

PUB-3126

REFERENCE COPY

Does Not

Circulate

Copy 1

Bldg. 50 Library.

SubmitData

User Manual

May 1995

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

SubmitData

User Manual

May 1995

Version 2.01

**Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720**

Supported by the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or The Regents of the University of California.

Lawrence Berkeley Laboratory is an equal opportunity employer.

Table of Contents

Chapter 1. Introduction	1
Overview	1
Version String	2
System Requirements	2
Program Development	2
Chapter 2. Installation	3
Installing VisualWorks	3
Installing SubmitData	3
Installing from a Distributed Image File	3
Installing from Distributed Source Files	3
Working on Different Platforms	4
Field Menus	4
Window Menu	5
Chapter 3. Starting SubmitData	7
Opening the Launcher	7
Opening a Database Module	8
Database Module Windows	9
Help	9
Opening the Transcript Window	10
Preferences	11
Look Selection	11
Host Window System	11
Smalltalk Utilities	12
Quitting	12
Chapter 4. Preparing a Single Submission	13
Transaction Menu: Creating a New Transaction	14
Setting Preferences	14
Opening a New Transaction	14
Filling in the Forms	15
Mandatory and Optional Fields	16

Types of Fields	16
Field Menus	17
Subforms	18
Help with Fields	20
Ending Data Entry	21
Data Entry Errors	21
Working with a Valid Transaction	22
File Menu	22
Saving a Transaction	22
Printing a Transaction	22
Viewing Transactions in the System Transcript	23
Submitting a Single Transaction	23
Installing a Transaction as a Template	23
Clearing a Transaction	23
Chapter 5. Preparing Multiple Submissions	25
Creating a Template	25
Specifying Variables	26
Fields in the Variable Editor	26
Editing a Variable	28
Ending Data Entry	29
Installing the Template	29
Editing a Template	29
Defining a Batch Format	30
Creating a Batch Format	30
Testing a Batch Format	33
Using a Batch Format	34
Making a Batch Submission	34
Making a Dialog Submission	36
Using Custom Batch Formats	37
Chapter 6. Additional Features	39
Editing a Saved Transaction	39
Submitting a Saved Transaction	40
Chapter 7. Database Modules	41
SubmitData\GSDB.....	43
The GSDB Forms	43
Loading AUTHORIN Output	44
Upgrading Existing Templates	44
SubmitData\GDB.....	45
Submission Button	45
Data Submission Forms	46
Index	47

Chapter 1

Introduction

Overview

SubmitData is a convenient user interface that formats data for submission to public databases; for example, the Genome Sequence database (GSDB) or the Genome database (GDB).

When you fill in a database submission form in SubmitData, you work within one of the modules, generating a transaction that matches the particular requirements of that database. Transactions can be viewed, edited, printed, or stored for future use.

SubmitData closely controls the accuracy of the data submission process. Each field is checked for data type, allowed ranges and controlled values, and the user is alerted to any problems.

Submitting large amounts of related data is as easy as selecting a format and designating an input file with SubmitData's template feature. In addition to one-time submissions, users may create templates that can be merged with data files. Template variables are replaced by values in defined columns from the data file. Text files from many popular spreadsheet or database programs can be used with SubmitData. SubmitData\GSDB also accepts files generated by the AUTHORIN database submission tool.

Three types of submissions are possible, depending on the user's requirements:

- Direct submissions.
- Batch submissions, in which templates are used to allow large amounts of data to be submitted in a single session.
- Dialog submissions, in which data are submitted interactively.

Once you have made a transaction, SubmitData lets you send it directly to the database through electronic mail.

SubmitData has been designed to readily adapt to changes in the submission process over time. As databases change their requirements, you will be able to upgrade your copy of SubmitData to keep current. More database modules will also be made available, allowing you to expand the capabilities of the application.

Version String

The version string on your copy of SubmitData reflects the application version number in the first three digits and the database module version number in the last two digits.

System Requirements

The application's graphical user interface and its compatibility with multiple window systems make it convenient for a wide variety of users. SubmitData can adopt the look of any of these window systems. The program consists of a Smalltalk image file (the application) with one or more database modules installed. Each module contains settings for a single database.

To run SubmitData, your computer must be able to run ParcPlace[®] VisualWorks 1.0, which is available for UNIX (Sun OS, Solaris, HP/UX, AIX, or OSF), Microsoft Windows, OS/2, or Macintosh operating systems. At least 5 MB of free disk space and 8 MB of RAM is recommended. For more information, see the installation notes that came with your copy of SubmitData.

Program Development

This program has been developed with support from the U.S. Department of Energy by

Manfred D. Zorn
Software Technologies and Applications Group
Lawrence Berkeley Laboratory
E-mail: SubmitData@lbl.gov

Technical writing by Annette Greiner and David Hand
Technical and Electronic Information Department
Lawrence Berkeley Laboratory

Chapter 2

Installation

Installing VisualWorks

To run SubmitData, you must have VisualWorks installed on your computer. Refer to the installation instructions provided with your copy of ParcPlace\VisualWorks 1.0. After installation you will have a file called st80 on your hard disk.

Installing SubmitData

SubmitData is distributed in one of two ways: as an image file or as a set of source files. Refer to the installation instructions that were provided with the software distribution for further details and to select the installation method.

Installing from a Distributed Image File

Copy the image file, a file named SubmitData or SubmitData.im, to your hard disk. Save a backup copy of the image in a separate place.

Installing from Distributed Source Files

SubmitData can be installed from distributed source files by executing an installation script into a running Smalltalk image.

The database module files are parts of the program that control data input for a specific genetic database. These files are designated by the names of the databases (GSDB, for example). Use the installer program to install each module on your hard disk.

When you install the first one, the SubmitData image file will be created. For details, refer to the installation notes that came with your distribution.

Working on Different Platforms

SubmitData is implemented in ParcPlace VisualWorks 1.0 and runs on a UNIX (Sun OS, Solaris, HP/UX, AIX, or OSF), Microsoft Windows, OS/2, or Macintosh operating system. Because Smalltalk's graphical interface is the same for all four types of systems, most of the instructions in this manual apply to all types of users. Mouse commands, however, differ according to the type of mouse you use. Table 2-1. shows how to issue mouse commands.

Table 2-1. Mouse commands

Mouse Type	To Select	To Pull Down a Field Menu	To Pull Down a Window Menu
Single-button (most Macintosh systems)	click	option-click	command-click
Double-button (most MS Windows-based systems)	click left button	click right button	control-click right button
Triple-button (most UNIX-based systems)	click left button	click middle button	click right button

Before you begin using SubmitData, you will need to understand the general techniques for using your operating system. Make sure that you know how to open and close applications, manipulate windows and files, and use the mouse. See your operating system manual if you need information about these basic procedures.

Field Menus

Most fields in the SubmitData application have an associated menu. When you pull down the menu, you will see options for entering data into the field or for programming a variable. For details about using the field menus, see "Filling in the Forms" in Chapter 4. For information about variables and templates, see "Creating a Template" in Chapter 5.

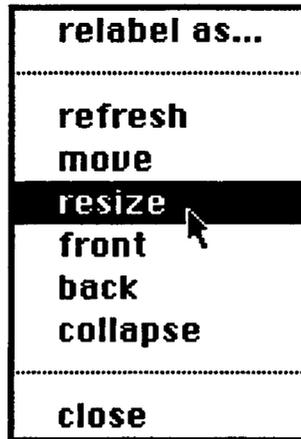


Figure 2-1. Window menu.

Window Menu

The Window menu (Figure 2-1.) lists options for manipulating the currently active window.

Relabel as...

This option lets you give the window a new name by typing the name in a dialog box. Changing SubmitData's window names is not recommended.

Refresh

Choosing this option causes the contents of the window to be redrawn.

Move

Selecting *Move* allows you to relocate the entire window by moving the mouse pointer and clicking in the desired location.

Resize

Resizing lets you change a window's size and shape. The upper left corner remains anchored to its original position, and the lower right corner moves with the mouse pointer. Select *Resize* and move the mouse pointer to a position on the screen. Click there, and the window resizes to that position.

Front or Back

Selecting *Front* moves the active window to the front layer of the screen; choosing *Back* sends it to the back layer.

Collapse

This shrinks the window down to the size of a small title bar or icon (depending on the window system you use), giving you more room to work with other windows.

Close

This option closes the window entirely.

Note: Closing the main window will quit the application.

Chapter 3

Starting SubmitData

Opening the Launcher

To start SubmitData, open the Smalltalk program (st80) and the SubmitData image file as you normally would any application and file on your system.

