EPA/600/R-94/034
March 1994

GENETOX MANAGER SYSTEM - PERSONAL COMPUTER SOFTWARE PROGRAMMER'S GUIDE

Prepared by:
Larry D. Claxton, Ph.D. and John Creason, Ph.D.
Health Effects Research Laboratory
U.S. Environmental Protection Agency
Research Triangle Park
North Carolina 27711

and

Joe Nader, Way Poteat, and John Orr
Computer Sciences Corporation
4401 Alexander Drive
Research Triangle Park
North Carolina 27711

Prepared Under
Contract # 68-01-7365
Delivery Order # 220-151
Task Officer: Larry D. Claxton

June 1993
Version 2.21

Printed on Recycled Paper

PREFACE

This programmer's guide is the reference manual for describing software system and programs used in constructing personal computer portion of the EPA Genetox Manager System.

GeneTox Manager is a data capture, data management, and statistical analysis program used for microbial mutagenicity data. Its main purpose is to provide a homogeneous environment for the collection, organization, and analysis of data generated in the laboratory while also supporting a quality assurance program. The complete system consists of both a personal computer (PC) system and a minicomputer (VAX) system, although the PC system can be used as a stand-alone system. The system controls the data collection and management through a user-friendly menu driven system that requests specific information and provides uniform methods for data and information entry. GeneTox Manager is capable of performing specialized statistical analysis of the data and creating tabular and graphical summaries. Two versions of the PC system exist: a stand alone PC only version, and a VAX version. The VAX version of the system is designed to function with VAX FOCUS databases so that the VAX can be used for long-term storage, archiving of files, and the analysis of large groups of data. Both PC versions are described in this guide.

We only make a few REQUESTS OF THE USERS. First, we request that you provide to the creators of this software proper acknowledgement. [See the title page to the software and to the User's Guide. Also in the near future, we hope to have references that you may cite.] Secondly, realizing that the software is provided free of charge, please understand that we cannot service this software. [Because the source code is available, many consultants and companies that use Clipper® could provide additional help and programming options for you.] Follow the installation and distribution instructions very carefully. If you have problems, first read the section of the User's and Programmer's Guides that deal with problems before contacting us. Most problems that we have seen are due to either not following installation instructions completely (especially in regard to autoexec.bat and config.sys files) or to memory allocation problems. Most all of the memory allocation problems are due to other "resident programs" - please see the guides. If you receive and especially if you distribute the software, be sure to do that in the proper manner. Be sure that you understand the statistics and proper interpretation of the statistical programs. Also understand that statistics do not replace sound scientific judgment and that statistics on a single test does not replace replication of an experiment. Lastly, be sure to read the "readme.1st" file.

We sincerely hope that this software will be useful to you and will aid you in your scientific endeavors.

Acknowledgements

We would like to acknowledge those individuals who used prototype versions of this software and gave us useful, productive comments. Without their suggestions, this product could have not been done. In this regard, we give our thanks to Sarah Warren, Erica Perry, Virginia Houk, Sam Schalkowski, Mary Beth Lawhorn, and Maria Taylor.

We also want to thank those who gave us permission to use the statistical analysis programs that they developed. These include Dr. L. Berstein, Dr. D. Krewski, and Mr. A. Stead. If you use information from these statistical programs, we hope that you will give proper acknowledgement to the authors of the statistical manuscripts.

We also want to thank Mr. Ken Laws for providing administrative support that made this project possible, and Drs. Lawrence W. Reiter and Michael D. Waters for creating the scientific support that made our endeavors a joy. Thanks also go to Mr. Ken Mullins, Dr. David Svendsgaard, and Dr. Bernard Most for review of this document.

Notice

The software and the user documentation are provided "as is". CSC and EPA make no warranty, express or implied with respect to the software and/or user documentation and specifically disclaims the implied warranties of merchantability and fitness for a particular purpose. EPA and CSC do not warrant that the software will be uninterrupted and/or error free. You are solely responsible for the selection of the software to achieve your intended results and for the results actually obtained.

This document has been reviewed in accordance with U.S. Environmental Protection Agency policy and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

Contents

Preface i

Acknowledgements ii

Notice ii

Chapter 1: Introduction 1

Chapter 2: Working With GeneTox Manager 3
- System Language 3
- Clipper 5.01 and OverLay() 4
- Separate EXEs 4
- Additional System EXEs 5
- Database Keys 6
- GeneTox Load/Install 6
- Data Entry 7
- Reports 7
- Data Status Flags 8
- Transfer Files 8
- Upload/Download Procedures 8
- Data Flow Diagrams 9
  - Upload/Download Experiment Data 9
  - Import/Export Experiment Data 10
  - Archive/Dearchive Experiment Data 11
  - Transfer Ownership (from Databases) 12
  - Transfer Ownership (from Local Archives) 13
  - Transfer Ownership (from Shared Archives) 14
- Protocol/Instruction Files 14
- Spiral Systems Data 15
- Statistical Model Input Format 16
- Procedures and Functions 17
- HELP! 17
- SNAP! 18

