the other entries in the Entry field. It does not copy the locus names of the sequence. Be careful to highlight the item. Do not use this command with Authorin submissions.

Acknowledge: email prepares an E-mail message to be sent to the corresponding author which reports the accession numbers of the sequence(s) submitted and explains update procedures. If the submission is HUP, more detail about HUP updates is included.

Acknowledge: Postal mail prepares a letter to faxed or mailed to the corresponding author which reports the accession numbers of the sequence(s) submitted and explains update procedures. If the submission is HUP, more detail about HUP updates is included.

Respond to author: email prepares a letter to be sent E-mail to the author. The letter will include the flatfile report. This command is usually used by GenBank annotators to respond to the author after a submission has been reviewed.

Respond to author: Postal mail prepares a letter to be sent through the postal service. The letter will include the flatfile report.

Link new entries is the command to create multiple entry forms at one time. When this command is issued, AWB asks, "How many?" The user must enter the number of sequences to be entered. A sequence entity will be created for each sequence, as well as and Entry entity.

Make flatfiles creates a flatfile report.

B.2.14 The Person Form

The Person form is form information about people in the database. Person entities may be authors, GenBank staff, journal editors, or any other people associated with GenBank or GenBank submissions.

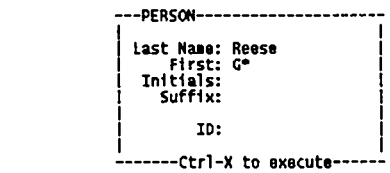

Figure B-31 The Person locator.

ID is the unique identifier for the Person.

The *First Name*, *Last Name*, and *Initials* fields are for the names and middle initials of the person. Include a period after each initial.

Entry Last Name is the last name of the person.

First is the first name of the person.

Initials are the middle initial(s). Include a period after each initial.

Suffix contains the text of any suffix for the name, like Jr. or III. This should not be a title, like Ph.D.

Sp. Address is a text field for a building number, mail stop, P.O. Box, or other specific, personal address information. The information in this field, along with the information in the Address entity, forms the complete address of the person.

Institution contains the single link to the address entity of the institution associated with this person.

E Mail is a text field for the complete E-mail address of the person.

Work Phone is a text field that for the complete work phone number.

FAX is a text field for the FAX number.

Home Phone is a text field for the complete home phone number.

Database login is the AWB login name of the person.

Perm. Level is the type of permissions this person has.

Corr. Author is an exclusive setting. Set this field to Yes if this is the person to whom correspondence about a submission should be sent.

Comments contains any Comments linked to this Person entity.

Special Commands

There are no Special commands for this form.

B.2.15 The Product Form

Products are linked to features and to genes. Though Product names are often used as keywords, Product entities are never linked into the Keyword field.

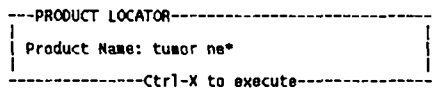

Figure B-32 The Product locator.

Product Name is the name of the product as it appears in the database. It is a good idea to use wildcards in this field.

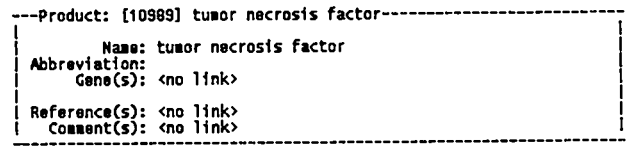

Figure B-33 The Product form.

Name is a text field for the name of the product.

Abbreviation is a text field for the three letter abbreviation for the product to be used in the locus name.

Gene(s) contains the multiple links to the Gene Occurrence entities.

Refs contains any references linked to this product. This field is rarely used.

Comments contains any comments linked to this product. This field is rarely used.

Special Commands

There are no Special commands for this form.

B.2.16 The Publication Form

The Publication form is for information about journals and other publications that appear in the database.

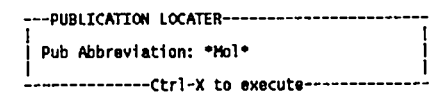

Figure B-34 The Publication locator.

Publication Abbreviation is the abbreviation for the publication as it appears in the database.

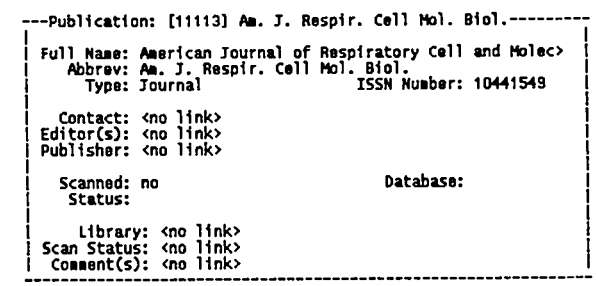

Figure B-35 The Publication form.

Full Name is a text field for the complete, official name of the publication, for example, "Journal of Clinical Biology."

Abbrev contains the official NLM (National Library of Medicine) abbreviation for the publication, for example, "J. Clin. Microbiol." This is the abbreviation that appears in the Publication field of the Paper (or Reference) form.

ISSN Number is a text field for the International Standard Serial Number of a journal.

Type is a toggle field for the type of publication (e.g., book or journal). The options are Book, Journal, Proceedings, Submission to DDBJ, Submission to EMBL, Submission to GenBank, Thesis, and Unpublished. Highlight and release to choose an option.

Contact contains the single link to the person to call in case of questions about GenBank interactions with this particular journal.

Editor(s) contains the multiple links to the names of the editors of this particular journal. Each item in this field is a link to a Person entity.

Publisher contains the single link to the Address entity of the publisher.

Scanned is a toggle field. Is the publication regularly scanned by the staff of any of the databases?

Database is a toggle field. If the publication is regularly scanned by the staff of one of the databases, which one scans it? There are for options for this field: EMBL, DDBJ, GenBank, or the field may be left blank.

Status is a toggle field.

Library contains a singel link to the library where the publication can be found.

Comments contains the links to any Comments for this Publication.

Special Commands

There are no Special commands for this form.

B.2.17 The Qualval Form

The Qualval form displays the Feature Qualifier information.

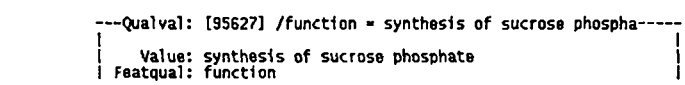

Figure B-36 The Qualval form.

Value is a text field that contains specific information that belongs with the Feature Qualifier linked in below it. For example, if the Qualifier EC_number is linked in to the Featqual field, the specific Enzyme Commission number is typed into the Value field.

Featqual contains the single link to name if the qualifier. `\eel` calls up a list of feature qualifiers. `^X` will expand the field to the Featqual form.

Special Commands

There are no Special commands for this form.

B.2.18 The Reference Form

The Reference form has the same fields as the Paper form with one exception. There is no Entry field in the Reference form. A Reference form and a Paper form which display the same ID number.

```
---REFERENCE-------------------------
|                                    |
| Database ID:                       |
|                                    |
| Publication: J. Mol. Biol.         |
|      Volume: 4*                    |
|  Start page:                       |
|                                    |
----------Ctrl-X to execute----------
```

Figure B-37 The Reference locator.

Database ID is the unique identifier for the Reference entity.

Publication is a single link field for the abbreviation of the publication.

Volume is the specific volume of a journal.

Start Page is the number of the first page of the article.

```
---Reference: [10378] [10378] J. Mol. Biol. 47, 15-28 (197-----
|                                                              |
| Publication: J. Mol. Biol.                      Year: 1970   |
|      Volume: 47      Issue:          Pages: 15    to 28      |
|       Title: Mutant tyrosine transfer ribonucleic acids      |
|                                                              |
|                                                              |
|   Author(s): Abelson, John N.                                |
|              Gefter, M. L.                                   |
|              Barnett, Louise D.                              |
|              Landy, A.                                       |
|              Russell, R. L.                                  +
|  Pub Status: Published              Contains Sequence:       |
|      Status: DATA released to public, Staff, GenBank , Au+>  |
|   Hold date: Jan  1 1900 12:00AM                             |
|    Pub Here:          Cit Address:                           |
| Submission: <no link>                                        |
| Comment(s): <no link>                                        |
--------------------------------------------------------------
```

Figure B-38 The Reference form.

Publication contains the single link to the publication. The link can be followed to the Publication form.

Year, *Volume*, *Issue*, and *Pages* are text fields for the specific reference information for this paper.

Title is a text field for the title of the paper. The title should be in lower case except for the first letter and any letters that, by convention, are always capitalized.

Authors contains the multiple links to the authors. Each name on this list represents a link to a Person entity.

Pub Status is an exclusive setting for the publication status of the paper.

Contains Sequence is a toggle field. If the paper contains a sequence, toggle the field to Yes.

Status contains the multiple links to the reference statuses applied to GenBank submissions.

Hold date is a text field for the date until which a sequence is to be held confidential. There must be a date for all HUP (Hold-Until-Published) papers. Information in the paper form cannot be distributed prior to this expiration date.

Pub Here is toggled to yes if the journal is part of the GenBank library.

Submissions contains the multiple links to the Submission entities.

Comments contains the links to any comments for this paper.

B.2.19 The Reference Status Form

The Reference Status form contains information about the statuses applied to Reference (or Paper) entities. It contains the single link to the person applying the status.

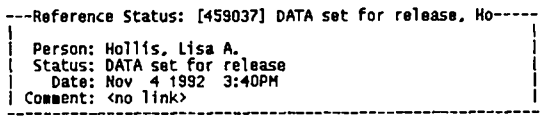

Figure B-39 The Reference Status form.

Person contains the single link to the Person who set the status. The link may be followed to a Person entity.

Date is a text field for the date and time the status was set. This field will be filled in automatically by AWB.

Status is a toggle field. \eo will call up the list of options below.

Comment contains the multiple links to Comments for the Reference Status form. Comments linked here are a useful record of special remarks sent to an author, or of updates and other alterations made to a submission.

Special Commands

There are no Special commands for this form.

B.2.20 The Region Form

This form is not currently in use.

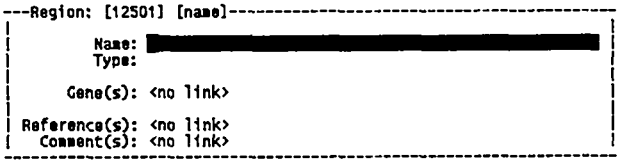

Figure B-40 The Region form.

Name is a text field that contains the name of the region.

Type is a text field that contains the type of region. Operon and chromosome are examples.

Gene(s) is an expandable field that contains related gene(s). This field expands to the Gene Occurrence form.

Reference(s) is an expandable field that contains any references linked to this region.

Comment(s) is an expandable field that contains any comments linked to this region.

Special Commands

There are no special commands for this form.

B.2.21 The Rsite Form

The Rsite form is not currently in use.

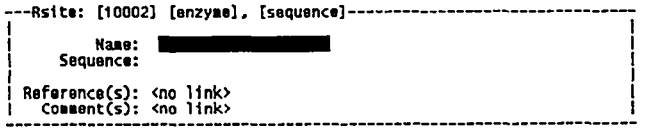

Figure B-41 The Rsite form.

Name is a text field that contains the restrictive enzyme name.

Sequence is a text field that contains the restriction enzyme recognition sequence.

Reference(s) is an expandable field that contains any references linked to this enzyme.

Comment(s) is an expandable field that contains any comments linked to this enzyme.

B.2.22 The Secondary Accession Number Form

Secondary accession numbers are linked to Entry entities.

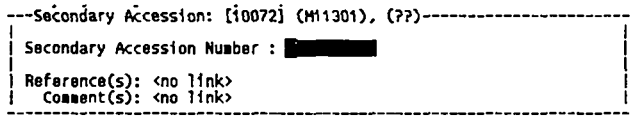

Figure B-42 The Secondary Accession form.

Secondary Accession Number is a text field that for the accession number that is considered secondary.

Comments contains any Comments linked to this secondary accession number.

Refs contains any References linked to this secondary accession number.

B.2.23 The Sequence Form

Sequence entities are linked to Entry entities. The Sequence field has the link to the Sequence Editor, where the base pairs of the sequence are displayed. The Sequence form also displays the links to the Source and Feature entities.

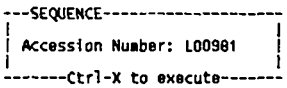

Figure B-43 The Sequence locator.

Accession Number is the unique database ID for a Sequence entity.

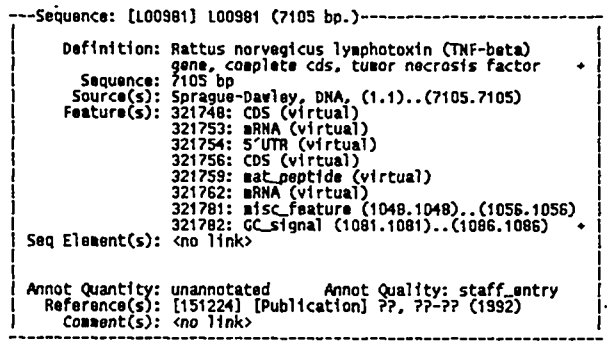

Figure B-44 The Sequence form.

Definition is a text field for the sequence definition. The definition should contain genus, species, product, gene name, molecule type and identifier which tells what portion of the gene is represented by the sequence. By convention, only the first word of the definition is capitalized; the gene name is put in parentheses (if there is a product); there is a comma after the molecule type and a period at the end of the definition. For example, "Salmonella typhimurium flagellar protein (fliS) gene, 3' end." For more information about sequence definitions see "Sequence Definitions" on page A-5.

Sequence has the link to the Sequence editor where the sequence is display. The number of base pairs appears in the field.

Source contains the multiple links to the biological sources of the sequence. Each link may be followed to a Source entity. Each Source will have a span associated with it.

Features contains the multiple links to the Features. For more information about features, refer to "The DDBJ/EMBL/GenBank Feature Table."

Seq Element(s) contains the multiple links to the entities that specify how the components of a virtual sequence are assembled.

Annot Quantity is a toggle field. The options are unannotated, simple, and full.

Annot Quality is a toggle field. The options are automatic, staff_entry, and staff_review. New annotators set this field to staff_review.

Comments contains any Comments linked to this Sequence.

Refs contains any References linked to this Sequence.

Special Commands

Check sequence runs consistency checks on all the listed features. For example, a cds is checked to be sure that it translates.

Find vector checks the sequence against a list of known vectors and informs the user if there is a match.

B.2.24 TheSequence Element Form

The Sequence Element form contains the spans of a virtual sequence with both the span numbers of the virtual sequence and the span numbers of the original sequences that are components of the virtual.

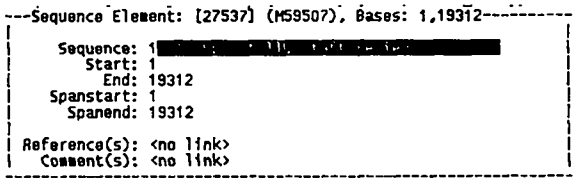

Figure B-45 The Sequence Element form.

Sequence contains the single link to the component Sequence.

Start is a text field for the number of the initial base pair of the span on the component Sequence that will be used.

End is a text field for the number of the final base pair on span of the component Sequence.

Suppose base pairs 30 to 150 of the component sequence will be used, then Start is 30 and End is 150. Also suppose these 90 base pairs will make up bases 110 to 200 on the virtual sequence. Then Spanstart is 110 and Span end is 200.

Spanstart is a text field for the start of the span on the virtual sequence that comes from this component.

Spanend is a text field for the end of the span on the virtual sequence that comes from this component.

B.2.25 The Source Form

The information in the Source form provides the biological context for the sequence. The Source form also displays the links to the Taxonomy nodes of specific and laboratory host organisms.

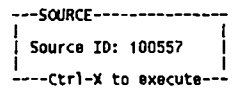

Figure B-46 The Source locator.

ID is the unique identifier for the Source entity.

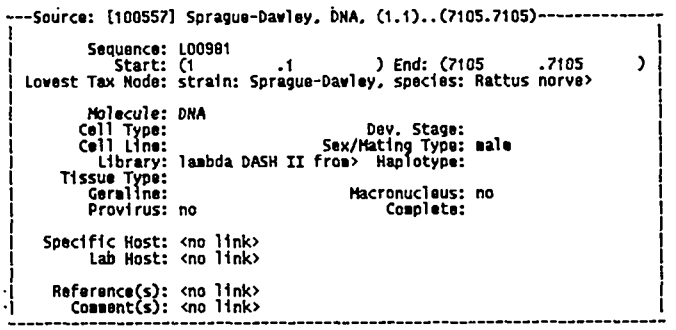

Figure B-47 The Source form.

Start contains two text fields. The first is the number initial base in the range for the beginning of the source span. The second is the number of the last bast in the range for the start of the source span. If the 5' end is a single base, these numbers should be the same.

End contains two text fields. The first is the number of the initial base in the range for the end of the source span. The second is the number of the last base in the range for the end of the source span. If the 3' end is a single base, these numbers should be the same.

The Start and End fields are filled in automatically when the source is linked to the sequence. The values will need to be changed if there is more than one source (i.e. for a synthetic recombinant gene). For example, the first 600 base pairs may come from one source and the next 1500 from another. Both sources are linked to the Source field of the Sequence entity and the Start and End field numbers are adjusted accordingly.

Lowest Tax Node contains the single link to the most specific branch of the taxonomy tree that pertains to the sequence. This could be, for example, species, strain, or chromosome. The link may be followed to the Taxonomy entity.

Cell Line is a text field for the cell line. Examples are HeLa, MKC, and HEP.

Cell Type is a text field for the cell type. Examples are leukocyte and hepatocyte.

Dev. Stage is a text field for the development stage. Examples are fetal, vegetative, and adult.

Tissue Type is a text field for the tissue type. Liver, brain, and blood are examples.

Library is a text field for the tissue or cell library. Examples are T. Maniatis, EMBL3, ATCC1234, and other culture collection libraries.

Haplotype is a text field for the haplotype. Examples are DQ and HA-DQ5.

Sex/Mating Type is a text field. Examples are male, female, and hermaphrodite.

Macronucleus is an exclusive setting. Is the sequence macronuclear?

Provirus is an exclusive setting. Is the sequence proviral?

Germline is an exclusive setting. Is the source germline?

Complete is an exclusive setting. Does the sequence represent the complete genome?

Specific Host contains the multiple links to the lowest Taxonomy node of the particular host from which the organism was isolated. Link in a specific host only if it is different from the entity linked to the Natural Host field in the Taxonomy form.

Lab Host contains the multiple links to the lowest Taxonomy nodes of laboratory host organism. This field will be empty unless the laboratory host is different from the natural host. (The Natural Host field is in the Taxonomy entity.)

Comments contains any References linked to this Source.

Refs contains any References linked to this Source.

Special Commands

Show sequence calls up the Sequence editor. The span entered in the Start and End fields will appear in the window.