On UNIX, make sure that the application program (st80) is in your command search path (i.e., \$PATH «variable») and type

```
st80 SubmitData.im
```

With Macintosh or Windows,

double-click on the SubmitData icon.

The ObjectWorks logo appears on the screen while Smalltalk is being loaded. The first SubmitData window is the launcher (Figure 3-1.). From the launcher window, you can choose a database module to work with, you can set preferences, or you can quit the application.

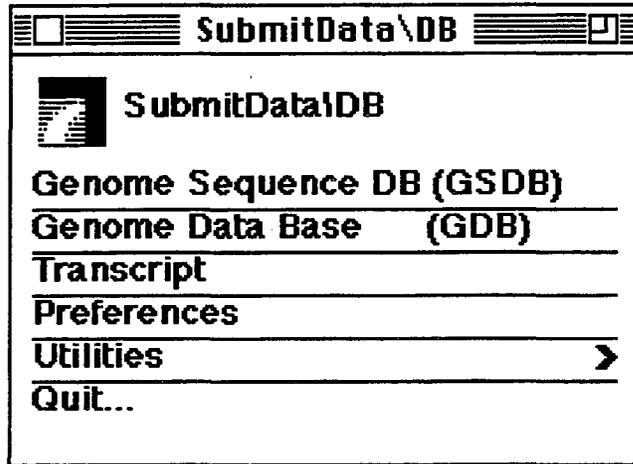


Figure 3-1. SubmitData launcher window.

Opening a Database Module

SubmitData has separate modules for the forms required by various databases. The names of the installed modules are the first items listed in the launcher window (Figure 3-1.). Click on one of these to open the Database Module Window (Figure 3-2.) and to start filling out the forms that correspond to that database.

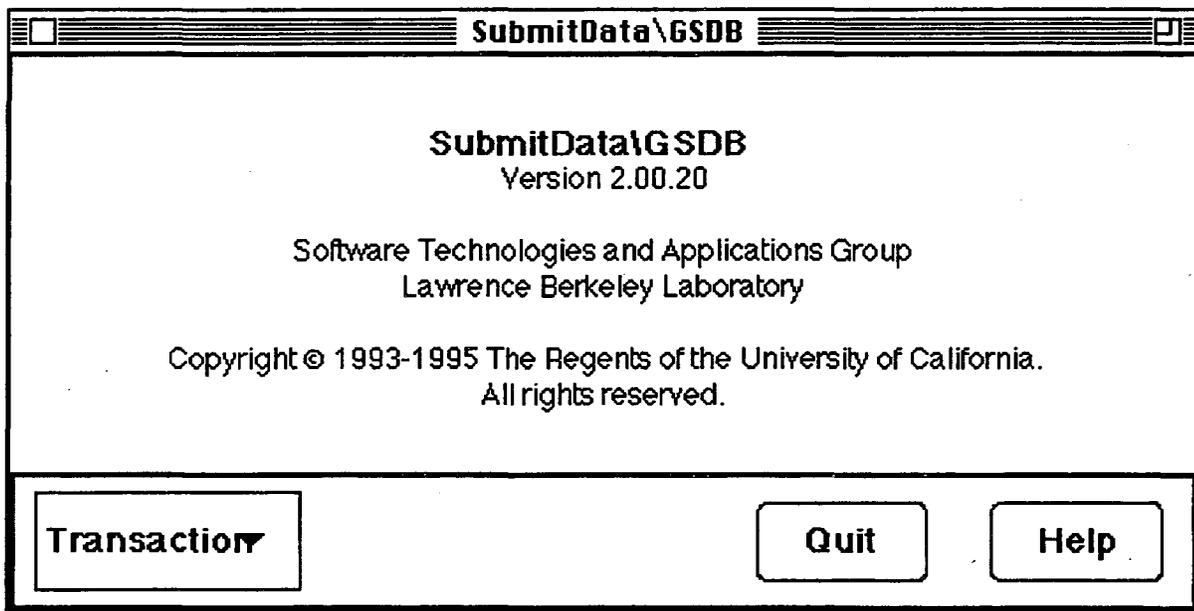


Figure 3-2. GSDB module window.

Database Module Windows

This window has three buttons: *Transaction* (in SubmitData\GSDB) or *Submission* (in SubmitData\GDB), *Quit* and *Help*. *Transaction* (or *Submission*) opens a menu from which you can begin working in the module. *Quit* closes the module and returns you to the launcher. *Help* opens the general on-line help menu for SubmitData.

Help

You can get on-line help by clicking on the *Help* button at the bottom of the module window. This button opens a two-part Help window (Figure 3-3.). Select a topic from a list in the upper part of the window. The lower part of the window displays help information for the selected topic. When you have finished looking at the help text, click on *Close* to return to the window from which you entered the help system.

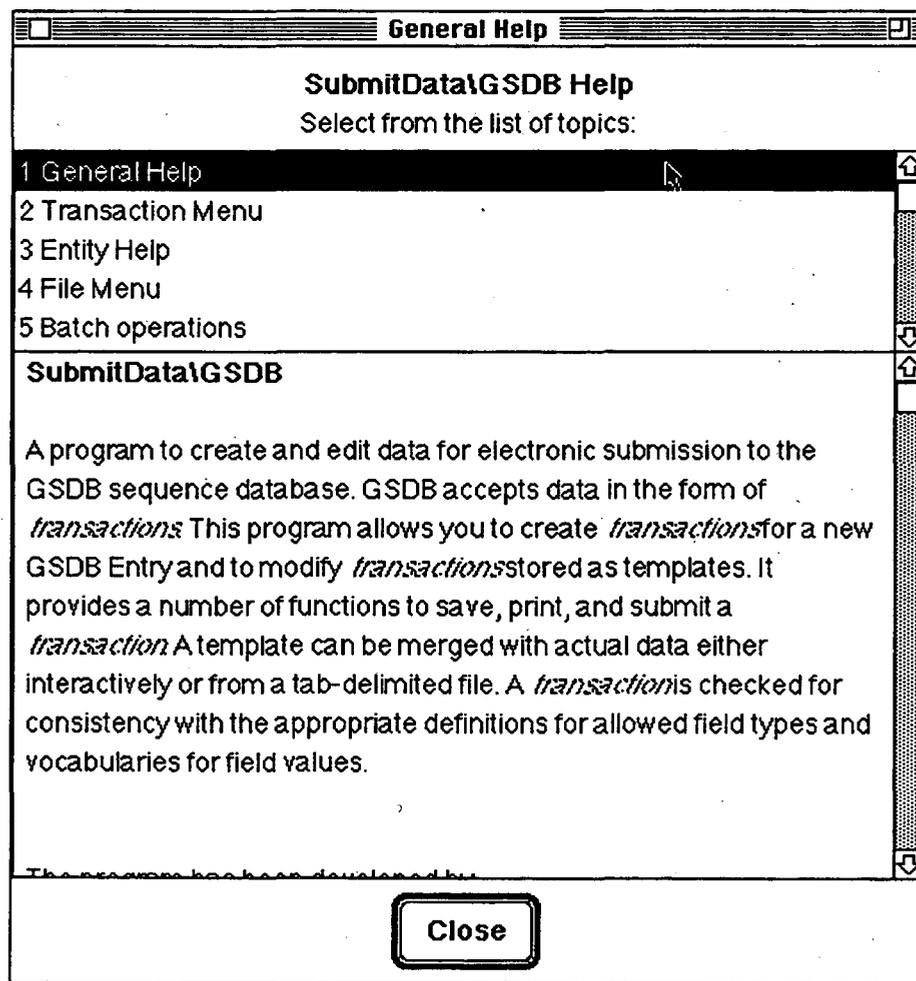


Figure 3-3. Help window for SubmitData\GSDB.

Opening the Transcript Window

The first time you open SubmitData, the *System Transcript* window (Figure 3-4.) appears beside the launcher. This window displays error messages, warnings, and other informational messages. You can also review a submission in the transcript window (see "Viewing Transactions in the System Transcript" in Chapter 4).

The window must be open for the transcripts to be recorded. Closing the window clears the transcript. To open it again, click on *Transcript* in the launcher.