Chapter 3: Program Maintenance 19
Editing Program Files 19
Compiling Program Files 19
Linking Program Files 20
RMake Facility 20
Error Messages 20
Using dBASE 21

Chapter 4: Program Files 23

Chapter 5: System Default Files 39

Chapter 6: Database Structures 41
Logical Structure 41

Chapter 7: Transfer File Formats 53

Chapter 8: Possible Enhancements 63

Chapter 9: SNAP! Diagrams 69

CHAPTER 1

INTRODUCTION

The GeneTox Manager System is an U.S. EPA system for the collection, organization, retrieval, and analysis of microbial mutagenicity data. Although the system was designed primarily for the *Salmonella typhimurium* plate incorporation assay, it will accept data and information for any mutagenicity assay where similar information is obtained.

Microbial assays, especially *Salmonella* bioassays, are the most common type of mutation tests. These tests determine whether or not a substance interacts with the DNA of the organism in such a way as to create permanent, heritable changes within the DNA. These tests are important because they can be used to screen for both mutagens and genotoxic carcinogens. The *Salmonella* assay has been reported to have been used by over 2000 laboratories. In comparison to other genotoxicity assays, these microbial assays are relatively rapid and inexpensive. These tests, therefore, generate very large databases.

In order to collect, manage, and analyze microbial mutagenicity data in an orderly and rapid manner, the U.S. EPA (through the services of Computer Sciences Corporation) developed the GeneTox Manager system. This system enables investigators to use a PC to collect information and data, to provide short-term storage of the information, and to provide selected types of statistical analysis. The complete GeneTox Manager system also includes a VAX FOCUS system that can be used for the archiving of databases and for the analysis of large groupings of data. The total system is designed to achieve compatibility between the PC and VAX system environments. The GeneTox Manager system is designed to be user friendly through the availability of a structured menu system, screen entry helps, and other help screens. Reports are both tabular and graphical in nature, and they can be seen on the computer screen, sent to a printer, or written to a PC file.

It is the hope of EPA that GeneTox Manager will not only be of benefit to the Agency but will also be of constructive use to others outside of the EPA.

CHAPTER 2

WORKING WITH GENETOX MANAGER

The organization of GeneTox Manager is actually fairly simple, though amazingly intricate in the implementation of this simplicity. Keep in mind that GeneTox Manager is little more than an overgrown data collection and storage program. When a problem occurs, or more development is necessary, work from the precept that this is first a data collection system, then a storage facility, then a data management (Modify) system, a facility to upload information to the VAX and to report information contained in the databases. Side issues are the Protocol view, HELP!, Instructions view, and Suggestion boxes. They all happen at once, and have been coded almost on top of each other at times, but they ARE separate systems: work on them separately.

The system has 18 databases and 21 indexes, designed to minimize duplication of data, and hence the size of the running application. This allows the user to maintain much larger informational reserves with the very limited PC diskspace ceiling. They are designed to look like and function in a similar fashion to hierarchical databases, but keep in mind that they are not. This means that there are NO parent/child relationships between the databases (a particularly important point when attempting data modification or reporting). Eight of these databases are used for data storage, 3 are temporary (used as interim Enter/Modify databases), and the remaining 7 are used for principal investigator information, Suggestions and the HELP! facility. The database structures and indices are explained in detail in Chapter 6.

The indices and databases are dBASE III+ compatible (DON'T USE dBASE IV - it creates problems inside the database that are difficult to correct), making for a useful system manager tool to ensure the integrity of the data.

System Language

GeneTox Manager is written in the Clipper language. Clipper is a compiler that extends the dBASE III+ language. GeneTox code was originally written under the Summer 87 version of Clipper, but has now been modified to use Clipper 5.01. Some features specific to Clipper version 5.01 were used in extending the functionality of GeneTox Manager. The system will **no longer compile** using the Summer 87 version of Clipper. Clipper 5.01 **MUST** be used to produce an executable version of GeneTox Manager from the source code.

All system executables were written in Clipper 5.01 with the exception of the following: Colony.EXE, the Atek colony counter interface, was written in C; BernScr.EXE,

SteadScr.EXE, and HistdSrc.EXE, the statistical modeling code, were written in Fortran. Programming and maintenance for the statistical modeling code is not covered in the document.

Clipper 5.01 and OverLay()

Clipper version 5.01 which includes a Virtual Memory Manager (VMM) which allows large applications to run in a small amount of actual memory. The executable part of the GeneTox Manager system is G_MENU.EXE, which is started by the GTM.BAT batch file.