Locate span finds a pattern of base pairs and prints a list of places in the sequence where that pattern occurs.

B.2.26 The Submission Form

The Submission form contains the text of the original submission. It also contains fields with the name of the author making the submission and the medium by which it arrived at GenBank.

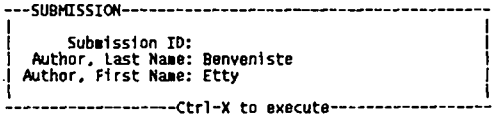

Figure B-48 The Submission locator.

ID is the unique identifier for the Submission entity.

Author First Name is the first name of the author.

Author Last Name is the last name of the author.

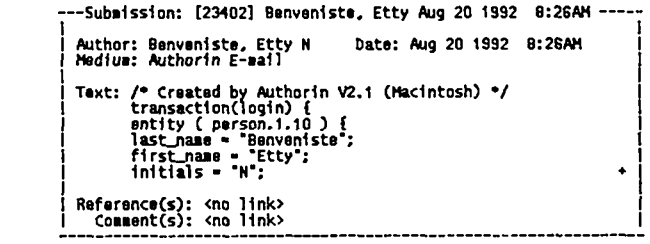

Figure B-49 The Submission form.

Author contains the single link to the Person entity of the author making the submission.

Date is a text field for the date and time the Submission entity was created.

Medium ia a menu button. What was the medium by which the submission was made? Right click and hold to display the list of options.

Text is a text field for the contents of the submission.

Comments contains any Comments linked to this Submission.

Refs contains any References linked to this particular Submission.

B.2.27 The Tax Level Form

The Tax Level is the position on the Taxonomy tree of a particular Taxonomy node. Examples of Tax Levels are kingdom, species, strain, cultivar, and individual_isolate.

There is no Tax Level locator. Instead, there is a list of Tax Levels.

Figure B-50 The Tax Level form.

Level Name is a text field for the level of the Taxonomy node.

Parent contains the single link to the Tax Level entity just above this one on the taxonomy tree (i.e., one step closer to the root).

Child contains the single link to the Tax Level entity just below this one on the taxonomy tree (i.e., one step away from the root).

B.2.28 The Taxonomy Form

Taxonomy nodes are linked to the Source entities. They are also linked to other Taxonomy nodes.

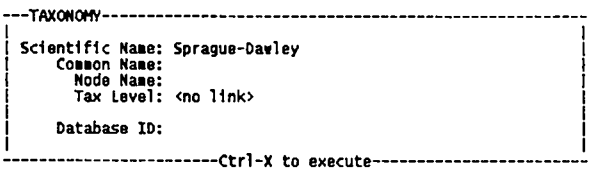

Figure B-51 The Taxonomy locator.

Scientific Name is the scientific name of the organism. Examples are Homo sapiens, Scotobacteria, and Prokaryota.

Common Name is the common name for the organism. Examples are cow, mouse, and human.

Node Name is the name of the Taxonomy node.

Tax Level is the name of the level. Genus, order, class, and kingdom are examples. This field is the same as the *Level Name* in the Tax Level sub-selector.

Database ID is the unique identifier for the Taxonomy entity.

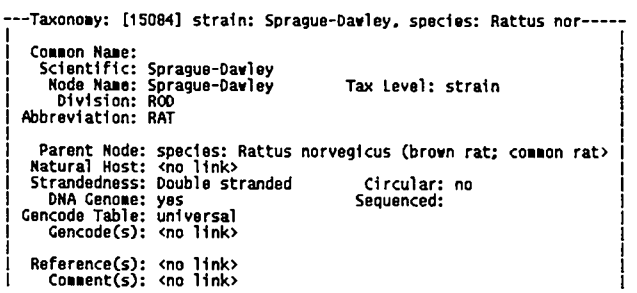

Figure B-52 The Taxonomy form.

Scientific Name is a text field for the scientific name of the organism. Some examples are Homo sapiens (genus and species), Scotobacteria (division), and Prokaryota (kingdom).

Common Name is a text field for the common name for the organism (e.g., man, cow, mouse).

Node Name is a text field for the node name of the current form.

Tax Level contains a single link to the level on the Taxonomy tree of this form (e.g., genus, species, family)

Division is a toggle field for the three letter division abbreviation.

Abbreviation is the three letter GenBank code for this organism. If this field is blank, the abbreviation can be found by following the Parent Node up the Taxonomy tree.

Parent Node contains a single link to the node of the taxonomy tree that is on step closer to the root. Follow this link to another Taxonomy entity.

Natural Host contains a single link to the Taxonomy entity of the organism that normally hosts the current organism.

DNA Genome is an exclusive setting.

Circular is an toggle field. Choose yes or no.

Strandedness is an toggle field for the strand type.

Sequenced is an toggle field.

Gencode Table is an toggle field for the coding scheme of the organism. Settings here affect items beneath this node on the Taxonomy tree (further from the root).

Gencodes contains the multiple links to general Gencode Exceptions for this organism and all organisms beneath it on the Taxonomy tree. Be careful when entering a new gencode and be aware that the gencode exception will propagate down the taxonomy tree.

Refs contains the multiple links to References for this entity.

Comments contains the multiple links to Comments for this entity.

Special Commands

There are no Special commands for this form.

B.2.29 The Worksheet

The Worksheet is often the starting place for an AWB session. It provides a space on which to link other entities of various kinds.

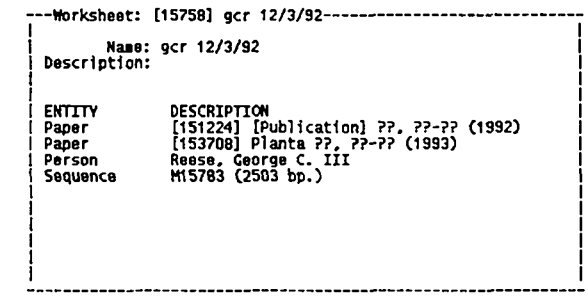

Figure B-53 The Worksheet.

Name is a text field. You may edit the name of the worksheet. Whenever you are in Insert or Overstrike mode. (Change mode with ^O.)

Description is a text field. You may add a description of the contents of the worksheet. Use of this field is optional.

Entity contains the multiple links to various entities. Usually annotators link Papers to the Worksheet. However, any entity from the Entity List may be linked to this field. It may at times be necessary or convenient to link in a Sequence, Entry, Person or other entity.

Special Commands

Set Status allows the user to set a reference status at the Worksheet level. The status will be given to all the papers of references in the Entity field.

Appendix C Utilities: Reference

This appendix describes useful programs that are run from the command line (outside AWB). Most of the programs are used to quickly call up information from the database.

The programs are listed in alphabetical order.

All programs that ask for a password will accept the AWB password.

Off-site users: None of the programs in the section involve entering or annotating sequences with AWB.

GenBank staff: The programs in this appendix are used by annotation and dataflow staff to accomplish a variety of tasks involving submissions to GenBank.

C.1 Auinsub

This program is used by dataflow staff with Authorin submissions.

C.1.1 Summary

The Auinsub program enters the information from an Authorin transaction into the database. It then reports the reference ID and the accession numbers of the sequences

`auinsub` *`filename`* `hup`

or

`auinsub` *`filename`* `pub`

hup or pub specify the Hold-Until-Published or public mode.

```
%auinsub mess0820.aa pub
Locking up database NOW!
Won't you be popular!!
Databse unlocked. What a relief!

>>
>> Legal transaction
>>

>>
>> Legal transaction
>>
This authorin transaction has a hold date that is later
than today and you are running it as published. If you
continue you could potenitally release confidential data.
You have three choices:
1) continue
2) exit
3) change to Hup
Please typ c for continue, e for exit, or h for change to Hup
h
couldn't locate person with uid 376
WARNING: publication  Exp. Parasitol. not found in database
This reference will be added with out a publication!!
Rf_id 10720 Accession # L01004
>>
>> Transaction processed successfully
>>
%
```

Figure C-1 The output of the Auinsub program.

C.1.2 Description

Authorin submissions are entered into the database with the Auinsub program. The program enters the information into the database and creates two new files: filename.accnumbs and filename.rewrite. The .accnumbs file contains the reference IDs and the accession numbers of the sequences entered. The .rewrite file contains the text of the submission with the accession numbers added. Both these files are placed in the user's directory. The .rewrite file is also placed in the Submission Text field in the database.

If Auinsub is run in the pub mode, but there is a hold date in the submission with a date later than the current date, Auinsub will alert the user and prompt him/her to change the mode to hup.

As Figure C-1 illustrates, the program prints messages regarding problems.

In the pub mode, Auinsub sets the statuses "GEN citation created" and "SEQUENCE read into database." In the hup mode, Auinsub sets the statuses "GEN citation created" and "SUB hold until published."

C.2 Compseq

This program is used primarily by annotators.

C.2.1 Summary

The Compseq program compares two or more sequences and locates areas of alignment.

compseq [-#] [-k] *filename1 filename2 filename3...*

is the minimum size repeat to consider. (Default is 10.)

-k option keeps the /tmp files for debugging.

```
In alphabet in which alignment was found:

    0   atggcaacgttttcaacgagattttctagggctattcgcgatccc.........................
    0   tatcgagtcgatgcgttttattagcgttttctagggctattcgcgatccc.....................
    0   tgcgtaatcgatgcgagcgccgtcctcgatgattatgcatggtcgggggggatcgtagtagctgatacgt

   45   .... [GATATACGATGC] tgggggactagcgattacgcgatcgatcgtagtttatcgtacggcgaga
   50   .... [GATATACGATGC] ggggattatctccgaatctatctaggtacg.....................
   70   taga [GATATACGATGC] ggggattatctccgaatctatctcatgctagctaactcggcatcg......

  107   gcggcgccgtaccggcgatcgtagcggagcggcgcgatcgat
   92   ..........................................
  131   ..........................................

 less - Hit space to continue (END)
```

Figure C-2 Screen output from the Compseq program: a comparison of three sequences. The matching base pairs are in capital letters and enclosed in brackets. A file called malign.out is also created. It contains a full comparison, including percentage alignments, distance between repeats, and starting positions.

C.2.2 Description

The Compseq program reads two or more files that contain sequences and locates areas of alignment. It produces an alignment map with the aligned areas enclosed in brackets and printed in capital letters. A full report is generated to the file malign.out. It contains the sequence lengths and percentage alignments, a table of aligned repeats, and two alignment maps.

This program is often used to compare two sequences suspected of being identical. It is also used to find small differences between almost identical sequences.

C.3 Humgene

This program is used by annotators and the person responsible for GDB links.

C.3.1 Summary

The Humgene program is used to search the humgene.dc2 document for lines that contain a given word (usually a product, gene name, or map location). The program the produces a list of those lines.

```
humgene word
```

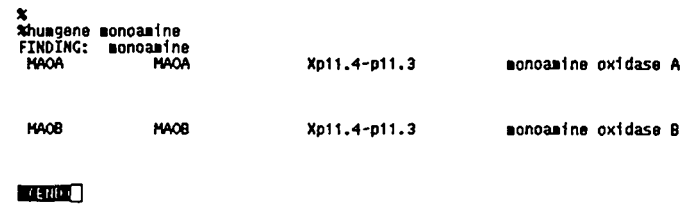

Figure C-3 The Humgene program. Pressing Return or q will bring back the system prompt.

C.3.2 Description

The program searches the humgene.dc2 document for the given word or character string. If you separate each word with a space, humgene will look for one word at a time. The output consists of all the lines that contain the word. Use Return to move through the list. When the end of the list is reached, Return will bring back the system prompt. You may leave the list at any time by pressing q.

C.4 The Loc Programs

The Loc programs search the database and display some of the links between various entities. The user specifies a particular entity, and the program finds information linked to that entity. All the programs will ask for a password. Enter your AWB password.

For most of the programs the % sign may be used for a wildcard, meaning any string of characters. Use wildcards with caution!

C.4.1 Summaries

Loc acc

The Loc acc program finds a given accession number and reports the locus name, sequence length (in base pairs), and reference ID associated with that accession number. Several accession numbers may be entered at one time.

```
loc acc accession_number accession_number...
```

```
%loc acc m11301
Password:
TTHTOPISO       M11301 (length = 1088)  rf_id = 10500
%_
```

Figure C-4 Output from the Loc acc program.

Loc des

The Loc des program finds descriptions of sequences using reference IDs. It produces the accession numbers of the sequences linked to the given rf_id as well as the definition line for each of those sequences. Enter only one ID at one time.

```
loc des rf_id
```

```
%loc des 10283
Password:
10283     M33807    E.coli periplasmic acid glucose-1-phosphatase (agp) gene, co
mplete cds.
%
```

Figure C-5 The Loc des program.

Loc key

The Loc key program finds the locus name and accession number of sequences linked to a given keyword. Enter only one keyword at a time. Sometimes a keyword is more than one word. Enter the entire keyword or just the first word, as Figure C-5 illustrates.

Be careful with wildcards and parts of keywords that are very common. For example, the command "loc key protein" will generate a very large list. And "%protein%" will be much larger.

```
loc key keyword
```

```
%loc key terminus
Password:
 FPV11KB    D00295     terminus of genome
 DROTER     M19140     terminus protein
 ECOTERE    X64000     terminus site
%
```

Figure C-6 The Loc key program.

Loc loc

The Loc loc program takes a locus name and finds the accession numbers and reference IDs linked to the entry that contains it. Wildcards do not work for this program. Enter one or more locus names.

```
loc loc locus_name locus_name...
```

```
%loc loc HUMTCARA BOVRS71LV
Password:
 locus      accession  rf-id
 HUMTCARA   X63455     140663
 locus      accession  rf-id
 BOVRS71LV  J00034     16611
%
```

Figure C-7 The Loc loc program.

Loc per

The Loc per program searches the database for the name of the person entered and prints the database ID of the person and the name as it appears in the database. Enter only one name at a time.

```
loc per name
```

```
%loc per Johnson
Password:
        10855              Johnson,       D.
        11110              Johnson,       Samuel  A
(2 rows affected)
%
```

Figure C-8 The Loc per program. The program prints the person ID and the name as it appears in the database.

Loc ref

The Loc ref program lists the database ID and entity type of a variety of entities linked to a given reference. Enter the reference ID of the paper. More than one reference may be entered at one time.

```
loc ref rf_id rf_id...
```

```
%loc ref 125479 139286
Password:
 reference  database_id entity_type
 125479     70839       en entry
 125479     M69416      sq sequence
 reference  database_id entity_type
 139286     76755       en entry
 139286     98994       en entry
 139286     98995       en entry
 139286     98996       en entry
 139286     98997       en entry
 139286     M76128      sq sequence
 139286     M96679      sq sequence
 139286     M96680      sq sequence
 139286     M96681      sq sequence
 139286     M96682      sq sequence
%
```

Figure C-9 The Loc ref program.

Loc wor

The Loc wor program finds the Worksheets of a given name and produces a list of the entities linked to each Worksheet. The list includes entity types and ID numbers. Enter only one Worksheet name, but wildcards are allowed

```
loc wor worksheet_name
```

```
%loc wor gcr
Password:
 gcr 5/22          pa paper          140289
 gcr5/27           pa paper          118874
 gcr5/27           pa paper          146114
 gcr5/7            pa paper          139855
 gcr5/7            pa paper          144490
 gcr5/7            pa paper          145229
 gcr5/7            pa paper          145468
%
```

Figure C-10 The Loc wor program.

C.5 Lookfor

This program is used by GenBank dataflow and annotation staff.

C.5.1 Summary

The Lookfor program takes one or more accession numbers and searches the database for papers linked to that sequence. It produces some citation information and a list of the reference statuses applied.

`lookfor` *`accession_number accession_number...`*

The program will ask for a password. Enter you AWB password.

```
%lookfor m37275
Password:
Searching genbank for m37275 . . .
m37275
PAPER 144762    J. Mol. Evol. 34, 254-258 (1992) Published
TITLE  Genetic code and phylogenetic origin of oomycetous mitochondria
         Apr 14 1992  GEN citation created          Melissa Pena
         Apr 14 1992  MATCH attempted              Melissa Pena
         Apr 14 1992  DATA set for release         Melissa Pena
         May 21 1992  DATA set for release         Melissa Pena
         May 21 1992  DATA distributed
         May 27 1992  DATA distributed
         Aug 10 1992  DATA distributed
//
PAPER 82169     Genetics 118, 649-663 (1988) Published
TITLE  Drosophila melanogaster mitochondrial DNA: Gene organization and
       evolutionary considerations
         Jan 12 1990  mda                          Felicia Trujillo
         Aug  1 1990  prp                          GenBank Staff
         Sep  5 1990  dsk                          Mia McLeod
         May 24 1991  DATA distributed
         May 21 1992  DATA distributed
         May 27 1992  DATA distributed
         Aug  3 1992  AN revision                  Maria McLeod
         Aug 10 1992  DATA distributed
//
%
```

Figure C-11 Screen output of the Lookfor program. The accession number has two papers associated with it.

C.5.2 Description

Lookfor finds all the papers linked to a given sequence (or group of sequences). The user types lookfor and the accession numbers, with a space between each. The program searches GenBank for the papers linked to the given accession numbers. It also looks in the HUP files for papers.

If the program finds a paper, it prints the paper's database ID, journal abbreviation with volume, pages, and publication status, the title of the paper, and a list of the reference statuses applied. If it finds more than one paper, they are listed one after the other with double slashes (//) between them.

If it does not find any papers, it prints a message: "*accession _number* not found"

C.6 Mailsplit

Mailsplit is used only by the person responsible for reading E-mail submissions sent to "gb-sub".

C.6.1 Summary

Mailsplit separates mail messages found in a given mail file, assigning each a filename. It also prints out a copy of each file. The program reads the incoming mail file if no filename is given. It places multiple messages from a single author in a single file.

```
mailsplit [-help] [-noprint] [-1stpage] [-nomail]
                                        [-automail] filename
```

-help option prints the usage above.

-noprint prevents the program from printing copies of the messages.

-1stpage prints only the first page of the messages.

-nomail prevents the program from sending the response letters.

-automail automatically mails the responses without piping a copy to the screen.

```
%mailsplit -nomail mailfolder1
splitting up and printing mailfolder1
leaving mailfolder1 as is (not copying)
Creating new files, starting with /home/transposon8/gcr/mess0903.ag
number of messages:     3
number of senders:      2
writing 2 file(s)
printing ENTIRE /home/transposon8/gcr/mess0903.ah
[ 3 pages * 1 copy ] left in /home/transposon8/gcr/mess0903.ah.ps
printing ENTIRE /home/transposon8/gcr/mess0903.ag
[ 1 page * 1 copy ] left in /home/transposon8/gcr/mess0903.ag.ps
OUTPUT:
splitting up and printing mailfolder1
leaving mailfolder1 as is (not copying)
Creating new files, starting with /home/transposon8/gcr/mess0903.ag
number of messages:     3
number of senders:      2
writing 2 file(s)
13,24 w  /home/transposon8/gcr/mess0903.ag
1,12 w  /home/transposon8/gcr/mess0903.ah
25,$ w >>  /home/transposon8/gcr/mess0903.ah
[ 3 pages * 1 copy ] left in /home/transposon8/gcr/mess0903.ah.ps
[ 1 page * 1 copy ] left in /home/transposon8/gcr/mess0903.ag.ps
%
```

Figure C-12 Running the Mailsplit program. The example is run in the nomail mode, which does not send response letters. Notice that three messages are sent to two files because two of the senders are the same. The file names begin with mess0903.ag, which indicates that mailsplit was run earlier in the day and mess0903.aa through mess0903.af were assigned.