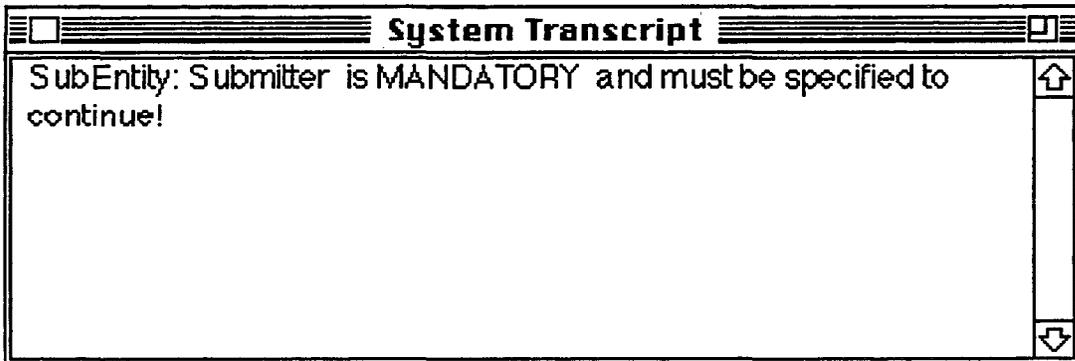


Figure 3-4. System Transcript window.

Preferences

You can set the look of SubmitData's windows to match the window system with which you are most comfortable. Click on *Preferences* in the launcher to see the options (Figure 3-5.).

Look Selection

This option sets the look of SubmitData. Choosing *Default Look* from the pull-down menu gives you the look of Smalltalk. If you choose *Auto Select*, the application will select a look based on your operating system. Clicking in the check box causes the basic tools (the launcher and the system transcript) to also adopt the selected look. Windows that are already open, such as the Preferences window, must be closed and opened again to show the changes.

Host Window System

This option allows you to set the host window system. When you choose *Auto Select* from the pull down menu, the window system is set to the one you are currently running on your computer.

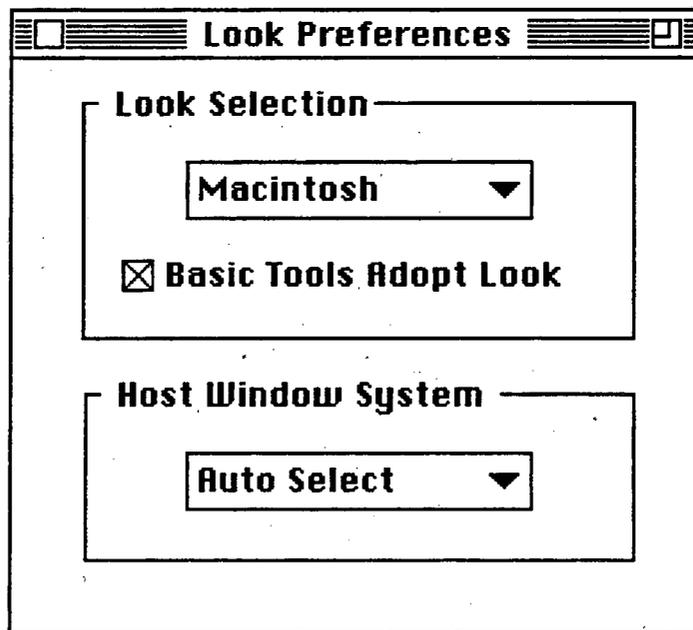


Figure 3-5. Preferences window.

Smalltalk Utilities

The Utilities button opens a menu of Smalltalk utilities. You will not need to use this menu in your everyday work; however, these utilities are important for updating SubmitData or for installing new database modules.

Quitting

Clicking on *Quit* in the launcher opens a small dialog box (Figure 3-6.) that gives you the following options:

- Choose *Quit* to quit SubmitData. If you have created templates or batch formats that still need to be saved, SubmitData will remind you of this when you click on *Quit*.
- Choose *Save then Quit*, to save all your changes before the application closes. You may also use this option to save an incomplete transaction.
- Choose *Cancel* if you do not want to quit.

When you save, you are actually saving the entire SubmitData application. A dialog box will prompt you to enter a name for the file. (The default file name is "SubmitData".) Press return to save without changing the default file name. The application is saved into the same directory or folder where you started it.

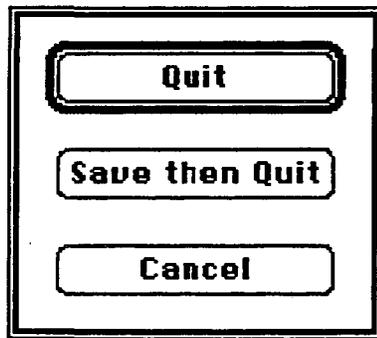


Figure 3-6. Quit dialog box.

Chapter 4

Preparing a Single Submission

<p>Use the following quick procedure to prepare and submit a single valid transaction to a database:</p>
1. Open SubmitData.
2. Select the desired database module in the launcher.
3. Pull down the <i>Transaction</i> menu and select <i>New transaction</i> .
4. Fill out all the forms for the transaction with actual data.
5. Pull down the <i>File</i> menu in the database module window, and select <i>Submit to database...</i>

To make a single submission, you must create a valid transaction (a prepared submission) that contains the data you wish to send. You may create a transaction by opening the appropriate database module and filling in the appropriate fields.

Once you have a valid transaction, you can submit your data in one step. You can also view and print your data, or turn the transaction into a template for making multiple submissions. For details about multiple submissions, see Chapter 5.

Transaction Menu: Creating a New Transaction

Clicking on the *Transaction* button in a module window opens a menu (Figure 4-1.) with options for handling transactions.

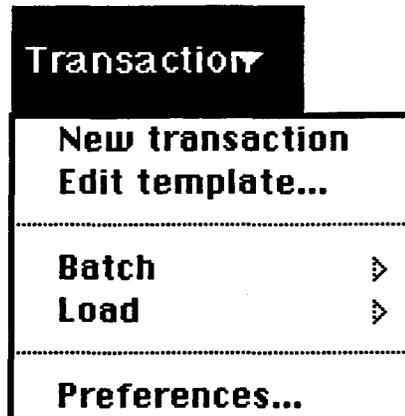


Figure 4-1. Transaction menu.

Setting Preferences

Selecting *Preferences* from the Transaction menu opens the Preferences dialog box. The option *Show empty fields* lets you specify whether or not blank fields are kept as part of the transaction. The default specification depends on the database module. With this setting turned on, you will see the names of all the fields in the form (whether you entered values for them or not) when you view a transaction. Changes to this setting must be made before you create the new transaction. If the option is turned off, only fields with values are shown while editing a template.

Opening a New Transaction

You may create a new transaction by selecting the first option on the Transaction menu: *New transaction*. This opens the module's form window (Figure 4-2.) so that you can begin entering your data.

Filling in the Forms

Featocc

Sequence: subform Featocc

feat desc

left start 0

right start 0

left end 0

right end 0

operator Select an operator:

comp strand Yes No

experimental Yes No

complete 5 Yes No

complete 3 Yes No

codon start Select a codon start:

isconsensus Yes No

replace

featkey featkey

Fill in at least the mandatory (bold) fields and press OK.

OK Cancel Help

Current Form

Parent Form

Menu

Radio Buttons

Scroll Bar

Subform Button

Status Line

Figure 4-2. A sample form.

Each database module contains hierarchical forms and subforms that relate to the type of submission that you are making. The subforms can be reached by selecting special buttons on the forms.

The information in all the forms and subforms needed for a valid submission constitutes a "transaction". Enter your data into the forms by typing in the fields or selecting from available choices.

At the top of every form or subform is a header that shows the name of the parent form (if it has one), a colon, and the form's name. The majority of the form is devoted to fields for data entry. Every field has a label on the left and an input field for values on the right. You can move from field to field (top to bottom) by pressing the tab key. You can also click on any input field to make it active.

Just below the fields is a status line, and at the bottom is a line of buttons. *Cancel* closes the current form or subform without accepting any changes, *Help* opens the on-line help menu, and *OK* accepts the changes and closes the form. The *OK* button is the default.

Mandatory and Optional Fields

Some fields must be filled in to make a valid transaction. The names of mandatory fields are indicated in bold lettering. If you forget to enter a value for a mandatory field, SubmitData will prompt you to do so before it accepts the form as completed.

If a button appears in a mandatory field, you **must** complete all the mandatory fields in the subform to which it leads. If the button occurs in an optional field, you need not complete any of the mandatory fields in the subform. If you want to fill in any of the fields in an optional subform, however, you will need to complete all the mandatory fields in it.

Types of Fields

As you enter data in the forms, you will find several types of fields.