The system does, however, make use of several separate executables to provide statistical modeling, the communications with the VAX databases, and the archiving and dearchiving of experimental data. These separate executables are run using a third party library called OverLay(). The main use of this library is the Overlay() function itself, which allows running large exes from "within" a Clipper program. It does this by swapping Clipper memory to EMS or disk and thereby freeing memory for use by the called program. After the called program terminates, Clipper's memory is restored and the original Clipper application continues. The syntax for the Overlay() call is Overlay("<command line>", nMemFree, "<temp file path>", "<environment variables>") where the first parameter is the command line to start the exe as it would be entered from the DOS prompt, nMemFree is an integer indicating the number of KBytes to free (0 indicates free as much as possible), the third parameter is the pathname for creating temporary files if memory must be swapped to disk, and the fourth parameter is a string of environment variables used to set the environment for the called program. Other useful OverLay() library functions include O_CD() - change directory, O_MD() -make directory, O_CurPath() - return the current path, and O_DirExist() - does a drive/directory exist.

The linkable Overlay() library code, contained in OVERCL.LIB, must be present in order to link compiled system .OBJs into executable code. This library is specific to linkage using the RTLink linker included with Clipper 5.01.

Separate EXEs

The following separate executables are used by the GeneTox system:

Actual statistical models written in Fortran:

BernScr.EXE
SteadScr.EXE
HistdScr.EXE

Communication program:

SetHost.EXE (DECnet Pathworks connection)

GeneTox.SCR	Execute VAX GeneTox Program
UplNew.SCR	Upload new data to VAX GeneTox databases
UplMod.SCR	Upload modified data to VAX GeneTox databases
DnLoad.SCR	Download data from VAX GeneTox databases

Data Compression Utility:

PKZip.EXE	Compress data for archival
PKUnZip.EXE	Decompress archived data

Additional System EXEs

In addition to the G_Menu and GenGTM files, and the separate executables mentioned above, two files for system distribution exist: GTMLoad, and G_Instal. The GTMLoad executable compresses the GeneTox system onto floppy disks for distribution. Three distribution formats are supported according to the type of floppy drives of the recipient. These are: three 360Kb 5-1/4 diskettes for low density 5-1/4 drives; two 720Kb 3-1/2 diskettes for low density 3-1/2 drives; and one 1.2Mb 5-1/4 or one 1.44Mb 3-1/2 diskette for high density 3-1/2 or 5-1/4 drives. GTMLoad will automatically detect the number of free bytes on the target diskette and determine the format to be used. G_Instal will similarly detect the number of diskettes used when the installation is run.

Also, two separate Upgrade programs (G_Upgrd, GTMRest) are provided to upgrade to version 2.2. For version 2.2 the G_Sample and G_ExSmDt database structures were changed, and the upgrade program (G_Upgrd) is provided to convert users databases to the new format. A restoration program (GTMRest) is provided to restore the system to its previous state should the upgrade be unsuccessful. There is no automated program to generate Upgrade diskettes. These will have to be hand generated according to the disk format specifications documented at the beginning of Upgrade.prg.

G_Instal.EXE and G_Upgrd.EXE are the actual Clipper executables that will preform the installation or upgrade procedure. The are called from DOS batch files named Install.BAT and Upgrade.BAT respectively. The batch files set necessary system environment parameters before executing the Clipper code.

Database Keys

Numeric key values in the range of 1..999,999,999 are used by the GeneTox Manager system for tagging and indexing experiment data. Four separate database keys exist: Experiment keys, which are unique to each strain entered for an experiment; sample keys, which are unique to each sample entered (although samples and sample keys may bridge several experiments); negative keys, unique to each negative control entered; and positive keys, unique to each positive control entered. These values are initialized to some value in the millions range, so that each separate copy of the GeneTox Manager system uses a unique set of database keys. Each installation of GeneTox Manager then has 999,999 keys it may utilize. System default files keep track of the next available keys, and these defaults are incremented as new key values are assigned.

GeneTox Load/Install

The GTMLoad and Install programs were designed for distribution the GeneTox Manager system to potential users. Running the GTMLoad program will produce a set of installation diskettes which may be used to install GeneTox on other machines. When generating a set of installation diskettes, the user is prompted to enter the name of the recipient of the GeneTox Manager system, the initial distribution key (the millions number at which to start database key values), and the GeneTox version (PC or VAX). The initial system password (for access to the protected functions menu) is set to "GENETOX". A database (GTMDist) is updated for each installation set generated with the recipient's name and the distribution key assigned. Distribution keys must be unique. The GTMDist database is also distributed with the system, so that users will not be able to select distribution keys already assigned to other users, even when generating additional installation diskettes.