C.6.2 Description

Mailsplit is used by the person who reads the daily "gb-sub" and "update" mail. It separates the mail by sender, creates files for the submissions, and prints them out.

The program, when run without options, does the following tasks:

- Copies the entire mail queue to the mbox.DATE folder.

- Splits the queue into separate files giving each a unique name.
- Prints out each file.
- Creates a response to each item and pipes the message to the screen.
- Prompts the user to edit and send the letter.

The naming convention for the files is messDATE.letters. For example, mess0726.aj will be the name of the file for the 10th (aj) message on July 26th. If mailsplit is run more than once on the same day, the names will pick up where they left off. For example, if the last file from the program run in the morning is mess0726.bf, then the name of the first file created by program in the afternoon will be mess0726.bg.

The -1stpage option is used when the message is very big and only the first page is needed. The -1stpage option will have an effect the entire mailfile. So it is best to put the first-page-only messages in a separate mail file and use the -1stpage option on that file.

The -automail option is for the confident user who knows that the response letters are correct and does not need to see them before they are sent.

C.7 Pubabbrev

C.7.1 Summary

The Pubabbrev program examines an authorin file for name of a journal, if the abbreviation for the journal is not correct, pubabbrev tells the user the correct abbreviation.

`pubabbrev` *`filename`*

The program will ask for a password. Enter your AWB password.

```
%pubabbrev auth2
Password:
checking for the following publications:
----------------------------------------
J. Virol.
Journal of Virology


checking for "J. Virol." ...
 CORRECT ABBREV

checking for "Journal of Virology" ...
 CHANGE TO:     J. Virol.
%pubabbrev auth2
Password:
checking for the following publications:
----------------------------------------
Journal of Virology


checking for "Journal of Virology" ...
 CHANGE TO:     J. Virol.
```

Figure C-13 Output from the Pubabbrev program. The file "auth2" contains an authorin transaction.

C.7.2 Description

Pubabbrev runs a quick check of an authorin transaction to be sure the journals referred to have the correct abbreviation. Enter the name of the program and then the name of the file with the authorin transaction code.

The program finds the publication names. If the abbreviation is correct, it displays the message, "CORRECT ABBREVIATION." If the abbreviation is incorrect, it displays the message "CHANGE TO:" and the correct abbreviation.

C.8 Ref

C.8.1 Summary

Ref goes through a file containing the text of an Authorin submission and prints out citation information.

```
ref filename
```

```
%
% ref example.authorin
hold_date = "01-OCT-1992";
pub_status = "In Preparation";
title = "Cloning and structural analysis of the rat TNF locus,
year = "1992";
%
```

Figure C-14 The output from the Ref program.

C.8.2 Description

Give Ref the name of the file and it will return with the hold date (if it exists), the publication status, the publication type, the title of the article, and the year of publication. This program works only on files that contain an Authorin submission.

C.9 Refsum

C.9.1 Summary

The Refsum program prints a summary of the reference information for a given reference ID or group of reference IDs.

```
refsum rf_id rf_id...
```

```
%refsum 10500
Password:
PAPER 10500     J. DNA Seq. 10, 159-172 (1992) Published
TITLE  Cloning, sequencing and expression of a novel topoisomerase gene
       from Thermus acquaticus.
AUTHORS  Johnson, Samuel
         Boswell, James
         Stone, John C.
//
%_
```

Figure C-15 The Refsum program.

C.9.2 Description

Refsum produces a brief summary of the reference information associated with a given reference ID. The output is the reference ID, the journal abbreviation (including volume and page numbers), the title of the paper, and the list of authors. Each of the summaries is separated by a pair of slash marks. If the program cannot find the reference ID, it makes no report.

C.10 Statlist

C.10.1 Summary

The Statlist program produces the title and statuses applied to a given reference. Enter the reference IDs with a space between each.

```
statlist rf_id rf_id...
```

```
%statlist 140289
Password:
PAPER 140289    (1992) In Prep.
TITLE  Subdivision of flagellar region III of Escherichia coli and
       Salmonella typhimurium and identification of two additional genes
         Jan 15 1992  GEN citation created            David Stretton
         Jan 15 1992  SUB hold until published        David Stretton
         Jan 15 1992  SUB acknowledged               David Stretton
         May 12 1992  SEQUENCE read into database     Deborah Cucchiara
         May 13 1992  AN paper received in queue      Michelle March
         May 22 1992  AN annotation begun
         May 22 1992  SEQUENCE read into database     George Reese
         May 22 1992  GEN citation created           George Reese
         May 22 1992  MATCH made (references combine  George Reese
         May 22 1992  DATA sent to submitter
         May 27 1992  DATA set for release
         May 27 1992  DATA distributed
//
%
```

Figure C-16 The Statlist program.

C.10.2 Description

Statlist finds the reference statuses that have been applied to a given paper. The output is reference ID, the publication status, the title of the paper and a list of reference statuses applied (including the date and the name of person who set the status).

C.11 Subfind

C.11.1 Summary

The subfind program finds information about submissions to GenBank. The user enters the last name and first initial of a submitter/author, and the program produces numbered list of authors with other submission information. If you want more information about the submission, enter its number at the prompt.

```
subfind Last_name First_initial
```

```
%subfind Baker J.
Password:
Searching for submissions for Baker, J....
No submissions in genbank submitted directly by "Baker, J."...
check for submissions in papers that have "Baker, J." as an author? y
1) Jack Parker Feb 11 1992 8:16AM Electronic mail                sb 18461 RF 1411
30
2) Kathleen Coelingh Jun 12 1992 11:25AM Authorin IBM Floppy     sb 21624 RF 1472
62
3) Stephanie Schneider Jan 28 1991 12:00AM Authorin E-mail               sb 11282
 RF 113217
show accession numbers and descriptions for which submissions?
(use numbers, e.g. "5", "1 3 6", or "all")
1 3
1) Jack Parker Feb 11 1992 8:16AM Electronic mail                sb 18461 RF 1411
30
3) Stephanie Schneider Jan 28 1991 12:00AM Authorin E-mail               sb 11282
 RF 113217
        M60848
        Bacteriophage 434 excisionase (xis) gene, 5' end and integrase (int)
        gene, complete cds.
Finished? (y/n) y
%
```

Figure C-17 The Subfind program.

C.11.2 Description

The subfind program takes the name of an author and produces a numbered list that contains the author's name, the date and time of submission, the submission medium, and the submission and reference ID numbers. It then asks if it should search for papers that have the person as an author (respond with a y or n). After that list, the program asks if the user wants it to show the accession numbers and sequence descriptions for the listed submissions. Respond with the number of the listed item(s) or type `all` to see the accession numbers and sequence definitions of all the listed items.

Appendix D The GenBank Database Schema

This appendix documents the structure of all of the tables used in the GenBank database.

Off-site users: You do not need to know any of the information in this appendix to enter and annotate a sequence in the GenBank database. However, it may be useful if you are interested in the structure of the database.

GenBank staff: Read this appendix for to learn the structure of the database.

D.1 Overview

The GenBank database is a database that contains a collection of experimentally determined nucleotide sequences.

D.1.1 Terminology

The following is a list of terminology used throughout this schema.

Entity - A real world object that this schema models. Typically they are represented as rows in tables with a single primary key.

Throughout this section we use the word "table" instead of the more formal term "relation" (from set theory, not to be confused with relationship). We use the word "row" to signify a single entry, or tuple, in a table. We use "column" to designate an attribute in the table.

ISSUE: Other terms: virtual sequence, abstract gene, gene instance, statuses, region, product?

D.1.2 Entity-Relationship Diagrams

The GenBank schema can be broken down into five groups of tables: bibliographic tables, physical context tables, functional context tables, features tables, and operational information tables. Each of the five groups is represented in an Entity-Relationship diagram.

The Bibliographic Tables

A journal article containing a sequence is the most typical source for information in the database. The bibliographic tables are based on the reference information found in an article. The journal article has a title, covers certain pages and is part of a particular publication. The database Reference Table contains one row for each such entity in the database.

The article will be part of a publication. The Publication Table will contain one row for each publication in the database. These may be journals or other types of publications, such as books or theses.

The Person Table in the database contains one row for each person referred to in the database. These include authors, editors, GenBank staff who annotate the data, database users and, in general, anyone who contributes data or annotation to the database.

There is a many-to-many relationship between authors and articles. One author may be associated with more than one article, and a given article may have more than one author. The AuthRef Table represents the many-to-many link between the authors in the Person Table and references in the Reference Table. In an analogous way the EdPub Table represents the many-to-many link between editors in the Person Table and publications in the Publication Table.

Reference statuses describe how a publication moves through the database, when it was received, annotated, distributed, etc. The Refstat Table links the reference with the status and with the person who applied the status.

Bibliographic information also includes an address. The Address Table has one row for each institution represented in the database, including libraries, publishing companies, research facilities, universities, and private companies. There is a one-to-many relationship between people and addresses and between publications and addresses.

The Submission Table contains records of submissions made directly to the database.

The Scan Table records information on articles scanned by database staff. It is linked to the Person and Publication Tables.

It is desirable to link reference information to many of the data items from a given reference, not only to the sequence. Therefore, almost every entity in the database may cite a reference. References are linked to entities through the Reflink Table.

Other Linking Tables

Two other entities, keywords and comments, may also be linked to almost any other entity in the database. The link is made by specifying the entity type and the entity value to which the keyword or comment is to be linked in the Keylink Table or Comlink Table. For example, if the comment number 12 needs to be attached to feature number 10, there will be a record on the Comlink Table. The comment ID would contain the value 12, the entity type would be FEATURE, and the entity value would be 10

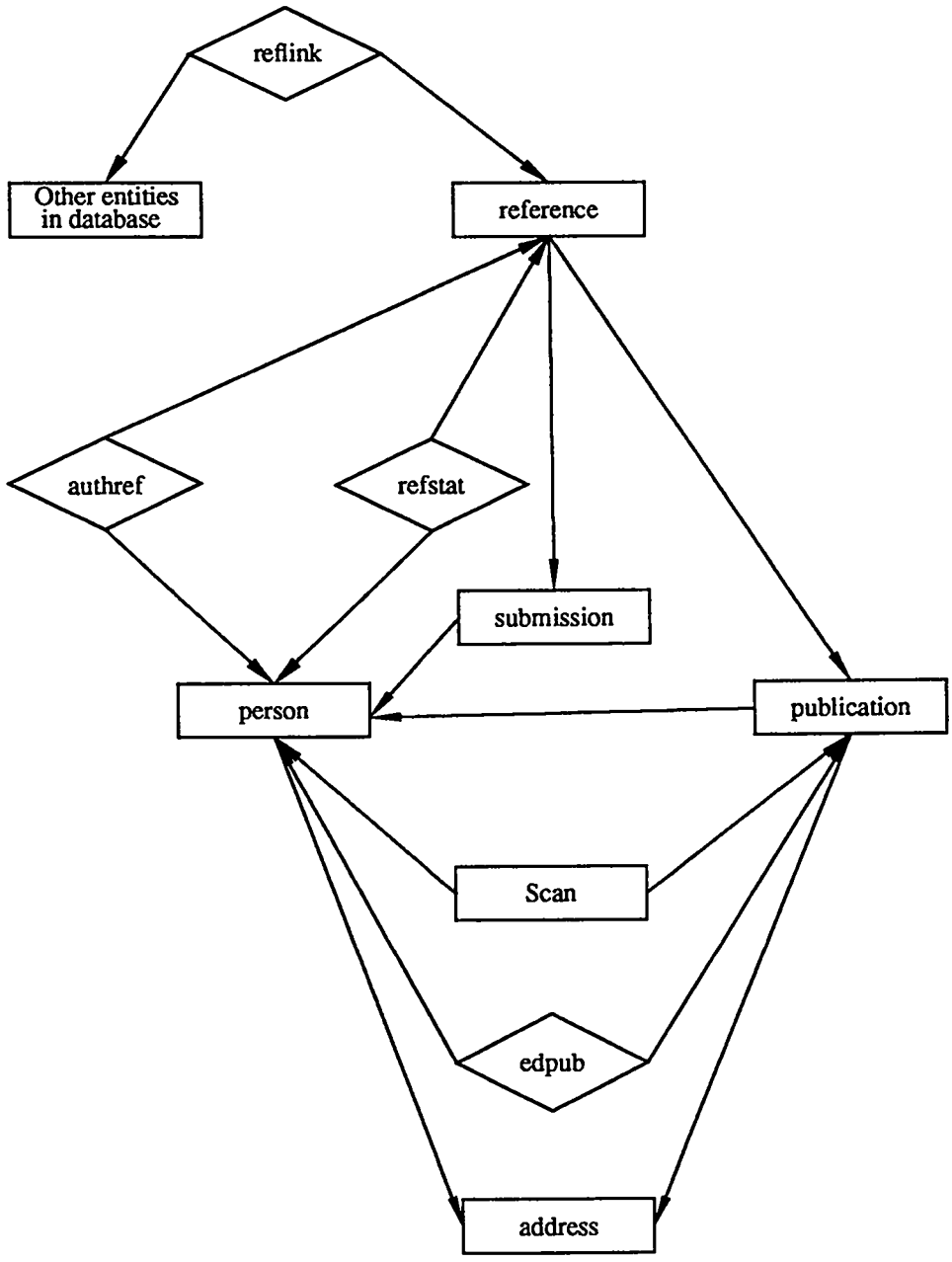

Figure D-1 The Bibliographic tables.

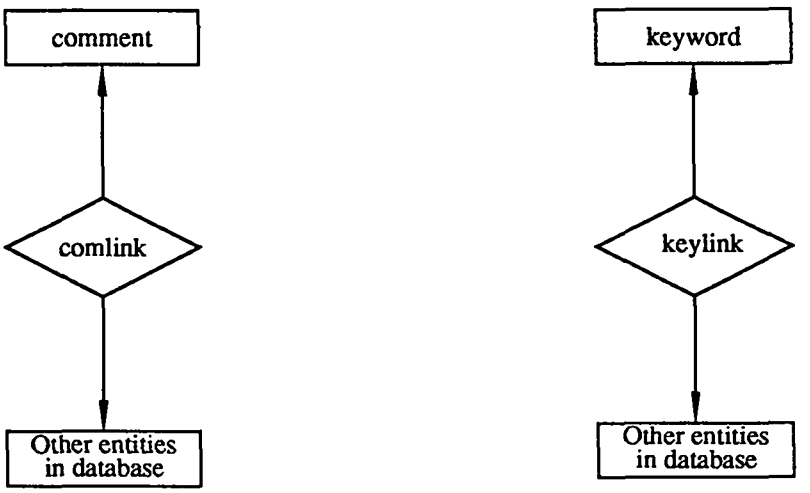

Figure D-2 The Bibliographic tables (continued).

The Physical Context Tables

The Physical Context tables document the biological sources of the sequences by recording taxonomy and source data. These include the Sequence Table, the taxonomy tables, the Entry Table, and the Source Table.

The Source Table contains information about the biological source of the interval. By linking a sequence (or subsequence) to a taxonomy node, this provides the biological context for the sequence.

The taxonomy tables include the Taxonomy Table, Gencode Table, Taxlevel Table, and the Nathost Table. The Taxonomy Table contains a taxonomic tree with each row describing one node. It also contains a column for the parent node, which enables the reconstruction of the taxonomy tree or a sub-tree.

The Gencode Table contains one row for each exception to the genetic code accumulated as one travels from the root to the leaves of the taxonomy tree. Thus, if one

specifies a gencode exception for a particular genus, all species from that genus will automatically use that exception, unless it is explicitly overridden at the species level.

The TaxLevel Table contains the classic taxonomic tree levels. Examples of values in the Taxlevel Table include Kingdom, Phylum, Class, genus, and species.

The Nathost Table links the taxonomy of an organism to the taxonomy of its natural host. For example, suppose that we have the taxonomic classifications of a cat and of a virus that causes Feline Leukemia stored in the database. To establish their relationship there would be record in the Nathost Table. The taxonomy node ID of the organism would be that of the virus, and the taxonomy node ID of the host would be that of a cat.

The Entry Table contains a collection of information associated with sequence. This information is used to create the flat-file format.

The sequence tables, which include the Sequence Table, the Seqel Table, and the Secacc Table, store and link two types of data: actual sequences and instructions to produce virtual sequences. An actual sequence is stored in the database exactly as it was reported in a publication or submitted to the database, without splitting, merging or, interpretation. A virtual sequence is one that is created when other sequences (actual of virtual) are merged.

The Sequence Table contains one row for each sequence in the database. While the information associated with the sequence is stored in the Sequence Table, the actual sequence information is stored in the Text Table.

The Seqel Table contains the instructions for creating virtual sequences from component sequences. (Recording the instructions for the merging, rather than the atomic and composite sequences, minimizes redundant storage of the sequence data. Either the component sequences or the composite sequence can be retrieved and analyzed as desired.)

The Secacc Table links secondary accession numbers with sequences, both actual and virtual.