Single-Line Input Fields

Single-line input fields hold one line of text or numbers. They will not accept a carriage return. (A carriage return will trigger the default button at the bottom of the form.) When a field of this type is active, a small cursor appears in it. Most fields require a string value and will accept alphanumeric characters and a few other special characters (i.e., \$, &, +, etc.). Some input fields have other specific data types, which are described in the label of the field menu.

- *Numbers*

Numeric fields accept only digits and may have a default value of zero. Some of these, however, require a value greater than zero. For example, "page start" and "page end" must be positive numbers to be valid. In such cases SubmitData will not accept zero values, even if they are in an optional field. You may delete the zeroes or enter positive numbers.

- *Dates*

Date fields require a date in one of the following formats:

30 April 1982
30-APR-82
April 30, 1982
4/30/82

Any date entered will be converted to a standard format.

- *Phone numbers*

Phone number fields will only accept digits, spaces, hyphens, parentheses, slashes, and plus signs.

Multi-line Text Fields

Fields with scroll bars are multi-line text fields. These fields do accept carriage returns. The number of lines they can hold is limited only by the amount of available memory.

Menu Fields

Menu fields have the appearance and functionality of a standard menu field in the chosen window system. Rather than entering a text string, you select a value from a controlled list of values.

Radio Buttons

Certain fields consist of pair of radio buttons, one for "yes" and one for "no." These radio buttons do not have associated menus.

Subform Buttons

Some fields are shaped like buttons. If you select one of these, a subform opens in a new window.

Automatic Fields

No data input is possible for automatic fields. SubmitData will automatically generate an appropriate value.

Field Menus

Most fields have an attached menu. These let you select from a list of entry options.

Menu Labels

The first selection on any field menu is a menu label, which gives a brief description of the field and the type of values allowed. Selecting it will clear the field.

Cut, Copy, and Paste

Selecting *Cut* from a field menu deletes the information selected in the field and puts it on the clipboard for future use. Choosing *Copy* also places the information on the clipboard but does not delete it. Selecting *Paste* places the contents of the clipboard into the active field.

Read from Data File...

This option reads the contents of a data file into the field. When you select it, SubmitData asks for the name of the file. Type in the name and click on *OK*. The text in the file will immediately appear in the field.

Define a Variable

For information about defining variables, see "Specifying Variables" in Chapter 5.

Cancel and Accept

Some text entry fields have these two options. Choosing *Cancel* cancels the last change you made to the field value. Choosing *Accept* accepts the changes to the current field.

Short-Cut Menu Items

The menus for some single-line input fields provide shortcuts for entering common values. For example, in date fields you can select *Today's Date* to enter the current date. Shortcuts save typing time and prevent data entry errors.

Subforms

Navigating

Forms may contain many layers of subforms. To help you keep track of where you are, the upper left corner of each subform shows the name of the form or subform from which you reached it (the "parent" form). A colon follows the name of the parent form, and the name of the subform follows the colon. If you are filling out a subform called "Person" that you opened from the "Reference" subform, this header will display: "Reference: subform Person". For top-level forms, the word "Form" appears before the colon (Figure 4-2.).

Modifying Subform Entries

A subform menu (Figure 4-3.) will appear if there are already-existing subforms that you have created previously. This menu presents a list of the present subforms and shows the subforms numbered and identified by field values. Select *Cancel* to abort adding a subform, or *new...* to create an entirely new subform. The subform menu helps you keep track of which subform entries you will keep as part of the submission. The subforms attached to the form are enumerated at the top of the menu.

Please choose an existing person:	
1. Smith	
2. Thompson	
.....	
cancel	
new person	

Figure 4-3. Subform menu.

- *Multiple Subforms*

You can easily fill out multiple subforms with SubmitData. You might do this, for example, if you want to cite three authors for a reference. SubmitData lets you fill in three-person subforms quickly.

1. Click on the *Person* button to open the subform menu. Fill in the form for the first person.
2. Click on *OK*. The subform window closes, and you are back to the parent form.
3. Click on the *Person* button again to open the subform menu.
4. Click on the line that represents the first author.
5. In the window that opens, make the changes for the second person. The button line at the bottom of the window will be different from the original one (Figure 4-4.). It will have *New*, *Edit*, and *Delete* buttons.
6. Click on *New* to add the edited subform to the submission as a new person form.

You can also make an additional subform entry from scratch. If you click on *new person* in the subform menu, an empty Person subform will open. Complete it and click *OK* to enter entirely new information for the new person.

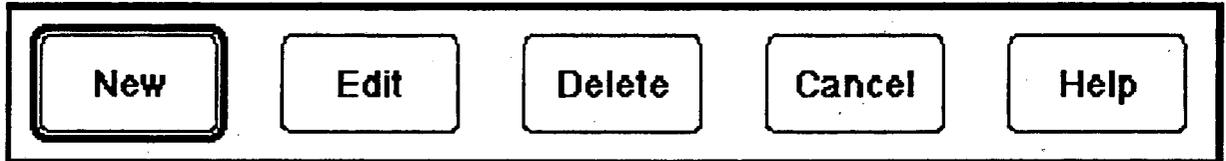


Figure 4-4. Button line when editing a subform.

- *Editing Subforms*

You may use the subform menu to change or delete completed subforms. Click on the line that represents the entry you want to alter. When the window for that entry opens, make your changes and click on *Edit* to accept the alterations and store the subform in lieu of the old subform.

- *Deleting Subform Entries*

To delete a subform entry, click on *Delete*. If you decide not to change the information after all, click on *Cancel*.

Help with Fields

As you work in the form and subform windows, you have access to on-line help about the fields in a particular window. Clicking on the *Help* button opens a menu (Figure 4-5.) listing the fields on a form. This menu also lets you open the Help window. If you choose one of the fields, you will see the field name, its type, and descriptions of the field's purpose (Figure 4-6.). For fields that have entry restrictions, you will also see allowed values and ranges.

Click on *OK* to close the Help window.

Note: You cannot edit forms while the Help window is open.

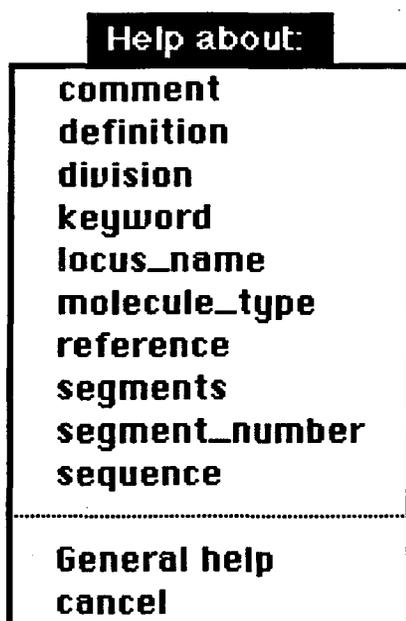


Figure 4-5. Help menu.

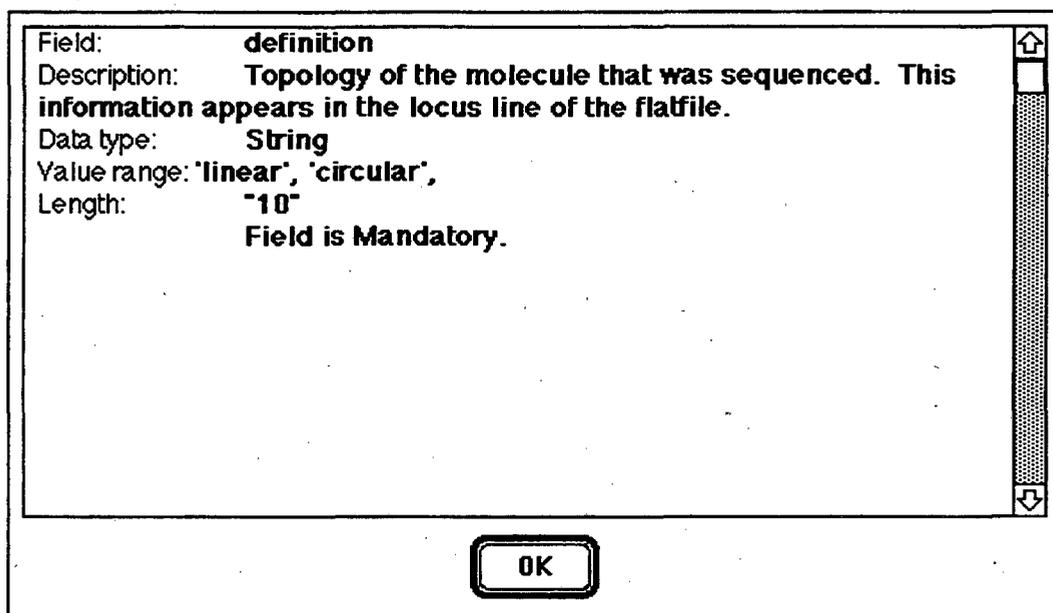


Figure 4-6. Help text for a field.

Ending Data Entry

As you finish filling in the fields for each form or subform, click on *OK*. When you select *OK* from the top-level form, the transaction will be complete. *SubmitData* will prompt you to finish entering data if you have not filled in all the mandatory fields. You will know that the transaction is valid when the application accepts it with no warning messages.

A *File* button appears in the database module window.

Data Entry Errors

Most of *SubmitData*'s form fields contain at least one field that requires data of a specific type. Failure to enter any data or the correct type of data in these fields (e.g., entering letters in a field that requires digits) will result in an error message.

To clear such messages, enter data of the required form and proceed.

Working with a Valid Transaction

File Menu

The *File* menu button (Figure 4-7.) appears in the database module window after a valid transaction has been created. File menu options are useful for sending in submissions, saving files, viewing or printing your data, and working with templates. These options are available only while a valid transaction is current in the application.

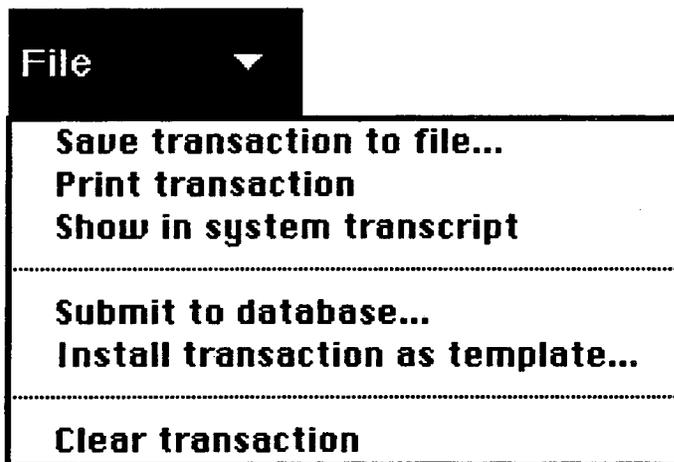


Figure 4-7. File menu.

Saving a Transaction

Select *Save transaction to file...* to save the current transaction in a file on your hard disk. A small dialog box will open, allowing you to enter a file name for the transaction. You can later submit a saved transaction (see Chapter 6 for more information).

Printing a Transaction

You can print a listing of all the entries in the current transaction in the format accepted by the database by choosing *Print transaction* from the file menu. The current transaction is transcribed into PostScript and sent to the printer.

Viewing Transactions in the System Transcript

Select *Show in system transcript* to review the current transaction, and the transaction will be displayed in the System Transcript window in a format accepted by the database. (Make sure that the Transcript window is open. If it isn't, bring Submit-Data's launcher to the front and click on *System Transcript*. Then return to the database module window.)

Submitting a Single Transaction

Select *Submit to database...* if you are ready to send the current transaction to the database. We recommend that you review the transaction by printing it or by viewing it in the system transcript window before you send it.

When you are satisfied that the data are correct, pull down the File menu and select *Submit to database....* You will be asked to verify that you do indeed wish to submit the data. The information will be sent to the database automatically.

Installing a Transaction as a Template

To make any kind of batch submission, you must first install a transaction as a template. A template is a special kind of transaction that has variables defined for field values. You can make a transaction into a template by selecting *Install new template* from the File menu. (For more information about templates, see Chapter 5.)

Clearing a Transaction

Select *Clear transaction* to start over on a new transaction. This clears the current transaction and resets the program. If you did not save the current transaction file, the information in it will be lost when you clear it. A warning dialog will ask for a confirmation in case you did not save or install the transaction.

Chapter 5

Preparing Multiple Submissions

The real strength of SubmitData lies in its efficiency for making multiple submissions. Through the use of templates, you can submit several transactions in little more time than it takes to submit one. Templates save you the trouble of entering duplicate data into more than one transaction. To build multiple submissions, you will need a template and a batch format.

A template is a transaction that has some fields programmed as variables. The template specifies the data that all the transactions share; the remaining changing information is either extracted from a data file (batch submission) or the user is prompted for it (dialog submission). A batch format tells SubmitData how to merge the data.

In a batch submission, a text file containing tab-delimited rows of data is imported. Each row holds the data for one transaction. When merging the data file with a template, SubmitData gathers the information in the columns you specified, builds transactions, and sends it to the appropriate database.

In a dialog submission, the user is prompted to provide values for all the variables. The dialog can be generated from the field name or supplied by user (prompt text in variable definition).

Creating a Template

Since templates are just a special kind of transaction, you create them in the same way as you would create a normal transaction. The only difference is that you in-

sert one or more variables in the data fields. For values that will be the same for all the transactions you plan to make with the template, you enter the value in the form as usual. For variable values, enter the appropriate variable. (If you do not know how to make a transaction, refer to the "Transaction Menu: Creating a New Transaction" and "Filling in the Forms" sections in Chapter 4.)

Specifying Variables

To enter a variable in a transaction, pull down the field menu for the field in which you want to enter it. Select the *Define a variable* option (Figure 5-1.). The Variable Editor (Figure 5-2.) will open. The *Select a variable type* menu shows a list of the types of variables. Select one that meets your needs.

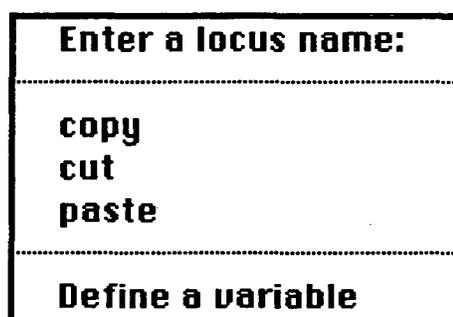


Figure 5-1. The menu for a text entry field.

Fields in the Variable Editor

SubmitData templates can accept seven different kinds of variables. Most of these can have a variable name, a column number, a default value, and a prompt text. The column number refers to a column in the data file for batch submissions. You need not define column numbers when you fill in the variables, although it is recommended. (This can also be done when you create a batch format.) Name **all** variables: this makes it easier to keep track of your variables and allows you to reuse them. Naming variables allows you to enter multiple variables in one field. The values in the "default" and "prompt text" fields are used to construct the dialog (Figure 5-7.) during a dialog submission.

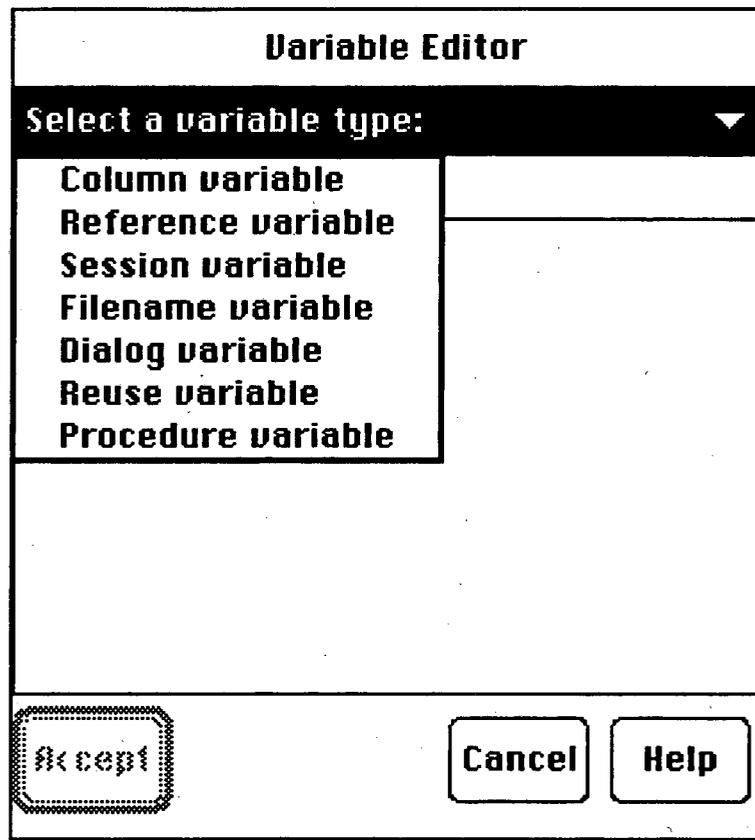


Figure 5-2. Variable editor.