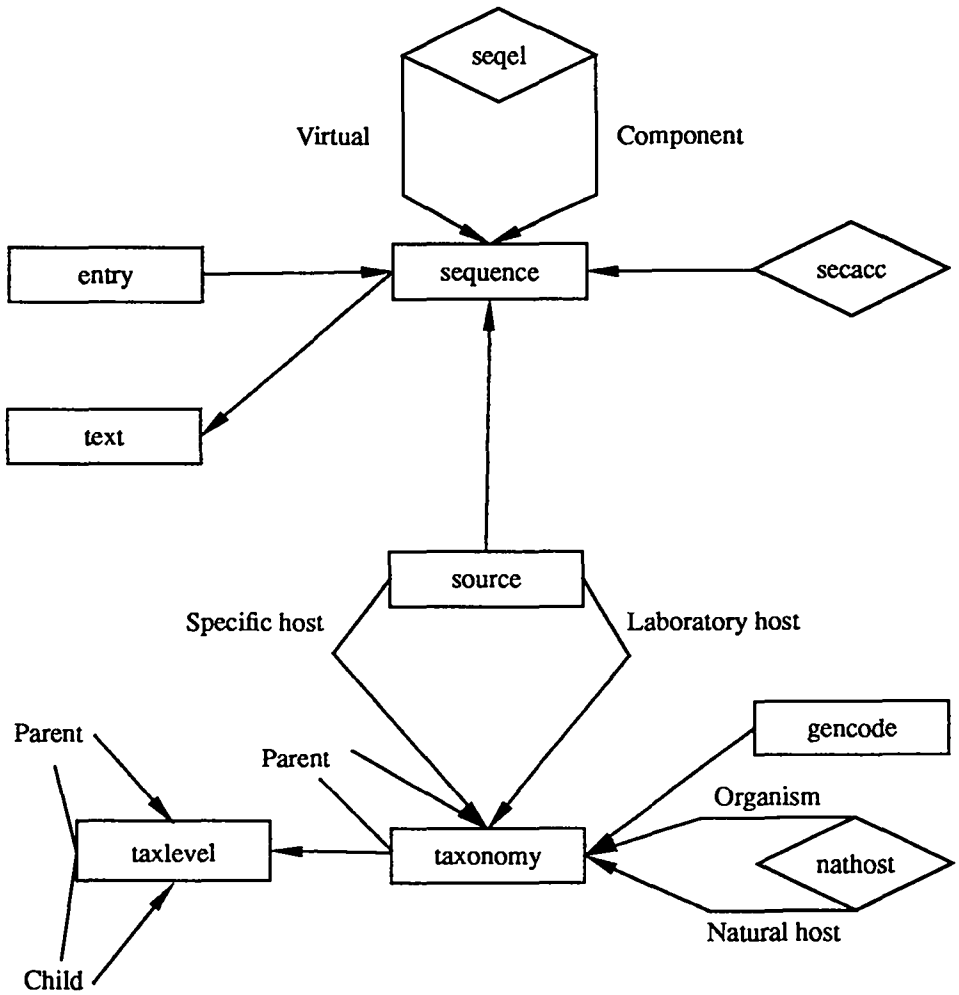

Figure D-3 The Physical Context tables.

The Functional Context Tables

The functional context tables contain information on the functions of the nucleic acid molecules. The Gene Table, the Region Table, and the Product Table allow us to record information on genes and their products without having a corresponding sequence in the database.

The Gene Table simply records gene names and gene symbols. The Geneocc Table is conceptually a picture of an actual gene occurrence, i.e., a specific instance of a gene. It is a link between a gene and (a map location on) an organism.

Gene occurrences can also be linked to regions. A region is a convenient way of grouping several gene occurrences that enables us to characterize them. Examples of these regions are operons, regulons, gene families, etc. The Genreg Table represents the many-to-many relationship between gene occurrences and regions.

The Product Table contains one row for each product in the database. A product is an entity, such as a protein, that is coded for by a region of DNA. The product information can be stored in a tree structure, though this is not presently implemented. The Genprod Table represents the many-to-many relationship between gene occurrences and products.

Gene occurrences and products are stored in separate tables primarily because any one product may have widely varying names in different organisms, several products may be produced from one gene, and two genes may produce different sets of products.

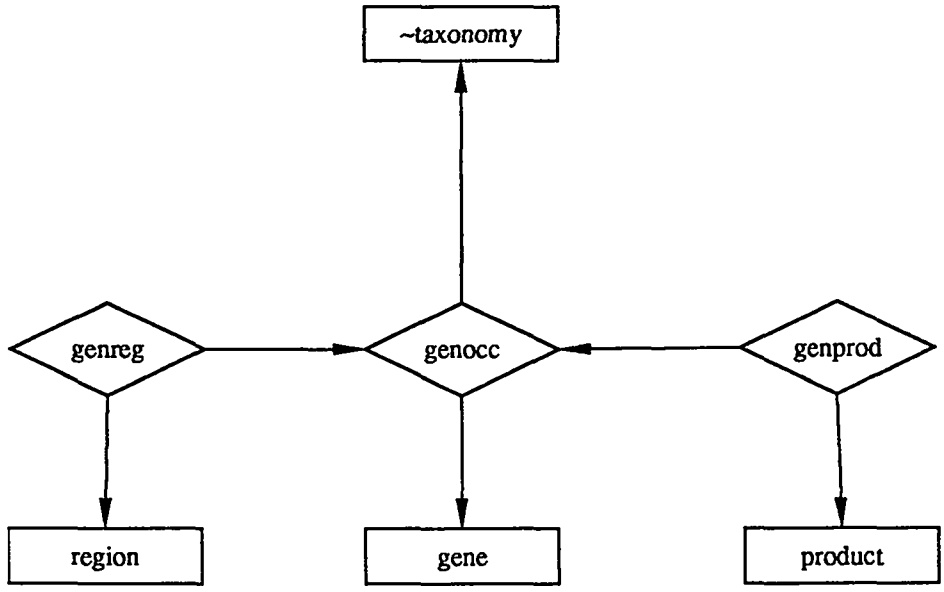

Figure D-4 The Functional Context tables.

The Features Tables

The Features Tables contain biologically interesting intervals on sequences and information about them. There are two types of features stored in the features tables: simple and virtual. Simple features are the single (potentially imprecisely defined) regions of a sequence. Virtual features are composed of one or more component features, each of which may be either simple or virtual. The Compfeat Table describes how component features are combined to produce virtual features.

Feature keys indicate the biological nature of a feature. Features are linked to a feature key through the Featkey Table.

Some features require qualifiers to completely describe them. The Featqual Table contains the qualifier names. The Qualval Table is the linking table for the many-to-many relationship between qualifiers and features.

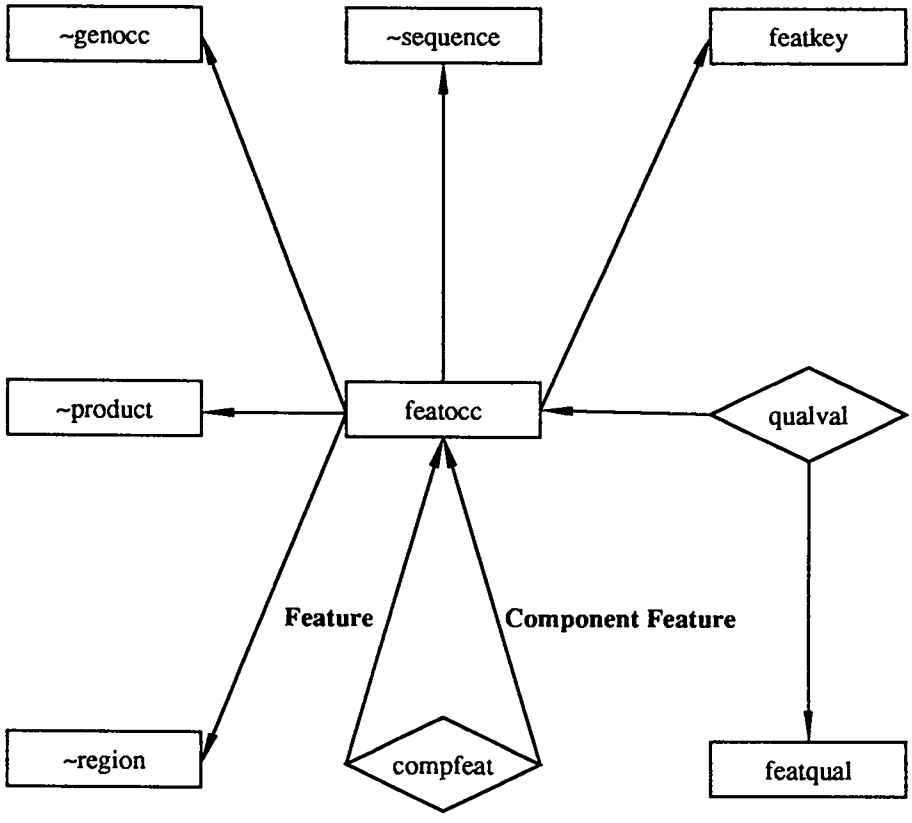

Figure D-5 The Features tables.

The Operational Information Tables

The operational information tables contain secondary information needed to access and understand the primary data.This information includes lists of values assigned to entities, lists of database users, worksheet information, tables that contain information about satellite transmissions, and lists of values for fields.

The Number Table contains the next available ID value for all entities in the database. It is used by the software for assigning numbers.

The Dbuser Table contains the list of users who have access to the database. It is linked to the Person Table.

The Worksheet Table contains the names of worksheets. A worksheet is a tool for grouping entities in a way that is convenient for database users. The Workper Table links a worksheet to a person or persons. The Worklink Table links a worksheet to other entities in the database, which make up the contents of the worksheet.

The Sendcnt Table maintains a count of the packets sent to a satellite site. It is only relevant for the master database. The Receivecnt Table maintains a count of the packets received at a satellite site. It is only relevant for the satellite site.

The Dblist Table contains values for all fields in the database which have a controlled vocabulary. For example, some entities in the database are integers which represent character strings. The Dblist Table contains these associations. It is only for human reference.

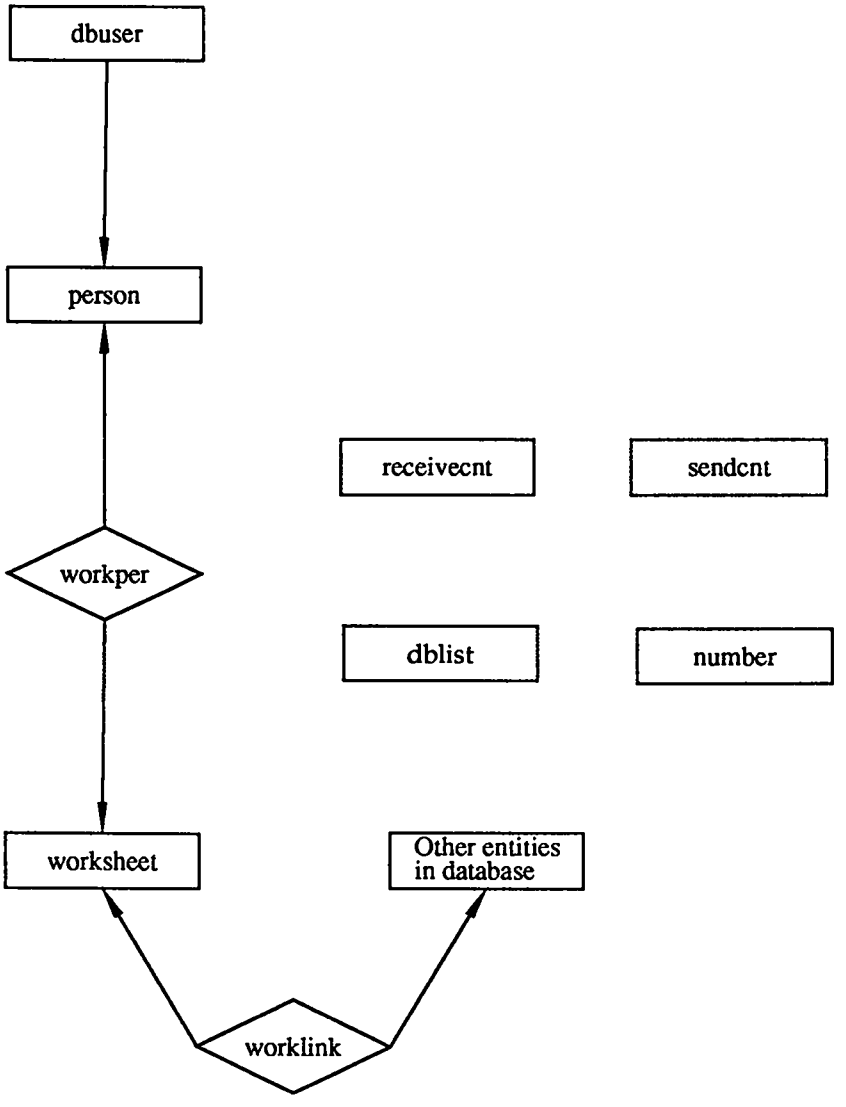

Figure D-6 The Operational Information tables.

D.2 The Schema

D.2.1 The Address Table (AD)

This table has one row for each institution represented in the database. The list includes addresses for libraries, publishing companies, research facilities, universities, private companies, and individuals.

Column	Type	Length	Domain	Nulls
ad_id	c	10		no
	Address ID. The primary key in this table.			
ad_name	c	60		yes
	Full name of the institution.			
ad_dept	c	40		yes
	Name of the department or division.			
ad_address	c	240		yes
	Address of the institution. Includes street address, post office box numbers, etc. Can be used with the specific address of a person (from the Person Table) to generate mailing labels.			
ad_city	c	30		yes
	City of the institution.			
ad_state	c	20		yes
	State of the institution.			
ad_country	c	30		yes
	Country of the institution.			
ad_zip	c	20		yes
	Zip code of the institution.			
ad_phone	c	20		yes
	Phone number of the institution.			
ad_phext	c	10		yes
	Phone extension of the department.			
ad_fax	c	20		yes
	FAX number of the institution.			
ad_email	c	100		yes
	Electronic mail address of the institution.			
ad_telex	c	20		yes
	TELEX number of the institution.			

ad_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

D.2.2 The Alignment Table (AL)

This table provides the instructions to align sequences by storing the boundaries of the overlapping or related regions.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
al_id	**c**	**10**		**no**
Alignment ID. This column plus al_elem from the primary key in this table.				
al_elem	**i**	**4**	**>0**	**no**
Element number. A subscript identifier for each interval within an alignment. Order is not implied by the element number. Order can be deduced from the positions of the overlapping sequences.				
al_sq_id1	**c**	**10**	**(sq_id)**	**no**
Sequence 1 ID number. ID number of one of the sequences.				
al_ltst1	**i**	**4**		**yes**
Left start base 1. Interval of the overlapping region in one sequence.				
al_rtst1	**i**	**4**		**yes**
Right start base 1. Interval of the overlapping region in one sequence.				
al_ltend1	**i**	**4**		**yes**
Left end base 1. Interval of the overlapping region in one sequence.				
al_rtend1	**i**	**4**		**yes**
Right end base 1. Interval of the overlapping region in one sequence.				
al_sq_id2	**c**	**10**	**(sq_id)**	**no**
Sequence 2 ID number. ID number of the other sequence.				
al_ltst2	**i**	**4**		**yes**
Left start base 2. Interval of the overlapping region in the other sequence.				
al_rtst2	**i**	**4**		**yes**
Right start base 2. Interval of the overlapping region in the other sequence.				
al_ltend2	**i**	**4**		**yes**
Left end base 2. Interval of the overlapping region in the other sequence.				
al_rtend2	**i**	**4**		**yes**
Right end base 2. Interval of the overlapping region in the other sequence.				
al_prefer	**i**	**1**	**1-2**	**yes**

Preferred sequence. Indicates the number of the sequence which will be used in the case of a difference. If the first sequence will be used (al_sq_id1) the value of this column will be "1".

al_diff **i** **1** **0-5** **yes**

Difference type. The type of difference between the two sequences: NULL= Unknown; 0= None; 1= Conflict; 2= Revision; 3= Variation; 4= Allele; 5= Mutation. When there is more than one type of difference between two sequences, separate alignment elements are recorded for each type of difference. For example, if an alignment between two sequences contains both a conflict and a variation, then one element would identify the interval containing the conflict and another element would identify the interval containing the variation.

This table also stores instructions to align non-overlapping regions -- segments. Virtual sequences and/or actual sequences may be aligned. One record of this table gives a pair of intervals, one from each of two sequences. Intervals are specified by a start span and an end span.

The set of all records with a given alignment number gives an extended alignment. Alignments may be extended by chaining through the relationships given in this table.

Sometimes we know the relative position of two sequence segments and know nothing about their relative distance. This situation will be represented in the Alignment Table by specifying both sequences and the overlap span will contain a fuzzy end. For example, if sequence 1 is 100 base pairs long and lies upstream of sequence 2 (50 base pairs in length) then sequence 1 positions 101 to NULL overlap with sequence 2 positions 1 to 50.

ISSUE: Finish description of using fuzzy ends in alignments. (09/88).

ISSUE: How will a gap be specified in this table?

ISSUE: Give an example of chaining through this table to create an alignment. (06/88).

D.2.3 The Authref Table (AR)

This table represents the many-to-many relationship between authors and references, linking an author to a corresponding article.

The ar_rf_id and ar_pn_id keys together form the primary key in this table.

Column	Type	Length	Domain	Nulls
ar_rf_id	**c**	**10**	**(rf_id)**	**no**
Reference ID. A foreign key in this table.				
ar_pn_id	**c**	**10**	**(pn_id)**	**no**
Author's person ID. A foreign key in this table.				
ar_lname	**c**	**30**		**yes**
Last name of the author.				
ar_fname	**c**	**30**		**yes**
First name of the author.				
ar_initials	**c**	**10**		**yes**
Author's other initials.				
ar_suffix	**c**	**10**		**yes**
Suffix (Jr., III, etc.) of the person as it appears on the publication.				
ar_order	**i**	**1**	**>0**	**yes**
Number in list. This column is used to re-construct the author list as it appears in the paper.				
ar_iscorr	**i**	**1**	**0-1**	**yes**
Corresponding Author? The value in this column is "1" if the author is the corresponding author for the reference to which it is linked.				
ar_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

This table will also store the author's name as it appears in the linked citation. For example, if Richard J. Robbins publishes one paper as R. Robbins and another paper as R. J. Robbins, we will be able to recreate the correct citation by storing the name as it appears in this table. The correct citation is the citation as it appears in the paper. Richard J. Robbins' full name and other information will be stored in the Person Table.

Note: This aspect is currently not implemented. 1/19/93

This table will also serve as the person synonym table. It would be used in locating the work of an author whose name has appeared in more that one form on articles. Whenever a user refers to a person, and a search of the Person Table fails to return a

match, the software can consult this table to find a synonym matching the supplied name.

This aspect is currently not implemented. 1/19/93

D.2.4 The Clone Table (CN)

This table contains the names of all the clones.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
cn_id	**c**	**10**		**no**
Clone ID. The primary key in this table.				
cn_name	**c**	**25**		**no**
Clone name.				

D.2.5 The Clonesource Table (CS)

This table links clones to sources.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
cs_id	**c**	**10**		**no**
CloneSource ID. The primary key in this table.				
cs_cn_id	**c**	**10**	**(cn_id)**	**no**
Clone ID. A foreign key in this table.				
cs_sc_id	**c**	**10**	**(sc_id)**	**no**
Source ID number. A foreign key in this table.				

D.2.6 The Comlink Table (CL)

This table links comments to other entities in the database.

For a given comment, there are two values involved in linking an entity with that comment. The first, cl_enumbtype, identifies the type of entity being linked, e.g., person, publication, sequence, etc. The second value, cl_enumbval, is the primary key of that entity, e.g., pn_id, pb_id, sq_id, etc.

A comment can be linked to any entity that has a primary key consisting of one column.

The three columns together form a primary key in this table.