Variable Types

- *Column Variables*
Selecting a column variable tells the application to read data from a column in a data file. In dialog submissions, SubmitData treats column variables as if they were dialog variables. The prompt text default for the "Column" field is "0".
- *Reference Variables*
A reference variable is one that takes its value from another field. The referenced field can be any other field (e.g., variable, automatic, constant). You select the form and specify the field in the form from the pull-down menus. The prompt text default for the "Form" field is the name of the current form.
- *Session Variables*
With a session variable, you can specify a value that will be the same for all the transactions in a submission session. Session variables are constant for one batch or dialog submission. At the beginning of the submission run,

SubmitData will prompt you to enter the value for the session variable. There is no prompt text default.

- *.Filename Variables*
If you choose a filename variable, SubmitData will interpret text in a column of the data file as a file name rather than as data. The entire contents of the named file will be read and will become the field value. In dialog submissions, SubmitData prompts you for the file name. The prompt text default for the "Column" field is "0".
- *Dialog Variables*
Dialog variables let you enter a new value for each record interactively, even in a batch submission. Using a dialog variable tells SubmitData to prompt you for the values when you make a submission. There is no prompt text default.
- *Reuse Variables*
With this option, you can reuse a named variable of any other type. The value of a variable is set dependent on its original type (e.g., from a column for a column variable; by the user for a dialog variable). Select the variable from a list of names on the pull down menu. The prompt text default for the "Variable name" field is whatever variable has been defined.
- *Procedure Variables*
Procedure variables calculate the value for a variable using predefined procedures. For example, you can insert the date on the day you run the batch submission by selecting the *Today's Date* procedure. The number and kind of predefined procedure may vary for different database modules. There is no prompt text default.

Using Multiple Variables

As long as you name your variables, you can use more than one in a field. Simply specify the first variable, close the Variable Editor, open it again, and specify the next variable. If you like, you can also use fixed text that will appear along with the variables. Type the fixed text and specify your variables in the order in which you want them to appear. Be careful not to delete or alter the variable definitions.

Editing a Variable

Select the variable definition (i.e., text between and including « ») and open the Variable Editor. If there is only a single variable offered in the field, you may select the entire contents of the field and open the Variable Editor.

Ending Data Entry

As with a regular transaction, you finish creating a template by selecting *OK* from the top-level form. *SubmitData* will prompt you to finish entering data if you have not filled in all the mandatory fields. The *File* button also appears in the database module window. Through the *File* menu, you can view or print the new template or save it as a file.

Installing the Template

You can use any transaction that has variables defined as a template. To install the current transaction as a template, click on the *File* button and select *Install transaction as template...* This adds the template to the list in the *Edit Template* menu, but it does not save it as part of the application. You must choose *Save then Quit* when you quit *SubmitData* for the new template to be saved. Once you have installed the template, choose *Clear transaction* from the *File* menu. This will restore the *Transaction* menu.

Editing a Template

You can edit an installed template by selecting *Edit Template...* from the *Transaction* menu. The procedure is as follows:

1. When you make this selection, a menu listing the existing templates will appear. Select the one that you want to make changes to.
2. Once you make a selection, *SubmitData* checks the template for syntactic errors and displays it in a new window. You can make any alterations you like by filling in the changes in the forms. If *Show empty fields* is turned off in *Preferences*, however, you will only see the fields that already contain values. To see and add new values to fields that do not have values, select *Show empty fields* in the *Preferences* window before starting to edit the template.
3. When you are done, select *OK* on the top-level form. The *File* button appears in the module window. To save your changes, you must install the template again.
4. Pull down the *File* menu and select *Install transaction as template....* You can install it under a new name to make a new template based on the old one, or you can install it under the original name to replace the old one.

Defining a Batch Format

Once you have installed a template, you can pair it with a data file to make a batch submission. You set up this pairing by defining a batch format. The batch format tells SubmitData which columns of the data file match up with fields in the template. If you are making a dialog submission, you will still need to make a batch format. It will tell SubmitData which template to use and prepare it to accept user input.

Creating a Batch Format

To create a batch format, the procedure is as follows:

1. Pull down the Transaction menu and select *Batch*. Then select *Define batch format* from the submenu (Figure 5-3.). The SubmitData\Batch window opens (Figure 5-4.).
2. Near the top is a pull-down menu listing the available templates. The "Batch format name" field menu lists already-installed batch formats. When you select any of these, a list of the column variables in that template appears in the middle of the window (Figure 5-5.). It shows the name of the form, the name of each field, and the column number and name of each variable (if you specified these last two in the template). If you already specified the column number when you defined the column variable, this list can serve as a summary overview of the columns and variables.

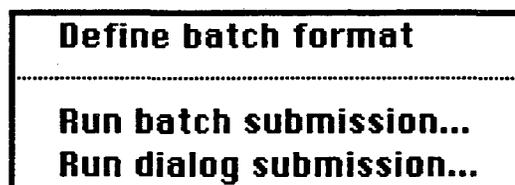


Figure 5-3. Batch submenu.

SubmitData\Batch

Define a Batch Format
 Select from the menu of templates and specify a TAB delimited column for each variable:

Select a Template:

Form name	Field name	Column	Name
<div style="text-align: center;">↑</div>			

Enter column number:

Batch format name:

OK Cancel Help

Figure 5-4. SubmitData\Batch window.

Batch Formats for Batch Submissions

For batch submissions, you must specify a column number for every column variable unless you already assigned a column number when you defined the variable.

1. Click on a line in the table to select that variable, then type the number of the column in the data file that corresponds to it.
2. The number appears in the "Enter column number" field below. Now press return or select another row in the table. The column number in the list changes to match the number you just typed.
3. Continue down the list until you have assigned a column number for each column variable in the template (Figure 5-5.).
4. Next, type a name for the batch format and click on OK. A message will appear to tell you that the new batch format has been added to the list of available batch formats and different buttons appear on the bottom of the window.

SubmitData\Batch

Define a Batch Format
 Select from the menu of templates and specify a TAB delimited column for each variable:

Select a Template: **badDOG**

Form name	Field name	Column	Name
Entry	locus_name	2	spe
Entry	locus_name	1	p1 num
Entry	locus_name	13	index
Comment	text	14	overlap
Sequence	sequence	10	seq
Sequence	sequence_description	7	species
Sequence	sequence_description	2	p1 name
Sequence	sequence_description	12	dog
Sequence	sequence_description	3	p1 DSnum
Source	library	6	vector
Source	library	7	vector

Enter column number:

Batch format name:

Edit **Test** **Delete** **Quit** **Help**

Figure 5-5. SubmitData\Batch window with a format loaded.

Batch Formats for Dialog Submissions

When you make a batch format for a dialog submission, you need not enter any column information. If you are using a template that has column variables, the information about these variables will appear in the SubmitData\Batch window. Since a dialog submission does not use a data file, however, the column numbers will have no effect on your submission, and SubmitData will ignore them.

1. Choose the correct template and specify a name for the batch format. When you are finished, click on *OK*.
2. A message will appear to tell you that the new batch format has been added to the list of available batch formats. Because SubmitData can ignore column designations in dialog submissions, you can use the same batch format to make a dialog as well as batch submission. If you plan to take advantage of this flexibility, make sure you fill in the column numbers in the batch format.

Editing Batch Formats

Once your new batch format has been created, new buttons appear in the definition window. You can alter the batch format by making changes and clicking on *Edit*. If you want to remove a batch format, select *Delete*. You can also edit other batch formats by selecting them from the "Batch format name" field menu. If you want to close the SubmitData\Batch window without making further changes, choose *Quit*. SubmitData will ask whether you want to save the new batch format. Before you quit, however, you should test the batch format with the corresponding data file.

Testing a Batch Format

You use the *Test* button in the SubmitData\Batch window to test a new format whenever you create one.

1. When you select *Test*, SubmitData asks you for the name of the data file. Type it in and click on *OK*. (You may use an asterisk as a "wild card" character to get a list of file names with the text you remember. For example, typing **.txt* will bring up a list of files with the *txt* extension.)
2. Next, the program asks you to name a new file that will contain the results of the test submission. Type a new name or leave the default file name, and click on *OK* again. SubmitData then reads the first row of data from the data file and applies it to the template according to the instructions in the batch format. It shows you the results for this first transaction in a new Batch Transcript window.
3. Scroll through the text of the transcript to see that the field values make sense, then close the window. SubmitData returns you to the

SubmitData\Batch window. Here you can edit the batch format, delete it, or return to the database dialog box. If you want to accept the batch format without further changes, click on *Quit*. SubmitData will prompt you to save the batch format.

Using a Batch Format

Making a Batch Submission

In a batch submission, you merge a template with a data file according to the instructions in a batch format. So, once you've defined the template and the batch format, the only remaining preparation is to create the data file.

The Data File

The data file holds the information that will be transferred into multiple transactions by the batch submission process. It must be a tab-delimited text file with a carriage return after each row of data. Text files generated by most spreadsheet or database programs are tab-delimited. In general, if your spreadsheet or database program allows you to save files as text, those text files will likely be compatible with SubmitData.

Even with a simple text editor, you can easily make a compatible data file. Type your data in rows with a tab after each entry and a carriage return after every row. Each row in the data file must contain the variable data for one transaction. For each field in the submission form that contains a column variable, there must be a corresponding column in the data file. When you enter your data as straight text, the columns probably will not line up. Do not try to align them; the number of tabs before a given value, not the visual alignment, defines the column.

Running a Batch Submission

If you have a data file, a template, and a tested batch format, you are ready to make a batch submission. Use the following procedure:

1. Go to the Transaction menu and choose *Batch*. From the submenu (Figure 5-3.), select *Run batch submission...*
2. SubmitData will then ask you to choose a batch format and to enter the name of the data file. Type it in the box and select *OK*.
3. Next, the application asks for the name of a file in which to save the transactions. The default name is the name of the data file plus the file extension `transaction`. Click on *OK* when you have chosen a file name. If you selected a file that already exists, the application will warn

you. You can click on *Yes* to use it anyway and overwrite the old file. Click on *No* to give a different file name.

4. SubmitData begins generating the new transactions. If you have a session variable in the template, it will prompt you for its value. A Batch Transcript window opens up to show you the transactions as SubmitData creates them (Figure 5-6.). When merging is complete, you will have the option to submit the transactions to the appropriate database. Whether or not you submit the transactions, the data will be saved in the file. You can view the file in a text editing program.
5. The batch transcript provides a report of the batch submission. You can save, print, or clear the transcript by selecting the appropriate button.

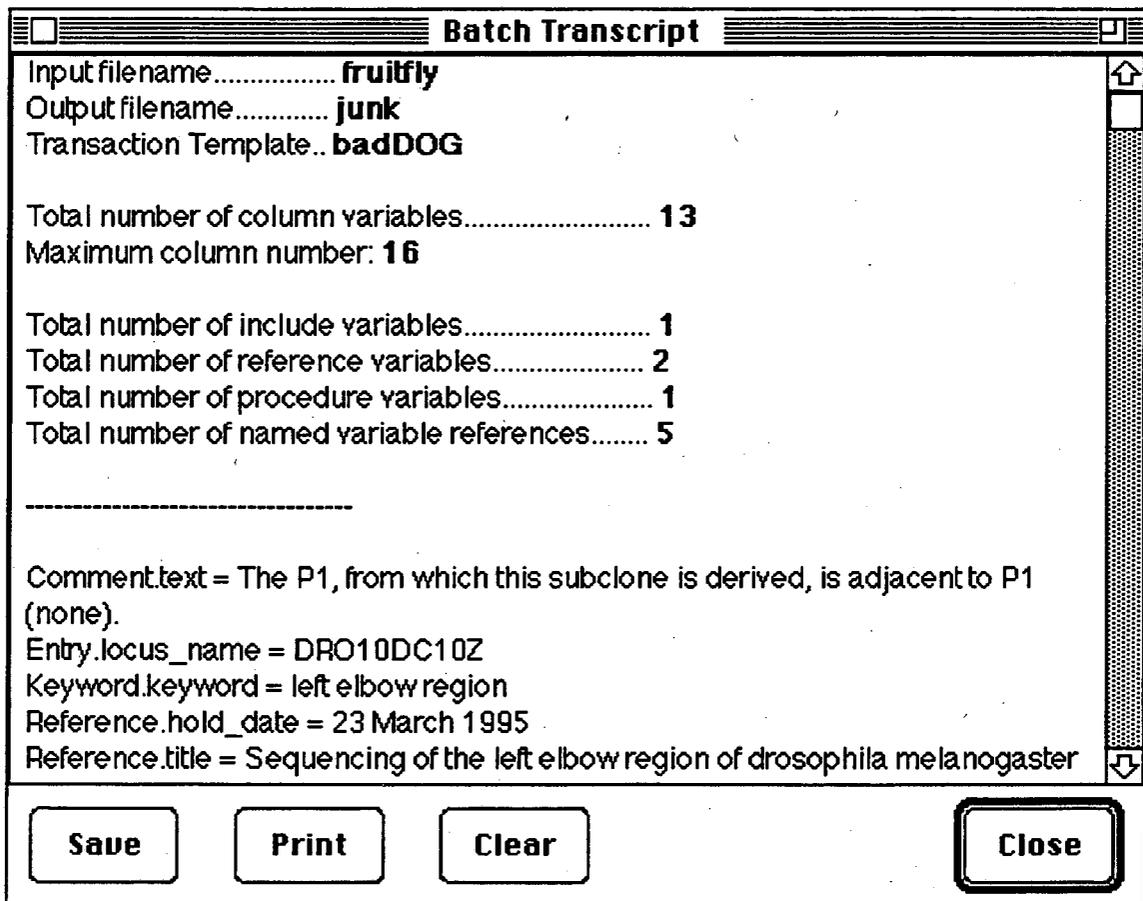


Figure 5-6. Batch transcript window.

Making a Dialog Submission

For a dialog submission, as for a batch submission, you must make a template and a batch format. When you have made both of these, you are ready to run the dialog submission.

Running the Dialog Submission

To run the dialog submission, use the following procedure:

1. Pull down the File menu and select *Batch*. Choose *Run dialog submission...* from the submenu (Figure 5-3.). SubmitData will ask you to choose from a list of batch files.
2. Click on the one that you set up for the current batch of data.
3. Next, the application will ask for the name of a file in which to save the transactions. The default name is the name of the data file plus the file extension *transaction*. Click on *OK* when you are satisfied with the file name. If you selected a file that already exists, the application will warn you of this. You can click on *Yes* to use it anyway and overwrite the old file. Click on *No* to type in a different file name.
4. The next prompt that appears will be one of the prompts that you chose when you built your template. SubmitData will take you through the batch submission, transaction by transaction, prompting you for the data as it needs it (Figure 5-7.).
5. At the end of each transaction, a dialog box will appear, allowing you to review the data for the last transaction and asking what you want to do next. Select *Redo* if you want to enter the data for the last transaction again (to make a correction). Click on *Accept* to add the data you just entered to the submission file and begin entering data for the next transaction.
6. Enter the data for as many transactions as you want to make in this one batch. When you are done, click on *Done* in the dialog box to accept the last transaction and end the data entry.
7. When SubmitData is done processing the information, it will give you the choice to submit the data or not. Whichever option you choose, the application will save the data under the file name you specified.
8. As with a batch submission, SubmitData generates a batch transcript (Figure 5-6.). You can save, print, or clear the transcript by selecting the appropriate button.

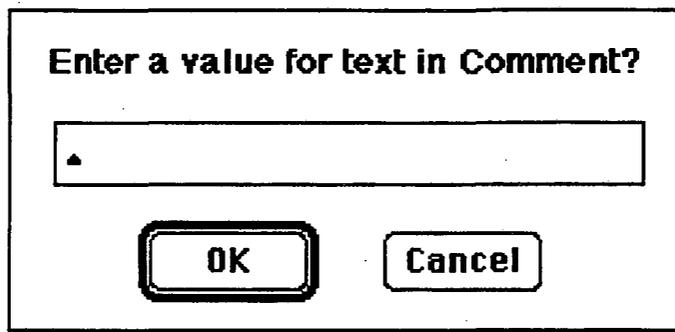


Figure 5-7. A prompt in a dialog submission.

Using Custom Batch Formats

SubmitData can also operate with "custom batch formats." These are Smalltalk routines used to solve specific problems for certain users. If you face a submission problem that you believe SubmitData does not handle, you may consider building a custom format. For details, contact

Manfred D. Zorn
Software Technologies and Applications Group
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720
E-mail: SubmitData@lbl.gov

Chapter 6

Additional Features

The Transaction menu has a submenu for loading different kinds of files into SubmitData. You can edit saved transaction files, submit saved transactions, or update the application's forms.