Column	Type	Length	Domain	Nulls
cl_cm_id	c	10	(cm_id)	no
Comment ID. ID of comment being linked to an entity. A foreign key in this table.				
cl_enumbtype	i	1	(nm_enumbtype)	no
Entity type. The type of entity being linked to a comment. For example, if the comment explains a sequence, this column will contain the numerical value which corresponds with the entity type "sequence."				
cl_enumbval	c	10	>0	no
Entity value. ID of the entity being linked to a comment. For example, if the comment belongs with entry 4509, this column will contain 4509.				
cl_owner	c	10		yes
The login name of the owner of information in this row.				

Note: The mapping of an integer to its corresponding table can be found in the Number Table.

D.2.7 The Comment Table (CM)

This table contains one row for each comment.

Column	Type	Length	Domain	Nulls
cm_id	c	10		no
Comment ID. The primary key in this table.				
cm_permlev	i	1	>=0	yes
Permission level. This column contains the minimum permission level required to access this comment. It is compared with user pn_permlev value from the Person Table. This allows for the existence of "footnotes"; that is, comments that are not available to the public. Value 0 = Private; 1 = Public.				
cm_pn_id	c	10	(pn_id)	no
Person ID of the author of the comment. A foreign key in this table.				
cm_date	date	8		yes
Creation Date. Date the comment was created.				
cm_owner	c	10		yes
The login name of the owner of information in this row.				

Comments are linked to entities by the Comlink Table. A comment can be attached to any entity in the database that has a primary key consisting of one column.

D.2.8 The Compfeat Table (CF)

This table describes how features are combined to produce other features. This table serves to construct virtual features from component features.

Column	Type	Length	Domain	Nulls
cf_id	c	10		no
The primary key in this table.				
cf_numb	i	4	>0	no
Argument number. The value in this column is used to determine the order that the component features are assembled into the virtual.				
cf_fo_id	c	10	(fo_id)	no
Feature occurrence ID. A foreign key in this table. ID from the Featocc Table of the virtual feature.				
cf_cfo_id	c	10	(fo_id)	no
Component feature ID. A foreign key in this table. Each value in this column refers to a row in the Featocc Table associated with the component feature. There may be many component features for each virtual feature.				
cf_owner	c	10		yes
The login name of the owner of information in this row.				

D.2.9 The Dblist Table (DB)

This table contains columns that have a restricted vocabulary associated with them. It also handles columns in which the internal database values are a set of integers, and the external value that the user sees is a character string.

Column	Type	Length	Domain	Nulls
db_column	c	30		no
Column name.				
db_alpha	c	80		yes
Associated character string.				
db_value	i	1		yes
Integer value within the database.				

Note: This table is only for human reference. It is not currently used by the software. 1/19/93

D.2.10 The Dbuser Table (DU)

This table contains one row for each user in the database.

Column	Type	Length	Domain	Nulls
du_id	**c**	**10**		**no**
Database user ID. The primary key in this table.				
du_lname	**c**	**30**		**yes**
Database user's last name.				
du_fname	**c**	**30**		**yes**
Database user's first name.				
du_initials	**c**	**10**		**yes**
Database user's other initials.				
du_suffix	**c**	**10**		**yes**
Suffix (Jr., III, etc.) of the database user.				
du_email	**c**	**100**		**yes**
Electronic mail address.				
du_username	**c**	**10**		**yes**
Unix user login name.				
du_usernumb	**i**	**4**		**yes**
Unix user number.				
du_password	**c**	**30**		**yes**
Database user's password				
du_lastlogin	**date**	**8**		**yes**
Last login.				
du_pn_id	**c**	**10**	**(pn_id)**	**no**
Person Table ID. A foreign key in this table.				
du_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

D.2.11 The Division Table (DV)

This table is a reference table that records corresponding taxonomic nodes from the Taxonomy Tree and the three letter GenBank division codes.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
dv_id	c	10		no
Division ID. The primary key in this table.				
dv_tx_id	c	10	(tx_id)	no
Taxonomy ID. The ID of the node in the Taxonomy Tree. A foreign key in this table.				
dv_division	c	3		no
Division. The division codes are: PRI, ROD, MAM, VRT, INV, PLN, BCT, VRL, PHG, ORG, RNA, and SYN.				

The division for a sequence can be determined by entering the Taxonomy Tree at the node corresponding to the sequence and moving toward the root of the tree until the first node specified in this table is reached.

D.2.12 The Document Table (DC)

This table contains on-line documents.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
dc_id	c	10		no
Document number. The primary key in this table.				
dc_pn_id	c	10	(pn_id)	no
Document author ID. Person Table ID for the author of the document.				
dc_title	c	80		yes
Document title.				
dc_date	date	8		yes
Creation date.				
dc_permlev	i	1	>=0	yes
Permission level. This column contains the minimum permission level required to access this document. It is compared with user pn_permlev value from the Person Table. This allows for existence of documents that are not available to the public.				

Some of the on-line documents contained in this table are documents such as guidelines for data submission, practice of GenBank work, help for using the database, distribution notes, and planning.

D.2.13 The Edpub Table (EP)

This table represents the many-to-many relationship between editors and publications, linking an editor to a corresponding publication. The two foreign keys together create a primary key in this table.

Column	Type	Length	Domain	Nulls
ep_pb_id	**c**	**10**	**(pb_id)**	**no**
Publication Table ID. A foreign key in this table				
ep_pn_id	**c**	**10**	**(pn_id)**	**no**
Person Table ID. A foreign key in this table.				
ep_lname	**c**	**30**		**yes**
Editor's last name.				
ep_fname	**c**	**30**		**yes**
Editor's first name.				
ep_initials	**c**	**10**		**yes**
Editor's other initials.				
ep_suffix	**c**	**10**		**yes**
Suffix (Jr., III, etc.) of the editor as it appears on the publication.				
ep_order	**i**	**1**		**yes**
Number in list. This column is used to re-construct the editor list as it appears in the publication.				
ep_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

This table will also store the editor's name as it appears in the linked publication. The full name and other information will be stored in the Person Table.

The design of this table allows for the tracking of a current editor of a publication. However the design does not, at this time, allow for the tracking of previous editors of a publication.

D.2.14 The Entry Table (EN)

This table contains one row for each GenBank entry.

Column	Type	Length	Domain	Nulls
en_id	c	10		no
Entry ID. The primary key in this table.				
en_sq_id	c	10	(sq_id)	no
Sequence Table ID. A foreign key in this table.				
en_entname	c	10		yes
Entry name. Old entry name (LOCUS name.)				
en_div	c	3		yes
Division code. Legal values are PRI, ROD, MAM, VRT, INV, PLN, BCT, VRL, PHG, RNA, ORG, SYN, UNA.				
en_entdate	date	8		yes
Date that will appear on the LOCUS line when a flat-file is generated.				
en_moltype	c	20		yes
Entry molecule type.				
en_distribute	i	1	1-3	yes
Distribution status. Legal values are private, public, distribute. 1 = public; 2 = private; 3 = distribute.				
en_distdate	date	8		yes
Date of last distribution.				
en_seg	i	1		yes
Identifies the specific segment of a segmented entry.				
en_segnumb	i	1		yes
Segment number. Total number of segments in the segmented entry.				
en_source	c	250		yes
Source line. Old source line.				
en_org	c	80		yes
Organism line. Currently not used. 1/19/93				
en_origin	c	250		yes
Old origin line. This column contains mapping and localization data.				
en_topology	c	20		yes
Locus line topology. Legal values: circular, linear.				
en_owner	c	10		yes
The login name of the owner of information in this row.				

D.2.15 The Featkey Table (FK)

This table contains one row for each feature key in the database.

Column	Type	Length	Domain	Nulls
fk_id	**c**	**10**		**no**
Key ID. The primary key in this table.				
fk_name	**c**	**50**		**yes**
Key name. The name of the feature key. (See table for legal values.)				
fk_gbkey	**c**	**50**		**yes**
GenBank name. The name of the current GenBank feature key.				
fk_ekey	**c**	**50**		**yes**
Former EMBL feature key.				
fk_def	**c**	**240**		**yes**
Definition of the feature key.				

D.2.16 The Featocc Table (FO)

This table contains one row for each feature in the database.

Column	Type	Length	Domain	Nulls
fo_id	c	10		no
Featocc ID. The primary key in this table.				
fo_desc	c	240		yes
Featocc description. This column contains free text information about the feature, which appears in the note column of the flat-file and is a /note qualifier in the flat-file.				
fo_label	c	15		yes
Featocc label. This column is meant to hold whatever label (number or name) that researchers apply to the feature. The point of the column is to allow people to assign a "name" (e.g., "exon 3" or "intron JC-1") to a feature by means of specifying what the feature is (exon, intron) and its number or name (3, JC-1). Order is not necessarily implied when researchers assign the numbers or names.				
fo_fk_id	c	10	(fk_id)	yes
Featkey Table ID. A foreign key in this table.				
fo_sq_id	c	10	(sq_id)	no
Sequence Table ID. A foreign key in this table.				
fo_go_id	c	10	(go_id)	no
Genocc Table ID. A foreign key in this table.				
fo_pr_id	c	10	(pr_id)	no
Product Table ID. A foreign key in this table.				
fo_rg_id	c	10	(rg_id)	no
Region Table ID. A foreign key in this table.				
fo_ltstart	i	4	>0	yes
Left Start. This is the initial point in the range of positions for the beginning of the feature.				
fo_ltend	i	4	>0	yes
Left End. The final point in the range for the beginning of the feature.				
fo_rtstart	i	4	>0	yes
Right Start. This is the initial point in the range of positions for the end of the feature.				
fo_rtend	i	4	>0	yes
Right End. The final point in the range for the end of the feature.				
fo_5sym	c	1	+,/	yes
5' Symbol from the flat-file features table. (Temporary column to keep from				

losing information during the flat-file to relational table conversion.)

fo_3sym c 1 +,/ yes
3' Symbol from the flat-file features table. (Temporary column to keep from losing information during the flat-file to relational table conversion.)

fo_operator i 1 0-5 yes
Complex feature operator. Operator used for construction. Value 1= (not used); 2= group; 3= join; 4= one-of; 5= order; 6= replace.

fo_iscomp i 1 0-1 yes
Complementary strand. This feature occurrence is on the complementary strand of the sequence when this flag is "1".

fo_isexper i 1 0-1 yes
Experimentally determined. This flag is "1" if the feature occurrence was identified experimentally and "0" if the feature occurrence was identified by pattern recognition.

fo_is5cmpl i 1 0-1 yes
5' Completeness. If the start position of the named internal is the start position of the feature occurrence (e.g., end of cds, end of 5'URT) then this flag would be "1", meaning that the 5' end of this feature is complete.

fo_is3cmpl i 1 0-1 yes
3' Completeness. If the end position of the named internal is the end position of the feature occurrence (e.g., end of cds, end of 5'UTR) then this flag would be "1", meaning that the 3' end of this feature is complete.

fo_cdn_st i 4 0-3 yes
Codon start. The reading frame, when applicable, of the feature occurrence. This column contains a phase number for determining where coding starts in the assembled sequence of the feature. 0 or NULL signifies that the feature is not translated; 1, that translation starts at the beginning of the feature; 2, one past the beginning; and 3, two past the beginning. Note that the numbering is relative to the assembled sequence; this is a change from the previous absolute system. (i.e., if a exon goes from 300 to 400 on sequence X12345, and a codon starts in the second position, fo_cdn_st = 2 not 301).
Legal values:1, 2, 3.

fo_isconsensus i 1 0-1 yes
Consensus. This featocc matches the consensus when this flag is "1".

fo_replace c 255 yes
String of characters used by "replace" operator. Currently not implemented 1/19/93.

fo_number i 1 yes
Feature number. For example, if this feature is exon 2 of a particular gene, the fo_number would be "2".

fo_ispsuedo i 1 yes
Is pseudo? If yes, the value of this column will contain "1".

fo_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

Some features "occur" on more than one sequence. For example, a feature may need to be attached to an actual sequence and a virtual sequence created from the actual sequence.

D.2.17 The Featqual Table (FQ)

This table contains one row for each qualifier in the database.

Column	Type	Length	Domain	Nulls
fq_id	**c**	**10**		**no**
Qualifier ID. The primary key in this table.				
fq_qualname	**c**	**20**		**no**
Qualifier name. The name of the qualifier.				
fq_ownner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

For a listing of qualifier names see Db List.

D.2.18 The Gencode Table (GC)

This table contains one row for each exception to the genetic code needed to translate protein coding regions.

Column	Type	Length	Domain	Nulls
gc_id	c	10		no
Gencode ID. The primary key in this table.				
gc_tx_id	c	10		no
The Taxonomy Table ID. A foreign key in this table.				
gc_codon	c	3		yes
Codon exception. The codon that codes for the "exception" amino acid.				
gc_aminoacid	c	3		yes
Amino acid. The exception amino acid.				
gc_position	c	10		yes
Position. The position of the codon, e.g. stop, start, internal.				

An example of a record in this table might be:

gc_tx_id - an identification number that points to the node in the Taxonomy Tree of the organism (strain, organelle, etc.) that utilizes the universal code exception;

gc_codon - tag;

gc_aminoacid - Gln;

gc_position - internal.

Exceptions to the genetic code are accumulated as one travels from the root to the leaves in the Taxonomy Tree, (i.e., the rule is inherited downward). There is one exception per record. The exceptions in this table are used when there is an alternate code at a certain Taxonomy Level (e.g., the organism or strain level). Every time a certain codon appears it codes for the amino acid recorded in this table. Point aberrations, alternate amino acid is incorporated at one or more discrete points, will be handled with a feature with separate features keys (e.g. codes_for_Ala, codes_for_Phe).

D.2.19 The Gene Table (GN)

The Gene Table contains one row for each gene name in the database.

Column	Type	Length	Domain	Nulls
gn_id	c	10		no
Gene ID. The primary key in this table.				
gn_name	c	100		no
Gene name or gene symbol.				
gn_allele	c	20		yes
Gene allele.				
gn_locus_id	c	20		yes
GDB gene locus_id.				
gn_loc	c	10		yes
Gene locus name.				
gn_ecnum	c	20		yes
Gene EC number.				
gn_probe	c	255		yes
Gene probe. List of probes for gene.				
gn_isofficial	i	1		yes
Is official? Is approved by GDB?				
gn_owner	c	10		yes
The login name of the owner of information in this row.				

We make a distinction between abstract genes and instances of genes or gene occurrences. A gene occurrence is a gene linked to a taxonomy node. An example of an abstract gene might be the cheA gene, and an example of a gene occurrence might be the cheA gene from E.coli K12.

ISSUE: How will gene names with non-ascii elements be handled?

ISSUE: Remaining outstanding issues that need to be figured out... how to handle alleles and how to handle phenotypes.

ISSUE: There are at least three examples in E.coli where there are two gene occurrences in tandem with the same name.

D.2.20 The Genesyn (GS)

The Gene Synonym Table contains synonyms for gene names. Alleles are also recorded in this table.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
gs_id	**c**	**10**		**no**
Gensyn ID. The primary key in this table.				
gs_gn_id	**c**	**10**	**(gn_id)**	**no**
Gene ID. A foreign key in this table.				
gs_genename	**c**	**20**		**yes**
Alternate gene name.				
gs_gn_idal	**c**	**10**	**(gn_id)**	**no**
Allele gene ID number.				

ISSUE: Alleles should be a link to gene occurrences. Alleles really probably belong in a separate table and not lumped with the gene name synonyms.

D.2.21 The Genocc Table (GO)

The Genocc Table represents the many-to-many link between genes and taxonomy nodes.

Column	Type	Length	Domain	Nulls
go_id	c	10		no
Gene occurrence ID. The primary key in this table.				
go_gn_id	c	10	(gn_id)	no
Gene ID. A foreign key in this table.				
go_tx_id	c	10	(tx_id)	no
Taxonomy node ID. A foreign key in this table.				
go_numbexon	i	4	>0	yes
Number of exons represented in this gene instance. For example, the human hCS-1 gene has five exons.				
go_isseq	i	1	0-1	yes
Is template sequenced? This flag is "1" if we have a sequence of the gene occurrence in the database, (i.e., we can store information on a gene occurrence without having a corresponding sequence in the database).				
go_map	c	20		yes
Linkage map position. The genetic linkage position map for the gene represented by this gene template.				
go_owner	c	10		yes
The login name of the owner of information in this row.				

We make a distinction between abstract genes and instances of genes or gene occurrences. A gene occurrence is a gene linked to a taxonomy node. An example of a abstract gene might be the cheA gene, and an example of a gene occurrence might be the cheA gene from E.coli K12.

D.2.22 The Genprod Table (GP)

This table represents the many-to-many relationship between genomic templates and products. The two foreign keys form the primary key in this table.

Column	Type	Length	Domain	Nulls
gp_go_id	**c**	**10**	**(go_id)**	**no**
Genocc Table ID. A foreign key in this table.				
gp_pr_id	**c**	**10**	**(pr_id)**	**no**
Product Table ID. A foreign key in this table.				

ISSUE: This table links gene occurrences to products. We may also want a gene/ product link.

D.2.23 The Genreg Table (GR)

This table represents the many-to-many relationships between gene occurrences and regions. The two foreign keys form the primary key in this table.

Column	Type	Length	Domain	Nulls
gr_rg_id	**c**	**10**	**(rg_id)**	**no**
Region Table ID. A foreign key in this table.				
gr_go_id	**c**	**10**	**(go_id)**	**no**
Genocc Table ID. A foreign key in this table.				

D.2.24 The History Table (HX)

This table tracks the history of any of the primary entities in the database.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
hx_id	c	10		no
History ID. The primary key in this table.				
hx_enumbtype	i	1	(nm_enumbtype)	no
Entity number type. This value identifies the type of entity being documented. The value for this column must come from the Number Table. For example, if the entity is a reference, this column would contain the value from the Number Table that identifies the reference number.				
hx_enumbvalue	c	6		no
Entity number value. This column contains the actual value of the entity number. For example, if the above reference had a reference number of 3275, this column would contain 3275.				
hx_transnumb	c	10	>0	yes
Transaction number.				
hx_action	i	1	1-3	no
Action. Description of action taken. Value 1= added; 2= updated; 3= deleted.				
hx_pn_id	c	10	(pn_id)	no
Person Table ID. A foreign key in this table.				
hx_date	date	8		yes
Date of work.				

This table can record every occasion on which the row for a particular sequence was changed. By looking at this history of changes, a user would be able to make a judgement about the reliability of the data as it now stands.

D.2.25 The Keylink Table (KL)

This table links keywords to other entities in the database.

For a given keyword, there are two values involved in linking that keyword with the entity. The first, kl_enumbtype, identifies the type of entity being linked, e.g., person, publication, sequence, etc. The second value, kl_enumbval, is the primary key of that entity, e.g., pn_id, pb_id, sq_id, etc.

A keyword can be linked to any entity that has a primary key consisting of one column.