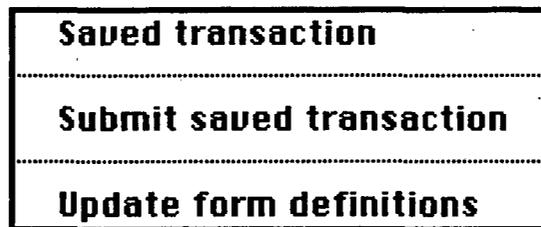


Figure 6-1. Load submenu.

Editing a Saved Transaction

If you wish, you can load a saved transaction into SubmitData to edit it. Select *Load* from the Transaction menu. In the submenu (Figure 6-1.), select *Saved transaction*. A dialog box will appear, asking for a file name. Type the name of the saved transaction, then click on *OK*. The saved transaction becomes the current transaction, which you can use as you would any valid transaction. For example, you might want to save the edited version under a different name to create a new transaction.

Submitting a Saved Transaction

You can use the Load menu to quickly submit a saved transaction to a database. Select *Load* from the Transaction menu and choose *Submit saved transaction...* from the Load submenu. SubmitData will prompt you for the name of the file and ask you to verify that you want to submit it.

Updating Form Definitions

Since genome databases often change their specifications for submissions, a function is provided to update SubmitData's forms in accordance with those changes. The *Update form definitions* option in the Load submenu serves this purpose. Future updates will include instructions for using this function.

Chapter 7

Database Modules

SubmitData works with a number of database modules. Although they are similar overall, there are a few specific differences. The following pages describe the unique features of each module. As you receive new modules, you can insert the documentation for them in this chapter.

SubmitData \ GSDB

SubmitData \ GSDB is the database module for the Genome Sequence database. A GSDB-specific help file is added to SubmitData when you install it.

The GSDB Forms

The top-level form in SubmitData \ GSDB is called "Entry." When you begin to fill in the GSDB forms, the words "Form: Entry" appear in the upper left-hand corner of the window. All other forms in this module are subforms that branch from the Entry form.

Form: Entry

locus name

division **Select a division:** ▼

molecule type **Select a molecule type:** ▼

definition **Select a definition:** ▼

number

segments

sequence

reference

comment

keyword

Fill in at least the mandatory (bold) fields and press OK.

OK Cancel Help

Figure GSDB-1 Top-level form for SubmitData \ GSDB.

Saved transaction AuthorIn output
Submit saved transaction
Update form definitions Upgrade existing template

Figure GSDB-2 Load submenu in SubmitData\GSDB.

Loading AUTHORIN Output

In SubmitData\GSDB, you can load data from the AUTHORIN database submission program into SubmitData. To do so, select *Load* from the Transaction menu (Figure GSDB-2) and choose *AuthorIn Output* from the submenu. In the dialog box that appears, type the name of the AUTHORIN file and click on OK. The AUTHORIN data is loaded as a SubmitData transaction. It becomes the current transaction, which you can use as you would any complete, valid transaction.

The AUTHORIN software is available from:

National Center for Biotechnology Information
National Library of Medicine
Bethesda, MD

Upgrading Existing Templates

This option lets you upgrade the templates already in SubmitData\GSDB to correspond with the most current version of the module. You can reach it by selecting *Load* from the Transaction menu and choosing *Upgrade existing template* from the submenu. SubmitData\GSDB template upgrades will be distributed with instructions for using this function.

SubmitData \ GDB

SubmitData \ GDB is the database module for the Genome database. The database module gives you a choice of data submission forms to match the type of data you wish to submit. A GDB-specific help file is added to SubmitData when you install the SubmitData \ GDB database module.

Submission Button

The leftmost button in the SubmitData \ GDB module window is called *Submission*. Its function is essentially the same as that of the *Transaction* button in GSDB. All procedures that involve the *Transaction* button in GSDB are accomplished with the *Submission* button in GDB.



Figure GDB-1 Submission menu.

Data Submission Forms

Since the SubmitData\GDB module requires you to choose the type of submission, selecting *New submission* from the Submission menu presents a list of data submission forms. When you select one, the proper form will open in a new window.

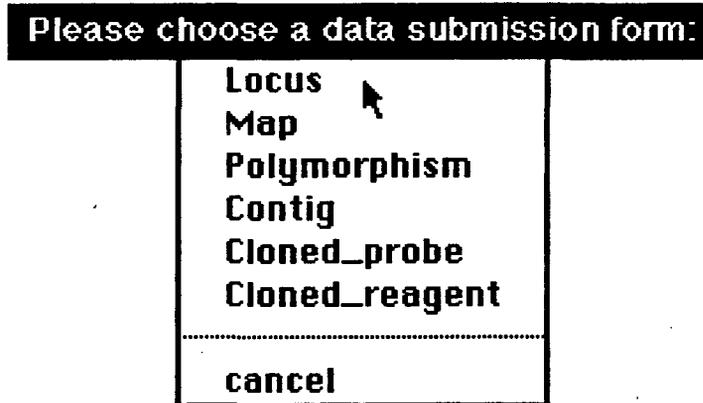


Figure GDB-2 GDB data submission form menu.

Index

A

AUTHORIN 44
Automatic Fields 17

B

Back 5
basic tools 11
Batch Format 30
batch format 25
 creating 30
 custom 37
 editing 33
 testing 33
batch submission 25, 34
 dialog 31, 33
 running 34
Batch Transcript 35

C

Cancel 12
Clearing a Transaction 23
clearing fields 17
clipboard 17
Closing windows 6
Collapsing windows 6
Column variables 27, 30
Copy 17
Custom Batch Formats 37
Cut 17

D

data file 32, 34
 in a field 17
 making from text 34

Database module
 opening 8
database module 2
Dates 16
Default Look 11
dialog submission 25, 36
 batch format 33
 running 36
Dialog Variables 28

E

Editing a Template 29
Editing a Variable 28
Entry form 43
error checking 29
error messages 10
errors
 data entry 21

F

Field Menus 4, 17
fields
 clearing 17
 mandatory 16
 menu 17
 multi-line text 17
 optional 16
 ranges 20
 single-line input 16
 subform buttons 17
 types 16
 values 20
File Menu 22
Filename Variables 28
Form definitions
 updating 40

Forms 15
Front 5

H

help 9
 with fields 20
help menu 20

I

Installing 3
installing
 SubmitData 3
 templates 23, 29
 VisualWorks 3

L

Launcher 7
launcher 8, 10, 11, 12
loading 39
 AUTHORIN output 44
 form definitions 40
 saved files 39
 templates 44
Look Selection 11

M

Menu Fields 17
Menus
 field 4, 17
 window 5
Mouse commands 4
Moving windows 5
Multiple Variables 28

N

Numbers 16

O

operating system 4
Overview 1

P

Paste 17
Phone numbers 16
Platforms 4
preferences
 look 11
Printing a Transaction 22
Procedure Variables 28

Q

Quit 12
Quitting 12

R

Radio Buttons 17
Reference Variables 27
Refreshing windows 5
Relabeling windows 5
Resizing windows 5
Reuse Variables 28

S

Save then Quit, 12
Saving
 batch formats 33
 templates 29
 transactions 22
Selection 4
Session variables 27, 35
Short-Cut Menu Items 18
Show empty fields 14
Smalltalk 7, 37
Source files 3
st80 7

- Starting up 7
- Subform Buttons 17
- subform menu 18, 19
- subforms 18
 - modifying entries 18
 - navigating 18
 - opening 17
- Submission button 45
- submission forms 46
- submissions
 - multiple 25
 - single 13
- SubmitDataGDB 45
- SubmitDataGSDB 43
- submitting
 - single transaction 23
- System Requirements 2
- System Transcript 23

T

- template 25
 - creating 25
 - editing 29
 - installing 23, 29
 - saving 29
 - upgrading 44
- transaction 15
 - clearing 23
 - editing saved 39
 - installing as a template 23
 - loading 21
 - printing 22
 - saving 22
 - template 25
 - valid 16, 21, 22
 - viewing 23
- Transaction Menu
 - Creating a new transaction 14
 - Opening a new transaction 14
- Transaction Menu preferences
 - show empty fields 14

U

- upgrading templates 44
- utilities
 - Smalltalk 12

V

- Variable Editor 26
- Variables 26
- version string 2
- Viewing Transactions 23

W

- Window menu 4, 5
- Window system 11
- windows
 - manipulating 5

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720