The three columns together form a primary key in this table.

Column	Type	Length	Domain	Nulls
kl_kw_id	c	10	(kw_id)	no
ID of keyword being linked to an entity. A foreign key in this table.				
kl_enumbtype	i	1	(nm_enumbtype)	no
Entity type. The type of entity being linked to a keyword. For example, if the keyword is associated with a sequence, this column will contain the numerical value which corresponds with the entity type "sequence.".				
kl_enumbval	c	10	>0	no
Entity value. ID of the entity being linked to a keyword. For example, if the word belongs with entry 4509, this column will contain 4509.				
kl_owner	c	10		yes
The login name of the owner of information in this row.				

Note: The mapping of an integer to its corresponding table can be found in the Number Table.

D.2.26 The Keyword Table (KW)

This table contains one row for every keyword in the database. Keywords can be linked to any entity by the Keylink Table.

Column	Type	Length	Domain	Nulls
kw_id	c	10		no
Keyword ID. The primary key in this table.				
kw_name	c	80		no
Keyword name.				
kw_owner	c	10		yes
The login name of the owner of information in this row.				

D.2.27 The Library Table (LB)

This table is used to record information on the physical location of publications, i.e. not all of the publications that are recorded in the database are available in all of the libraries that the database staff uses. This table is a quick reference for in-house management.

Column	Type	Length	Domain	Nulls

lb_id	**c**	**10**		**no**
Library ID. The primary key in this table.				
lb_pb_id	**c**	**10**	**(pb_id)**	**no**
Publication ID. A foreign key in this table.				
lb_ad_id	**c**	**10**	**(ad_id)**	**no**
Library address ID. A foreign key in this table.				
lb_spanstart	**c**	**5**		**yes**
Volume span start. The span of volumes found in any particular library.				
lb_spanend	**c**	**5**		**yes**
Volume span end. The span of volumes found in any particular library.				

D.2.28 The Nathost Table (NH)

This table links the taxonomy nodes of the source organism to the taxonomy nodes of their natural hosts.

Column	Type	Length	Domain	Nulls
nh_tx_ido	c	10	(tx_id)	no
Organism taxonomy node ID.				
nh_tx_idh	c	10	(tx_id)	no
Host taxonomy node ID.				
nh_owner	c	10		yes
The login name of the owner of information in this row.				

D.2.29 The Number Table (NM)

This table is used to record the last value of id numbers for tables that have a single column as their primary key.

Column	Type	Length	Domain	Nulls
nm_enumbtype	i	1	>=0	no
Entity number type. This is simply used to allow other tables to refer to number type in this table in a short-hand way.				
nm_numbname	c	20		no
Name of number type.				
nm_lastvalue	c	10	>0	no
Last value used. The value in this column is the value last used for this type of number. When getting a number from this table, this should be incremented.				
nm_distribution	i	1		yes
Indicates if a table is distributed to satellites. Values: 0= Not distributed, 1= Distributed publicly, 2= Local field.				
nam_indexcnt	i	1		yes
Number of indexes on this table at the time of last distribution.				
nm_rowcnt	i	4		yes
Number of rows in this table at the time of last distribution.				

An example of the use of this table would be when adding a new reference to the Reference Table the rf_id should be equal to one greater that the current value for the nm_lastvalue (for the appropriate entity) column in this table.

One reason for using this table instead of simply consulting the table in question to find the next available number is speed. Checking this table should be faster.

Another reason is that this table allows us to pre-assign blocks of numbers before they actually appear in the database. For example, when a block of numbers is assigned to a journal, the reference value would simply be incremented by the number needed.

D.2.30 The Person Table (PN)

This table has one row for each person represented in the database.

Column	Type	Length	Domain	Nulls
pn_id	c	10		no
Person ID. The primary key in this table.				
pn_lname	c	30		yes
Last name.				
pn_fname	c	30		yes
First name.				
pn_initials	c	10		yes
The initials of any and all middle names. Additional information will not appear as output according to the flatfile definition.				
pn_suffix	c	10		yes
Suffix (Jr., III, etc.) of the person.				
pn_ad_id	c	10	(ad_id)	no
Address Table ID. A foreign key in this table.				
pn_address	c	60		yes
Specific address. This specific address would be used along with the corresponding address from the Address Table to create the full address of the person. An example of a specific address is: Mail Stop K710.				
pn_email	c	100		yes
Electronic mail address.				
pn_wkphone	c	20		yes
Work phone number.				
pn_phext	c	10		yes
Phone extension.				
pn_hmphone	c	20		yes
Home phone number.				
pn_fax	c	20		yes
FAX number. (If different than the institution.)				
pn_telex	c	20		yes
TELEX number. (If different than the institution.)				
pn_permlev	i	1	>= 0	yes
Permission level. This determines what each person is allowed to do in the database. For example, a user with level 0 permission would be able to submit data but could not change existing data.				

pn_usernumb	**i**	**2**	**>= 0**	**yes**
Operating system user number.				
pn_lastlogin	**date**	**8**		**yes**
Date of last login.				
pn_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

The list of people in this table covers everyone who contributes data or annotation to the database, including journal editors, journal contacts, authors, database staff, and users.

D.2.31 The Prodsyn Table (PX)

This table contains synonyms for nodes or alternate entry points in the product tree. There can be synonyms for all levels of the product tree, not just the terminal nodes.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
px_id	**c**	**10**		**no**
Prodsyn ID. The primary key in this table.				
px_pr_id	**c**	**10**	**(pr_id)**	**no**
Product Table ID. A foreign key in this table.				
px_nodesyn	**c**	**50**		**no**
Node synonym.				

D.2.32 The Product Table (PR)

Each row in this table describes one node of the tree. Each node gets a unique number. Each node of the tree is a record of the table.

Column	Type	Length	Domain	Nulls
pr_id	c	10		no
Product tree node ID. The number of a node in the product tree. The primary key in this table.				
pr_nodename	c	100		no
Product node name.				
pr_abbrev	c	3		yes
Product abbreviation as used by Locus names.				
pr_pr_id	c	10	(pr_id)	no
Product Table ID. A foreign key in this table.				
pr_ecnumb	c	15		yes
Ec number for the product.				
pr_owner	c	10		yes
The login name of the owner of information in this row.				

A product is that which is coded for by a region of the DNA (gene). Product information is stored in a tree structure. The top of the tree will list general products, e.g. RNA's and Proteins.

D.2.33 The Publication Table (PB)

This table contains one row for every publication in the database. This table includes special publications for direct submissions (e.g. "Submission to GenBank").

Column	Type	Length	Domain	Nulls
pb_id	c	10		no
Publication ID. The primary key in this table.				
pb_abbrev	c	60		yes
Abbreviation. This is the official NLM abbreviation.				
pb_name	c	240		yes
Full name of the publication. The title and the year of publication for a monograph or the title of a book will be recorded in this column.				
pb_issn	c	10		yes
ISSN number.				
pb_ad_id	c	10	(ad_id)	no
Address Table ID of the publishing company or institution. A foreign key in this table.				
pb_pn_id	c	10	(pn_id)	no
Person Table ID of the contact person with the publication. A foreign key in this table.				
pb_type	i	1	0-1	yes
Publication type. This column indicates the type of publication. Submissions are data supplied directly to the database. A monograph would be recorded as a book. Value NULL= Unknown; 1= Journal Article; 2= Thesis; 3= Proceedings; 4= Book; 5= Unpublished; 6= Submission to GenBank; 7= Submission to EMBL; 8= Submission to DDBJ.				
pb_db	i	1	0-3	yes
Overseeing database. This column records which database is responsible for overseeing the scanning of this publication. Value NULL= Unknown; 0= None; 1= GenBank; 2= EMBL; 3=DDBJ.				
pb_isscan	i	1	0-1	yes
Is scanned? This column is "1" if this publication (journal) is routinely scanned by the databases.				
pb_dbinteract	i	1	0-5	yes
This column records the database/journal interaction status.				

Value NULL= Unknown; 0= None; 1= Approached; 2= Refused; 3= Interested; 4= Implementing; 5= Adopted;

ISSUE: Will we handle journals that change names as separate journals?

D.2.34 The Pubsyn Table (QV)

This table is used to aid in understanding user input.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
py_id	**c**	**10**		**no**
Pubsyn ID. The primary key in this table.				
py_pb_id	**c**	**10**	**(pb_id)**	**no**
Publication Table ID. A foreign key in this table.				
py_pubsyn	**c**	**40**		**no**
Synonym.				

Whenever a user refers to a publication, the software can consult this table to find out what was probably meant. Given a string which is supposed to be a publication name, a program would first consult the Publication Table. If there is no "name" or "abbrev" record matching the supplied string, the program would search this table for a synonym matching the string or, failing that, the synonym "closest" to the supplied string. (The definition of "closest" would change as our software became more sophisticated.)

D.2.35 The Qualval Table (QV)

This is a linking table for the many-to-many relationship between qualifiers and features.

Column	Type	Length	Domain	Nulls
qv_id	c	10		no
Qualval ID. The primary key in this table.				
qv_fo_id	c	10	(ft_id)	no
Featocc Table ID. A foreign key in this table.				
qv_fq_id	c	10	(fq_id)	no
Featqual Table ID. A foreign key in this table.				
qv_enumbtype	i	1	(nm_enumbtype)	yes
Entity number type holding the qualifier value.				
qv_enumbval	c	10	>0	yes
Entity number value.				
qv_value	c	200		yes
Qualifier value if not stored in an entity type/entity value combination.				
qv_owner	c	10		yes
The login name of the owner of information in this row.				

D.2.36 The Receivecnt Table (RC)

This table assures that the packets are processed in the proper order. It is used only at the receiving end of the satellite.

Column	Type	Length	Domain	Nulls
rc_email	c	75		no
Presently not used.				
rc_cnt	i	4		no
Count. Holds the number of the last packet processed by the satellite.				

D.2.37 The Reference Table (RF)

This table contains one row for every citation in the database.

Column	Type	Length	Domain	Nulls
rf_id	c	10		no
Reference ID. The primary key in this table.				
rf_pb_id	c	10	(pb_id)	yes
Publication Table ID. A foreign key in this table.				
rf_volume	c	5		yes
Volume.				
rf_issue	c	5		yes
Issue				
rf_pgst	i	4	>0	yes
Start page.				
rf_pgend	i	4	>0	yes
End page.				
rf_year	i	4	>1900	yes
Year.				
rf_title	c	250		yes
Title.				
rf_pubstat	i	1	0-4	yes
Publication status.				
Value Null= Unknown; 0= Unpublished; 1= Thesis; 2= Published; 3= In Press; 4= In Preparation; 5= Submitted for publication.				
rf_holddate	date	8		yes
Hold date. This is the date until which we hold the data from distribution.				
rf_ishere	i	1	0-1	yes
Is paper here? This column contains a flag indicating whether we have a copy of the actual paper or not. If we have the paper, it is either in our file or in the possession of the current owner.				
rf_hasseq	i	1	0-1	yes
Has sequence? This column is "1" if the paper contains at least one sequence. NULL implies that we will look at this reference to see if it contains a sequence.				
rf_cit_address	c	100		yes
Citation address. A temporary column to hold the citation address as it appears in the current GenBank row in unpublished, thesis, and book citations. This column will contain the publishing company and place of publications				

for books. This column will probably disappear when we are operating fully in the relational system.

rf_data_sub **c** **10** **(sb_id)** **yes**
Data submission ID number. When the data is directly submitted to the database from the author, the submission is archived in the Submission Table and linked to its corresponding reference by this column.

rf_data_ref **c** **10** **(rf_id)** **yes**
Data reference ID number. Sometimes we have data previously entered into the database. Then we get a reference that apparently covers the same data. Rather than enter the data again we will create a record in the reference table reflecting the new reference. The column, rf_data_ref will then contain the ID number of the reference from which the original data is from.

rf_owner **c** **10** **yes**
The login name of the owner of information in this row.

ISSUE: Expand description to cover Thesis and books.

ISSUE: How will we handle references with spilt page spans? (i.e., 200-221 and 264-265)?

D.2.38 The Reflink Table (RL)

This table links references to other entities in the database.

For a given reference, there are two values involved in linking that reference with the entity. The first, rl_enumbtype, identifies the type of entity being linked, e.g., person, publication, sequence, etc. The second value, rl_enumbval, is the primary key of that entity, e.g., pn_id, pb_id, sq_id, etc.

A reference can be linked to any entity that has a primary key consisting of one column.

The first three columns form a primary key in this table.

Column	Type	Length	Domain	Nulls
rl_rf_id	**c**	**10**	**(rf_id)**	**no**
Reference Table ID. A foreign key in this table.				
rl_enumbtype	**i**	**1**	**(nm_enumbtype)**	**no**
Entity number type. Type of entity being linked to a reference. For example, if the reference is to be linked to a sequence, this column will contain the value which corresponds with the entity type "sequence."				
rl_enumbval	**c**	**10**	**>0**	**no**
Entity number value. ID of the entity being linked to a reference. For example, if this reference belongs with entry 4509, this column will contain 4509.				
rl_issource	**i**	**4**	**0-1**	**yes**
Data source. The value in this column is "1" if the reference is the primary source of the data.				
rl_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

The mapping of an integer to its corresponding table in found in the Number Table.

This table can link any two references which supposedly present the same data. For example, this table would link a published article and the direct submission of the same data or a sequence and a published revision. By the same data we mean data from the same sequencing event. When either sequence is retrieved a "merged" sequence would normally be presented.

D.2.39 The Refstat Table (RA)

This table contains one row for every status applied to every reference in the database.

Column	Type	Length	Domain	Nulls
ra_id	**c**	**10**		**no**
Reference Status ID. The primary key in this table.				
ra_rf_id	**c**	**10**	**(rf_id)**	**no**
Reference Table ID. A foreign key in this table.				
ra_date	**date**	**8**		**no**
The date on which the above status was applied.				
ra_pn_id	**c**	**10**	**(pn_id)**	**no**
Person Table ID. A foreign key in this table.				
ra_type	**c**	**50**		**no**
Type of status. The status being applied to the above reference.				
ra_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

For a complete listing of status references see Dblist.

D.2.40 The Refsub Table (RS)

Column	Type	Length	Domain	Nulls
rs_is Refsub ID	c	10		no
rs_rf_id A foreign key.	c	10		no
rs_sb_id A foreign key	c	10		no

D.2.41 The Region Table (RG)

The Genomic Region Table records information on regions of the genome, such as operons, regulons, gene families, etc. Each region is given a unique number.

Column	Type	Length	Domain	Nulls
rg_id Region ID. The primary key in this table.	c	10		no
rg_regname Region name.	c	100		yes
rg_regtype Region type. (operon, regulon, gene family, chromosome, genome, etc.)	c	20		yes

ISSUE: Think about changing name of this table. Folks seemed to think that region was a funny way to describe groups such as gene family. "Grouping" table was suggested.

D.2.42 The Regsyn Table (RS)

The Region Synonym Table contains synonyms for regions.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
rs_id	**c**	**10**		**no**
Regsyn ID. The primary key in this table.				
rs_rg_id	**c**	**1-**	**(rg_id)**	**no**
Region Table ID. A foreign key in this table.				
rs_regname	**c**	**20**		**no**
Alternate region name.				

D.2.43 The Rsite Table (RE)

This table is used to verify restriction enzyme data on input.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
re_id	**c**	**10**		**no**
Enzyme ID. The primary key in this table.				
re_name	**c**	**20**		**yes**
Restriction enzyme name.				
re_seq	**c**	**20**		**yes**
Restriction sequence. The restriction enzyme recognition sequence where "^" indicates the cut site. For example, cc^agg, means that the enzyme TaqXI will recognize the sequence "ccagg" and cuts after the second "c". Ambiguities are represented by the IUPAC code.				

D.2.44 The Scan Table (SN)

This table records the database scanning status for the publications.

Column	Type	Length	Domain	Nulls
sn_id	**c**	**10**		**no**
Scan ID. The primary key in this table.				
sn_pb_id	**c**	**10**	**(pn_id)**	**no**
Publication Table ID. ID of the publication scanned. A foreign key in this table.				
sn_volume	**c**	**5**		**no**
Volume.				
sn_issue	**i**	**2**	**>0**	**yes**
Issue.				
sn_pgst	**i**	**2**	**>0**	**yes**
Start page.				
sn_pgend	**i**	**2**	**>0**	**yes**
End page.				
sn_year	**i**	**2**	**>1900**	**yes**
Year.				
sn_pn_id	**c**	**10**	**(pn_id)**	**no**
Person Table ID. ID of the person who did the scan. A foreign key in this table.				
sn_numseq	**i**	**2**	**>=0**	**yes**
Number of sequences found in the issue.				
sn_pubdate	**date**	**8**		**yes**
Publication date as it appears on the journal.				
sn_scandate	**date**	**8**		**yes**
Scan date.				
sn_libdate	**date**	**8**		**yes**
Library date. Date journal arrived at the library.				
sn_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

D.2.45 The Secacc Table (SA)

This table links all of the accession numbers found in any current GenBank entry with the primary accession number for that entry. This primary accession number is the accession number used in the Entry Table.

Column	Type	Length	Domain	Nulls
sa_id	c	10		no
Secondary Accession ID. The primary key in this table.				
sa_sq_id	c	10	(sq_id)	no
Sequence Table ID. A foreign key in this table.				
sa_secacc	c	10		yes
Secondary sequence ID number.				
sa_owner	c	10		yes
The login name of the owner of information in this row.				

One use of this table is to help locate an entry given any of the accession numbers. The ACCESSION number line from the flat-file can be recreated with this table.

D.2.46 The Sendcnt Table (SD)

This table keeps track of the last packet sent to the satellites.

Column	Type	Length	Domain	Nulls
sd_satcnt	i	4		no
Packet count. Last packet sent.				
sd_peercnt	i	4		no
Presently not used.				

D.2.47 The Seqel Table (SE)

This table provides the instructions to create virtual sequences. Each element describes one piece of a virtual sequence (including sequences that are relevant but not used, these being identified by 0 start and end).

Column	Type	Length	Domain	Nulls
se_id	c	10		no
Seqel ID. The primary key in this table.				
se_vsq_id	c	10	(sq_id)	no
Virtual Sequence ID. A foreign key in this table. This is the ID of the sequence made up of the pieces identified in this table. The virtual sequence element ID plus the segment number form the secondary index for this table.				
se_segment	i	4	>0	yes
Segment number. A subscript identifier for each interval within a virtual sequence. The segment number implies order. To extend a virtual sequence upstream a new virtual sequence would be created with one of the components being the virtual sequence being extended.				
se_csq_id	c	10	(sq_id)	no
Component sequence ID. A foreign key in this table. This is the ID of the sequence from which a piece of a virtual sequence is taken.				
se_start	i	4		yes
Start base. The starting base on the component sequence that is being used to assemble the virtual sequence.				
se_end	i	4		yes
End base. The final base on the component sequence that is being used to assemble the virtual sequence				
se_spanstart	i	4		yes
The beginning of the span of the virtual sequence that is credited to the reference which contains the component sequence. This number will be reported as the beginning base in the REFERENCE line of the flat file.				
se_spanend	i	4		yes
The end of the span of the virtual sequence that is credited to the reference which contains the component sequence. This number will be reported as the final base in the REFERENCE line of the flat file.				
se_owner	c	10		yes
The login name of the owner of information in this row.				

The keys se_start and se_end are used to assemble the virtual sequence. Se_spanstart and se_spanend are used to credit the reference which contains the component sequences. To clarify this, consider the following example.

Suppose we have two sequences A and B. The se_start of A is 1, and the se_end is 200. The se_start of B is 1, and the se_end is 300. The two sequences are combined to produce the virtual sequence V. However, the bases 101 to 200 in A happen to be the same as bases 1 to 100 in B. In the virtual sequence this span is merged, so the total length of the virtual sequence is 400 bases.

To give proper credit for the components of the virtual sequence, se_spanstart will be 1 and se_spanend will be 200 for A. For B se_spanstart will be 101 and se_spanend will be 400.

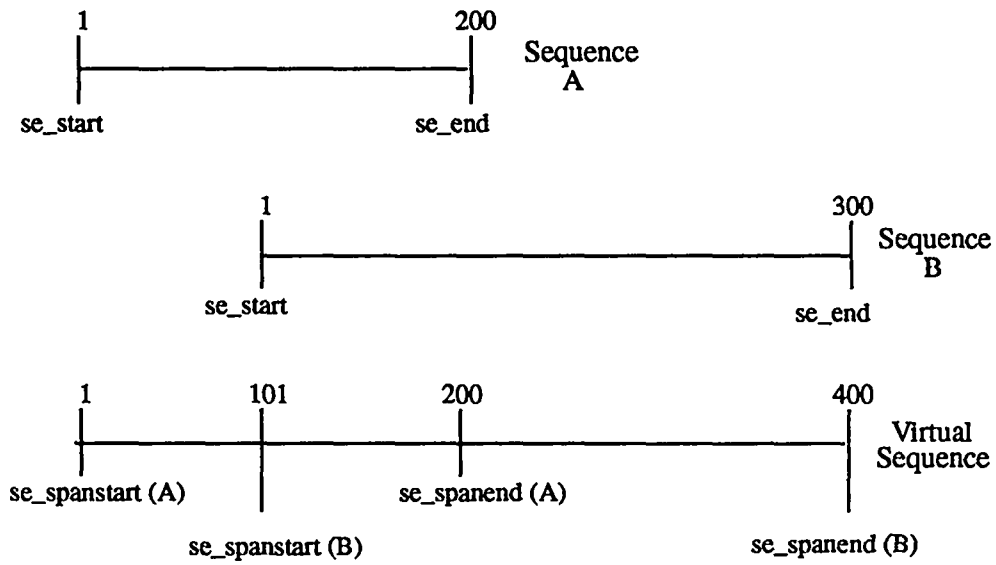

Figure D-7 Span information in the Seqel table.

D.2.48 The Sequence Table (SQ)

This table contains one row for each sequence in the database.

Column	Type	Length	Domain	Nulls
sq_id	**c**	**10**		**no**
Sequence ID. The primary key in this table. The sequence ID is the accession number.				
sq_length	**i**	**4**	**>0**	**yes**
Sequence length.				
sq_isvirtual	**i**	**1**	**0-1**	**yes**
If this is a virtual sequence, this flag is "1".				
sq_annotqual	**i**	**1**	**0-2**	**yes**
Annotation quality. The code for the annotation quality of a component sequence. Value Null= Unknown; 0= Automatic; 1= Staff entry; 2= Staff review.				
sq_annotquan	**i**	**1**	**0-2**	**yes**
Annotation quantity. The code for the annotation quantity of a component sequence. Value Null= Unknown; 0= Unannotated; 1= Simple; 2= Full.				
sq_desc	**c**	**250**		**yes**
A concise description of the sequence.				
sq_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

Reported sequences are stored as presented, without splitting, merging, or interpretation. Merged sequences have only a virtual existence.

ISSUE: Perhaps we should add a sq_confidence column to store the average (?) confidence level for the sequence.

D.2.49 The Source Table (SC)

This table gives the biological source of the interval. This is in an interval table rather than a sequence table since authors may merge sequences before making a figure in a publication. Molecule type, host, and completeness are functions of the particular sequencing experiment.

Column	Type	Length	Domain	Nulls
sc_id	c	10		no
Source ID. The primary key in this table.				
sc_sq_id	c	10	(sq_id)	no
Sequence Table ID. A foreign key in this table.				
sc_moltype	i	1	1-12	yes
Molecule type.				
Value Null= Unknown; 1= DNA; 2= RNA; 3= mRNA; 4= rRNA; 5= tRNA; 6= snRNA; 7= scRNA; 8= cDNA to mRNA; 9= cDNA to genomic RNA; 10= cDNA to other RNA?; 11= cDNA; 12= peptide.				
sc_ltstart	i	4		yes
Left start. The initial point in the range of positions for the beginning of the sequence.				
sc_rtstart	i	4		yes
Right start. The final point in the range for the beginning of the sequence interval.				
sc_ltend	i	4		yes
Left end.The initial point in the range of positions for the end of the sequence interval.				
sc_rtend	i	4		yes
Right end. The final point in the range for the end of the sequence interval.				
sc_tx_id	c	10	(tx_id)	no
Taxonomy Table ID. A foreign key in this table.				
sc_devstage	c	120		yes
Developmental stage.				
sc_tissue	c	120		yes
Tissue type.				
sc_cell	c	120		yes
Cell type.				
sc_line	c	120		yes
Cell line.				
sc_library	c	120		yes

Tissue or cell library name.

sc_haplo c 120 yes
Haplotype.

sc_sex c 20 yes
Sex/mating type.

sc_ismacro i 1 0-1 yes
Is macronucleus?

sc_isgerm i 1 0-1 yes
Is germline?

sc_iscmpl i 1 0-1 yes
Is complete? This column is "1" if this interval of sequence fully represents the taxonomic node (e.g. complete chromosome). If an interval fully represents more than one taxonomic node (e.g. clone and chromosome) there would be two records in this table, one for the complete clone and one for the complete chromosome.

sc_isprov i 1 0-1 yes
Is provirus? This column is "1" if the sequence is a provirus.

sc_tx_idh c 10 (tx_id) no
Laboratory host tax node (where the experimentalist grew the sample).

sc_tx_idn c 10 yes
Particular host out of which the organism was isolated.

sc_eltype i 4 yes
Element type. For example, mitochondrion or chromosome.

sc_elname c 15 yes
Name of the element or number of the chromosome.

sc_map c 15 yes
Genetic map position. Not necessarily the official map location determined by the data.

sc_desc c 250 yes
Manually added source description.

sc_owner c 10 yes
The login name of the owner of information in this row.

The Source Table and the Taxonomy Table together give a complete taxonomy, including the usual taxonomic divisions, information needed to define a particular individual, information specifying a particular cell and chromosome, and the name of a particular fragment of DNA or RNA.

If an author has presented a "merged" sequence there would be a row in the table for each interval. For example, if a sequence of 500 base pairs is a composite of 100 base

pairs of mRNA and 400 base pairs of DNA, there would be a row in the table for the mRNA with boundaries 1 and 100 and a row for the DNA with boundaries 101 and 500.

ISSUE: Should we also include "specific" source and catalog number of the library? This question applies to the tissues libraries here and to the clone libraries in the taxonomy table. Catalog number refers to like the ATCC number or a catalog number form a company

D.2.50 The Submission Table (SB)

This table contains one row for each direct submission to the database.

Column	Type	Length	Domain	Nulls
sb_id	**c**	**10**	**>0**	**no**
Submission ID. The primary key in this table.				
sb_pn_id	**c**	**10**	**(pn_id)**	**no**
Person Table ID. A foreign key in this table.				
sb_date	**date**	**8**		**yes**
Submission date.				
sb_medium	**i**	**1**	**1-5**	**yes**
Submission medium. Values can be found in the Dblist Table.				
sb_text	**text**			**yes**
The text of the submission.				

A copy of submitted data needs to be archived. It may be possible that the submitted data may differ from the published version (if the data were published). This table is an archive of the text of direct submissions as they first entered our computers.

In reality, the relationship between references and submissions is many-to-many. However, we feel, that the most common asked question will be "what was the data source of this reference?" It is unlikely that greater than one submission will be submitted for each reference. Therefore, we will represent the relationship between references and submissions as a one to n (submission to reference).

We will create a record in the Reference Table for every submission. For example, if data are submitted directly to the database that correspond to published data, we will check the database for the citation. If the citation exists then we will create a record in the submission table for the submission. The submission id number (sb_id) will be added to the record in the Reference Table corresponding to the citation and the column rf_data_sub will contain this submission id, (i.e., the link of submitted data to a published article will be made through the Reference Table by the rf_data_sub column.

If the citation is not in the database a new record will also be created in the Reference Table.

If the submitted data does not correspond to any of our publication types, then a citation will be created in the reference table reflecting that the data are form an unpublished submission (the database to which the submission was made will be specified)

We will allow only one person to be attached to each submission. This person may be an author or another person responsible for giving us the data. The authors may be found by finding all authors linked to the corresponding reference.

Direct updates to the database by the author will also be "submissions".

ISSUE: Do we want to put other info in comment or explicitly in table? If diskette the computer? Operating system? Editor? If mag tape then record length? Block size? Label type? Density? Character code?

D.2.51 The Taxlevel Table (TL)

This table describes the taxonomy tree used to classify sequences. Each record in this table describes one level of the tree.

Column	Type	Length	Domain	Nulls
tl_id	**c**	**10**		**no**
Taxonomy level ID. The primary key in this table.				
tl_levname	**c**	**20**		**yes**
Level name; e.g., "Sub-Phylum".				
tl_tl_idp	**c**	**10**	**(tl_id)**	**no**
Parent level ID. The number of the next higher level.				
tl_tl_idc	**c**	**10**	**(tl_id)**	**no**
Child level ID. The number of the next lower level.				

An example of the use of this table could be, because the top level of the taxonomy tree is kingdom, there is a record in this table where the name of the level is kingdom. In addition, the tl_tl_idp in this row will be zero (meaning there is no parent level), and the tl_tl_idc value will be equal to the tl_id of the next level down the chain.

The taxonomy, as stored in the database, is an extended version of the standard taxonomy. In addition to the inclusion of sub- and super- levels, the taxonomy is extended all the way to clone. This extension allows for the flexibility needed when doing dynamic merges of sequences in the database.

Organelles (mitochondria, chloroplasts, and kinetoplasts), as well as any extrachromosomal piece of DNA (plasmids, etc.), are treated as additional chromosomes of the cell.

ISSUE: Expand description of extended taxonomy... need to acknowledge that not classical and why we make the extension.

D.2.52 The Taxonomy Table (TX)

This table contains our taxonomic tree, with each row describing one node. Each node in the taxonomic tree gets a unique number. Each edge of the tree is a record of the table.

Column	Type	Length	Domain	Nulls
tx_id	**c**	**10**		**no**
Taxonomy node ID. The primary key in this table.				
tx_tl_id	**c**	**10**	**(tl_id)**	**no**
Taxlevel Table ID. A foreign key in this table. This number indicates the level in the tree of the node in question. (Kingdom level, or at the phylum level, etc.) There are two reasons for specifying the level. First, the names of the levels may be different for different organisms. Second, it makes it much easier to check that this table really represents a tree.				
tx_tx_idp	**c**	**10**	**(tx_id)**	**no**
Parent node number. The node number of the parent node in the Taxonomy Tree. By referring to the parent node, this column enables the tree structure (or any sub-tree) to be reconstructed. Given the tx_taxnode of the top of the desired tree, one need only search (recursively) for all nodes with the given parent.				
tx_nodename	**c**	**80**		**yes**
Node name. This column contains the official name (scientific name) of the node.				
tx_name	**c**	**80**		**yes**
Common name.				
tx_div	**c**	**3**		**yes**
The taxonomic division into which this sequence goes.				
tx_abbrev	**c**	**3**		**yes**
The three letter code to be used in the names of entries for this taxonomy node.				
tx_gc_table	**i**	**1**		**yes**
Which gencode table to use. Values are found in the Dblist Table				
tx_seqrep	**i**	**1**	**0-2**	**yes**
Sequence representation. This column records the level of sequence representation of the node stored in the database.				

If this column is "0" (none), we have no sequence representation of this taxonomic node. If this column is "2" (full), merging sequences at this taxonomic level gives a complete representation of the organism's genome at this level.

For example, if we had several sequences that when merged would give the

complete genome of a BALB/c mouse, the tx_seq_rep column would be "2" (full) on the node corresponding to the mouse BALB/c strain. If we had the complete sequence of a clone, then the tx_seq_rep column would be "2" (full) on the corresponding clone node.

Value Null= Unknown; 0= None; 1= Partial; 2= Full.

tx_isdna i 1 0-1 yes
Is DNA in vivo? This column is "1" when the in vivo molecule is DNA.

tx_strand i 1 1-3 yes
Strandedness. This is the strandedness of the in vivo nucleic acid.

Value Null= Unknown; 1= Single stranded; 2= Double stranded; 3= Mixed stranded.

tx_iscircle i 1 0-1 yes
Is topology circular? This column is "1" if the topology of the in vivo nucleic acid molecule is circular.

tx_sciname c 80 yes
Scientific name. Many times the scientific name of an organism can be derived from the taxonomy. However, there are examples (e.g., viruses) where the scientific name is not derivable from the taxonomy.

tx_maploc c 80 yes
Map location. This column has meaning only for those taxonomy levels that are sub-"chromosomal". In those cases, it means the map location of the subchromosomal node on the parent (or, grand parent, etc.) chromosome node. (e.g., if a tax node describes a clone, then the map_location column can contain the location of the clone on the first parent node representing a complete molecule).

tx_isapprove i 1 yes
Is this an official GenBank entry?

The Source Table and the Taxonomy Table together give a complete taxonomy, including the usual taxonomic divisions, information needed to define a particular individual, information specifying a particular cell and chromosome, and the name of a particular fragment of DNA or RNA.

It is possible for two different organisms to have the same name because naming is regulated by different bodies. Thus the only guaranteed unique name of an organism is its full taxonomy.

D.2.53 The Taxsyn Table (TS)

This table contains any alternate names for any node in the Taxonomy Tree.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
ts_id	**c**	**10**		**no**
Taxsyn ID. The primary key in this table.				
ts_tx_id	**c**	**10**	**(tx_id)**	**no**
Taxonomy Table ID. A foreign key in this table.				
ts_taxnodesyn	**c**	**20**		**no**
Taxonomy node synonym.				

D.2.54 The Text Table (TT)

This table is used to store free text data such as sequences and comments. The length of the text column is RDBMS-dependent.

The table also links the text to other entities in the database.

Column	Type	Length	Domain	Nulls
tt_id	**c**	**10**		**no**
Text ID. The primary key in this table.				
tt_enumbval	**c**	**10**	**>0**	**no**
Entity number value. The ID of the entity being linked to the text. For example, if the text belongs with entry 4509, this column will contain 4509				
tt_enumbtype	**i**	**1**	**(nm_enumbtype)**	**no**
Entity number type. The type of entity linked to the text. For example, if the text is associated with a sequence, this column will contain the value that corresponds with the entity type "sequence."				
tt_seqno	**i**	**2**	**>= 0**	**no**
Sequence number. Used to keep track of text spread over several rows for one enumbtype and enumbval. Presently not used. 1/19/93				
tt_text	**text**	**16**		**yes**
Entity Text. The text itself.				
tt_owner	**c**	**10**		**yes**
The login name of the owner of information in this row.				

D.2.55 The Virtseq Table (VS)

This table holds symbols of virtual sequences.

Note: This table is currently not in use. 1/19/93

Column	Type	Length	Domain	Nulls
vs_id	**c**	**10**		**no**
Virtual sequence ID. The primary key in this table.				
vs_segment	**i**	**4**		**yes**
Segment number.				
vs_sq_id	**c**	**10**	**(sq_id)**	**no**
Sequence Table ID. A foreign key in this table.				
vs_lstart	**i**	**4**		**yes**
Left start.				
vs_lend	**i**	**4**		**yes**
Left end.				
vs_rstart	**i**	**4**		**yes**
Right start.				

vs_rend	**i**	**4**		**yes**
Right end.				

D.2.56 The Worklink Table (WL)

This table links entities to worksheets.

The three columns together for a primary key in this table.

Column	Type	Length	Domain	Nulls
wl_ws_id	**c**	**10**	**(ws_id)**	**no**
Worksheet ID. A foreign key in this table.				
wl_etype	**i**	**1**	**(nm_enumbtype)**	**no**
Entity type. This value identifies the type of entity being linked to the worksheet. For example, if the worksheet is linked to a sequence, this column will contain the value corresponding to the entity type "sequence."				
wl_eval	**c**	**10**	**>0**	**no**
Entity value. ID of the entity being linked to a worksheet. For example, if the worksheet belongs with entry 4509, this column will contain 4509.				

A worksheet is a collection on entities (a list of entities and their IDs). A worksheet is an entity itself, therefore, a worksheet may be a collection of worksheets. The description of the worksheet is recorded in the Worksheet Table. Worksheets are linked to people through the WorkPer Table.

D.2.57 The Workper Table (WP)

This table links worksheets to people. The two keys form a primary key in this table.

Column	Type	Length	Domain	Nulls
wp_ws_id	**c**	**10**	**(ws_id)**	**no**
Worksheet Table ID. A foreign key in this table.				
wp_pn_id	**c**	**10**	**(pn_id)**	**no**
Person Table ID. A foreign key in this table.				

A worksheet is a collection of entities (a list of entities and their IDs). A worksheet is an entity itself, therefore, a worksheet may be a collection of worksheets. The description of the worksheet is recorded in the Worksheet Table. Entities are linked to a worksheet through the WorkLink Table.

D.2.58 The Worksheet Table (WS)

This table records worksheets.

Column	Type	Length	Domain	Nulls
ws_id	**c**	**10**		**no**
Worksheet ID. The primary key in this table.				
ws_name	**c**	**20**		**yes**
Worksheet name.				
ws_desc	**c**	**80**		**yes**
Worksheet description.				

A worksheet is a collection of entities (a list of entities and their IDs). A worksheet is an entity itself, therefore, a worksheet may be a collection of worksheets. Entities are linked to a worksheet through the WorkLink Table. Worksheets are linked to people through the WorkPer Table.

D.3 Changes to the Schema

6 July 1992

Added the following keys:

ad_owner, ar_owner, cl_owner, cm_owner, cf_owner, du_owner, ep_owner, en_owner, fo_owner, fq_owner, gn_owner, go_owner, kl_owner, kw_owner, nh_owner, pn_owner, pr_owner, qv_owner, rf_owner, rl_owner, ra_owner, sn_owner, sa_owner, se_owner, sq_owner, sc_owner, tt_owner, sb_text.

These keys were added for off-site users, to establish ownership of the information they enter. The value in the owner fields will often be null.

Off-site users will not have access to the contents of the Submission table. They will have access to the Text table. So, the text of the submission form will now be stored in the Submission table rather than the Text table.

29 January 1992

Added the Refsub Table

Appendix E Satellites

Off-site users: This is the document sent to sites interested in creating a remote database. The information is not relevant to entering or annotating sequence data with AWB.

GenBank staff: This is the document sent to sites interested in creating a remote database.

E.1 Requirements

Requirements and Procedures for the Installation of the Sybase Version of the GenBank Database and Application Software for Satellite Sites.

E.1.1 Hardware and Software Requirements.

A. Sybase Database Requirements -- Sybase Version 4.8 or higher.

B. System Requirements -- Sun UNIX OS 4.0 or higher Ultrix 3.0 or higher

C. Hardware Requirements

1. Database disk size 1.2GB or higher(see A.2. below for partition information).
2. Software and database distribution file space size 300MB (not including Sybase software)
3. Exabyte tape drive(not necessary for distribution but needed for dumping the database)

D. Must have access to internet to ftp files from genome.lanl.gov

E.1.2 Installation Procedures

A. Database Requirements

1. Database named genbank should be larger than 1.2GB; ideal size is over 1.5GB.
2. Database should have four internal segments which can span multiple partitions. The database segments should be on different devices if possible for performance improvements. The four segments are:

'default/system'	segment - 20% of database
'log'	segment - 10% of database
'genbank_noncindex'	segment - 15% of database
'genbank_text'	segment - 55% of database

If necessary, The whole database can be put in one segment; you may sacrifice database performance. For more information on the use of segments, reference the Creating and Using Segments section, page 3-24, of Sybase's System Administration Guide.

The distribution code that will load the satellite database validates the sizes of the database segments to see if the minimum amount of space is availlable for the new database. The minimum amount in megabytes is:

As of Dec. 17, 1991:

default/system segment	265 MB
log segment	100 MB

genbank_noncindex segment	150 MB
genbank_text segment	595 MB

The above values will change as the database continues to grow. Here are the approximate growth patterns for the last year:

default/sytem segment	30 MB
genbank_noncindex segment	25 MB
genbank_text segment	300 MB

The text segment has doubled in the last year. Every indication is that this trend will continue. The actual size of you text segment will be around 40% less than this number because submissions, footnotes, and some comments are not in the satellite databases. Please prepare for this increase.

Using Sybase, create the database in small pieces. First create the database where you want the default/system segment to be, include the log creation. This will guarantee that the system tables are where you want them. Then use alter database to create the rest of the database. Create the two remaining segments and drop that system and default segments from them. This will map your database segments correctly. If you want to put two segments on the same partition, this will be more work and you can contact us directly to find out how to do this. Here is an example of the database creation commands(if you put these commands into a script, remember the 'go' command after each sql command; note also that the create and alter database commands will take on the order of hours to run):

```
create database genbank on genbank_data = ____ (system/default in MB)
alter database genbank on genbank_logs = ____ (log segment in MB)
sp_logdevice genbank,genbank_logs
-or-
create database genbank on genbank_data = ____ (system/default in MB)
log on genbank_logs = ____ (log segment in MB)
alter database genbank on genbank_index = ____ (noncindex segment in MB)
alter database genbank  on  genbank_text1 = ____,(text segment in MB)
genbank_text2 = ____
use genbank
sp_addsegment genbank_noncindex,genbank_index
sp_addsegment genbank_text,genbank_text1
sp_extendsegment genbank_text,genbank_text2

sp_dropsegment 'system',genbank_index
sp_dropsegment 'system',genbank_text1
sp_dropsegment 'system',genbank_text2
```

sp_dropsegment 'default',genbank_index

sp_dropsegment 'default',genbank_text1f

sp_dropsegment 'default',genbank_text2

3. Two logins should be added to the server. Here is the Sybase syntax:

   sp_addlogin genbank,genome,genbank

   sp_addlogin libda,conifer,genbank

4. The user genbank should be made the database owner of the genbank database. Here is the Sybase syntax that should be ran from the genbank database:

   sp_changedbowner genbank

   All additions to the genbank database will be done by installation scripts.

5. The tempdb database should be altered to be at lease 15MB in size to handle large queries. 20 to 25MB would be better if other applications are using the database.

6. Make modifications to the configuration of your server. Use sp_configure to make these changes:

   sp_configure memory,<75% of internal machine memory>

   sp_configure locks,100000

   reconfigure

   <reboot your database server>

   For more information on configure values and the best number to put in the memory reconfiguration, see the section in the System Administration Guide on "Changing the Configuration Variables" starting on page 7-27 in the release 4.2 documentation. Note that memory is in 2Kb pages so the memory entry of 2400 is 4.8MB.

   We give the memory configuration of 75% as a guideline only. Your sybase server when you install it has the minimum setting of 2400 2K pages(or 4.8MB). Everything will work at the initial setting but the database and applications will run faster and do less page swapping if there is more internal memory configured to the dataserver.

7. Satellite servers experience problems handling the large conditional statements while executing sql from the genbank software. Follow the next steps to enlarge one of the dataserver's run-time parameters.

   % /usr/sybase/bin/buildmaster -d/dev/<master_raw_device> -ycstacksz=32768

   This command changes the stack size configuration value for the dataserver. This command is to be run when the dataserver is NOT running. Take the Sybase server down by running the sql command 'shutdown' or kill the dataserver process. Run the buildmaster command. Restart the server and the new value will be used.

E.1.3 Software and System Requirements

1. Disk space requirement is 100MB. This space will be used for software and for update packets awaiting processing.
2. Need one account for a user. (suggested name genbank)

Once the database and user account are setup, notify rds@life.lanl.gov and gmk@life.lanl.gov that you are ready for shipment of database scripts, a test database, and application software.

Also, any feedback on the helpfulness or improvements of this document would be appreciated.

E.2 Installation

Installation guide for the GenBank satellite database.

This document will help you with first time and continued installation of your GenBank satellite database. Please read all the instructions carefully.

First, you must fulfill all the requirements as specified in the GenBank Satellite Requirements Document. You may have obtained a copy when you first showed interest in the satellite program. If you do not have a copy of this document, you may retrieve a copy by using anonymous ftp from genome.lanl.gov. The requirement document is found in retrieve this and other files specified in the document:

```
% ftp genome.lanl.gov
Name (genome:rds): anonymous
Password: <your email address>
ftp> cd <dir>              /* cd to desired directory like pub/doc */
ftp> bin                   /* For executable programs like install_gb */
ftp> get <filename>        /* File you want like requirements.doc */
ftp> bye
```

Once all of the requirements are fulfilled, you may continue to the shipping and loading process. The installation program checks to see that all the requirements are completed before continuing with the procedure.

The next step is to ftp over the file ~ftp/pub/bin/install_gb executable. This is the program that installs the GenBank database and software. Once this file is brought over, use "chmod 700 install_gb" command to make this file executable. Running install_gb will take a long time but if it completes successfully you will have installed the latest public distribution of the GenBank database and have all the software and machinisms in place for automatic updates to take place.

In order to run install_gb, an environment variable "GB_HOME" must be set:

```
setenv GB_HOME /usr/GenBank
```

where /usr/GenBank is the directory in which you want it to run. Install_gb expects to find three sub directories below the directory specified by GB_HOME. They are: bin, tmp, and tables. If they do not exist, the program will create them. The annotators

workbench (AWB) satellite updating software (newsat), and other GenBank software will be placed in the bin directory. The tmp directory is used as a staging area for newsat. The tables directory is where the install_gb program will put the all the bcp files to load the database. This directory must be quite large but if it is more convenient for you, it may be created as a link. Once the install_gb program is finished it will remove the large bcp files and this space will be freed up again.

The install_gb first ftp's a file called "datafile" from GenBank. This file controls the behavior of the program. Included in it are the sql scripts to drop and create the tables in the database, lists of files to transfer to the remote site, lists of tables to fill, and minimum sizes for the database segments and file system.

If it is possible, we strongly recommend that you set up the different database segments with the names and sizes outlined in the requirements document. On some systems, however, this is not possible. If this is the case when you run install_gb you will get a message saying that certain segments in the database are too small or do not exist. In this case you will need to edit the datafile and make the sizes of the segments appropriate for your system. Warning: If you do this it is possible that there will not be enough room in the database to load the tables.

The first thing install_gb program does after bringing over the datafile is to check the physical resources of the database and file system. If they are met it will report this and begin to transfer all of the bcp files from GenBank. This takes some time as the files are quite large. Once the files are transferred they are varified to be the correct size. Next all of the tables are dropped and recreated to insure that any possible schema or data inconsistencies are reconciled. Next the tables are loaded and then indexes are created on the database. Once this has successfully completed the software checks, the database to make sure that it actually has the correct number of rows and indexes.

The GenBank software, including the AWB and flatfile generating program, is automatically brought over by install_gb. If the file newsat.sh does not exist, a shell script and crontab to run it are created. The default time for the job to run is Monday through Friday at 10:00 pm. If you wish to change this, edit the crontab.

Glossary

This glossary contains terms and acronyms used throughout the manual. More specific biological definitions of interest to GenBank annotators can be found in the /usr/gb/doc/annot/tator_dictionary file. Another glossary of biology terms is in the /usr/gb/doc/facts/glossary file.

accession number The unique identifier for each sequence. Accession numbers are issued exclusively by the databases.

alias An abbreviation for a frequently used, longer UNIX command. Aliases are usually defined in a file such as .cshrc but may be entered at the command line.

annotators In-house term for the team that reviews submissions and conducts database maintence.

The Annotator's WorkBench (AWB) A database browsing and editing tool. The primary software tool used by GenBank annotators.

Abstract Syntax Notation (ASN.1) A data representation format. The flatfile used by NCBI.

attribute In this manual attributes are the characteristics of entities that define the entity for the purposes of the database. For example, Title is an attribute of a Paper and E-mail Address is an attribute of a Person.

Authorin A remote database submission generating tool. Generates submitted data in GenBank transaction format. Runs on IBM pc or Macintosh.

Bioserve Group T-10 document retrieval servers. For information, send an E-mail message with only the word "help" to bioserve@life.lanl.gov.

command line Refers to execution of a program from the UNIX prompt.

computation domain A term that refers to the software development and maintenance team at GenBank at Los Alamos.

core file An image file of a program that terminated abnormally. It is an image of the last state of the program. Used for debugging.

coversheet A paper used in submission processing. The coversheet is an aid for classifying, filing, and updating submissions.

.cshrc A file of UNIX commands that are run upon initialization of the C-shell. Along with other "dot" files, it is used to customize the computer workspace environment.

DNA Database of Japan (DDBJ) A member of the International Nucleotide Sequence Database Collaboration, based in Mishima, Japan. Database can be identified by accession numbers with Q through --.

database management system (DBMS) Software that handles the organization and speedy retrieval of data.

dataflow In-house term for the team that conducts initial receipt and processing of submissions, including issuance of accession numbers. An E-mail alias for that group.

directory On a computer, a collection of files or a storage location for files.

Electronic Data Publishing (EDP)

E-mail response (EMR) The procedure whereby copies of reviewed submissions are referred back to the submitter.

European Molecular Biology Laboratory (EMBL) A member of the International Nucleotide Sequence Database Collaboration, based in Heidelberg, Germany. Database can be identified by accession numbers with X through --.

entity In the relational database, an entity is a distinguishable, cohesive collection of data that models a real world object.

FASTA/BLAST Sequence similarity search packages. Both are available by E-mail server.

Feature A biological attribute of a sequence.

Feature Table Document (FTD) Contains the authoritative definition of current Feature table syntax.

Feature Key A specific category of biological attributes of a sequence. In the GenBank database a Feature Key is linked to every Feature entity. Refer to the Feature Table Document for more information on the use of Feature Keys.

field A data value that describes one attribute of an entity. An AWB form object that consists of a prompt and a data area.

flatfile Term historically used to describe the GenBank line-type structure.

flatfile report A formatted report generated from AWB or from the command line.

form A template for the display of related data. In AWB, a form is used to display the information associated with a database entity.

gbarc The archival directory that holds the records of submissions, updates, and response letters.

gb-software An in-house term for the team in charge of programming and software maintenance at GenBank in Los Alamos. A mail alias for that group.

gb-sub The user name of the person to whom submissions are sent. The person who reads the E-mail submissions does so as gb-sub.

GenBank Submission Form An electronic questionnaire containing sequence submission information that is filled out by an author and sent to GenBank. GenBank encourages the use of the Authorin program over this form.

Genetics Computer Group (GCG) Commercial software development group specializing in DNA sequence analysis and manipulation software.

Genome Database (GDB) A database that contains human genome mapping information and was designed to collect, organize, store and distribute data generated by scientists.

gentest A database analogous to GenBank that contains only test data. Used by gb-software to test programs.

Global Regular Expression Parser (grep) A UNIX utility progam used to search files for patterns.

hold date The date on which HUP data may be released.

hold-until-published (HUP) Refers to data submitted in confidence pending publication.

Human Genome Mapping Library (HGML) Earlier version of GDB, now defunct.

Humgene Internal term used to describe manual GenBank-GDB linking procedures. It is also the name of a file that contains the latest complete list of genes and other information downloaded from GDB.

ID number A number assigned to an entity in the database to insure its uniqueness with respect to other entities of the same type. Accession numbers and rf_id numbers are examples. No two Sequences will have the same accesion number and no two Papers will have the same rf_id.

Immunoglobulin (Ig) A type of protein that is the subject of intensive genetic investigation. There are GenBank conventions designed specifically for the annotation of Igs. Refer to the ig documetent in the directory /usr/gb/doc/annot directory. See also, Locus Names on page A-3.

Information Retrieval Workbench (IRX) User query interface to flatfile databases. Used by GenBank annotation to query the flatfile version of GenBank.

Interactive Structured Query Language (ISQL) The Sybase command line SQL utility.

Laboratory Host In AWB this field in the Source form contains the link to the Taxonomy node of the orgranism that served as the host in the laboratory for the organism that was sequenced. It should be different from the Natural host that linked to the Taxonomy node that is linked to the Source.

link A relational database term for a connection between two entities.

.login A file containing initialization processes. It is run once, upon login.

Major Histocompatability Complex (MHC) A class of genes that are the subject of intensive genetic investigation. There are GenBank conventions designed specifcally for the annotation of MHCs. Refer to the mhc documetent in the directory /usr/gb/doc/annot directory. See also, Locus Names on page A-3.

Medline Bibliographic database of all medical and biological journals based at the National Library of Medicine. It is also an alias for accessing the databases (including GenBank) maintained by NCBI.

Medline-Subset A subset of the Medline database that contains articles associated with DNA sequences.

National Center for Biotechnology Information (NCBI) A bioinformatics research/administrative group at the National Library of Medicine.

National Institutes of Health (NIH) The parent organization of the National Library of Medicine.

National Library of Medicine (NLM) The parent organization of the National Center for Biotechnology Information.

Natural host In AWB, the Nat Host field in the Taxonomy form contains the link to the Taxonomy node of the organism that is the normally hosts the organism that has been sequenced.

<no link> In the ASCII AWB, this symbol indicates that there is no connection between the current entity and another. For example, if the <no link> symbol appears in the Publication field of the Paper form, it means that a journal (or other publication) has not been linked to the particular Paper.

off-site user A scientist not on-site at Los Alamos who directly deposits sequence data into the GenBank database with the AWB rather than sending a submission.

Operating System An instruction set for a computer which allows for the interpretation of commands and the execution of programs.

Protein Information Resource (PIR) Term used to describe an international protein sequence collaboration.

reference ID (rf_id) A unique identifier given to each Reference entity in the GenBank database. In AWB it appears at the top of the Paper form and the Reference form.

Reference Status (Status) A statement linked to a Paper/Reference entity that gives information about the work being done with the submission. The list of statuses linked to a Paper provides a history of the a submission's motion through GenBank.

relational database management system (RDBMS) Relational database software that handles organization and speedy retrieval of data.

schema In a relational database, the schema is the layout and definition of the database components. See "The GenBank Database Schema" on page D-1.

segmented entry A set of sequences (and entries) that appear in the genome in a given order and are derived from a single nucleotide molecule.

server An independent computer process that performs some service.

span refers to a particular section of a sequence. In AWB spans are identified by numbers assigned to base pairs.

Sparc systems Most recent class of Sun machines. They use the Sparc Chip architecture.

Sun OS A Sun computer specific operating system.

Structured Query Language (SQL) A high-level language for reltational database systems. ISQL stands for Interactive SQL.

Sybase A relational database software vendor. The GenBank database uses the Sybase relational database system.

Submission Form See GenBank Submission form.

tators An E-mail alias for annotators at GenBank at Los Alamos.

Taxonomy node The entity that represents an organism and its place in the taxanomic classifcation tree used at GenBank. Taxonomy nodes are linked to Sequence entities through Source entities to indicate the biological source of the sequence. Natural Hosts, Laboratory Hosts, and Specific Hosts of the source organism are also Taxonomy nodes.

TREMBL Software package used for importing data from flatfile format used by EMBL.

UNIX An operating system, developed at AT&T Laboratories, used on all GenBank computers.

VAX A Digital Electronics computer line.

virtual feature A feature construct consisting of joins or other operations on a system of components.

virtual sequence A representation of a merger or join of component sequences, that preserves

visual text editor (vi) A UNIX standard screen text editor. Introductory information for using vi can be found in the SUN user manuals and in the book, Introducing the UNIX System by McGilton and Morgan.

Index

Response Page

E-mail comments and suggestions regarding this manual may be sent to gb-manual@t10.lanl.gov or complete this form and mail it to the address on the following page.

Please check the sections of the manual you used.

Chapter 1	General Introduction	_____
Chapter 2	Introduction to the Annotator's WorkBench	_____
Chapter 3	Submission Processing	_____
Chapter 4	Sequence Entry	_____
Chapter 5	Annotation	_____
Chapter 6	Review	_____
Chapter 7	In-House Curation	_____
Appendix A	Conventions	_____
Appendix B	ASCII AWB: Reference	_____
Appendix C	Utilities: Reference	_____
Appendix D	The GenBank Database Schema	_____
Appendix E	Satellites	_____

Were you using the manual to

enter a sequence? _____
learn AWB? _____
learn about GenBank in general? _____

What problems did you encounter with AWB? _____
_____
_____
_____

Please give us your suggestions for improving the documentation. _____
_____
_____
_____

GenBank User Manual

Group T-10
MS K710
Los Alamos National Laboratory
Los Alamos, New Mexico 87545
USA