The Install program installs a complete GeneTox Manager system from a set of installation diskettes generated by GTMLoad. Experiment databases for the installed system will be empty. System databases, such as suggestion text, investigator lists, and system help screens will have the same contents as the GeneTox Manager system that generated the installation diskettes. Once the installation is finished, the initial database keys on the installation diskettes are incremented by 1 million so that the next installation from these diskettes will have different starting database key values.

Currently, the distribution scheme automatically distributes the GTMLoad program when generating the installation diskettes. For some distributions however, you may not wish to allow potential users to generate more installation diskette sets. In this case, remove the line "gtmload.exe" from the GTMMisc.TXT file. (For completeness, you can modify the

GTMLoad source code so that the GTMDist.DBF and G_DstInd.NTX files are not included on the distribution diskette.)

Data Entry

The method for adding data to the databases is to use temporary databases until the user finishes entry and chooses to save the data. This helps simplify removal of data in an abort situation, since until the last step of data entry nothing has actually been saved to the permanent databases. It also helps ensure that the proper database keys are used, since these are set at the time of final submission to the permanent databases.

This is also how the databases are modified: the data for the particular experiment is copied to temporary holding databases and removed from the permanent ones. Once the updates are performed the 'new' data is saved to the permanent databases with the old database keys or if no changes are made the old data is re-saved to the databases.

Temporary databases include: G_TSTRN, which hold the strains entered for the experiment until they are saved; G_TCNTRL and G_HCNTRL which hold positive and negative control information; and G_DOSE which holds dose response data for each sample as it is entered.

All data that has been entered by a user for the current sample is saved when they reach the "Continue" screen and choose to "Continue Entering Dose Response Data", "Change Sample ID & Description", or "Save Data and Exit". Data is saved to the permanent databases at the point when all data for the sample has been entered. Strain data is saved when a sample is analyzed under that particular strain. Strain data will NOT be saved if no samples were entered for a particular strain even though it was entered on the strains screen. GeneTox Manager will warn the user if an attempt is made to exit without analyzing the samples under each strain.

Reports

All reports generated by GeneTox Manager are first generated to temporary disk files, and then directed to the specified output source. This includes Raw Data reports, statistical models (input to and output from), and database experiment listings (using the EXPLSTP.SCR report screen file). The output file for the Raw Data report is RawDat.DAT; the file for statistical model input is called Model.DAT; the statistical model output files are (StatMod<c>.TXT) where the last character is either "B", "S", or "H", depending on the model run (Bernstein, Stead, or Histidine); the experiment listing file is ExpList.DAT. NOTE! the statistical models produce output lines greater that 80 characters and are intended to print in

the printers compressed print mode. GeneTox pauses after generating the model output to allow users to set their printers to condensed print.

Data Status Flags

All GeneTox Manager data records are tagged with a status marker field (typically called STATUS in the database). This status marker may have values of "A", "I", "M", "R", or "V". An "A" status indicates the experiment has been recently added, and has not yet been uploaded to the VAX Master databases. All entered experiment data will be in "A" status on a PC version. On VAX versions of GeneTox, you may not run statistical model or archive data that has not been uploaded. Modifying experiment data that is in "A" status does not change the status flags. An "I" status indicates that the data has been uploaded to the VAX (Inactive). Data in "I" status that is modified will have the status reset to "M" for those records which have been altered. Data that has a status of "R" or "V" is in read-only format and may not be modified, archived, or uploaded. A status of "R" indicates the data was imported from another PC; a status of "V" indicates that data was downloaded from the VAX.

Transfer Files

GeneTox Manager has the capability of creating ASCII text files that contain the entire data set for a particular experiment. This transfer file format (see specifics in Chapter 7) is used for sharing data between PC copies of GeneTox Manager (import/export), uploading and downloading data to the VAX Master databases, storing backup copies of experiment data, and archiving. The GeneTox Manager system contains routines for generating (G_VaxFle) and reading (G_Parse, G_RetDat) data in this transfer file format.

When using the transfer file format for import/export of experiment data, the DOS filename of the created transfer file will be aaaayynn.nnR where aaaayynnnn is the experiment name and the letter "R" has been appended to create a 12 character DOS filename (first 8 characters filename, remaining 3 extension). When creating backup files the appended character will be a "B". When writing data in transfer file format for compression into GeneTox Manager archives, the appended character will be an "X".

Upload/Download Procedures

GeneTox upload and download procedures also make use of the transfer file format. The upload routine will generate a transfer file of experiment data which can be transferred to the VAX and subsequently read into the VAX Master databases (this process is automated for the HERL laboratory site). This file will contain either a complete set of experiment data, if

the data has not been previously uploaded, or a subset of data representing the changes to the data set if the data has already been uploaded. The transfer file generated for uploading is named PCFILE.VAX and it is created in the default directory (pointed to by DftStg). The VAX database is responsible for extracting and generating a transfer file in the appropriate format if downloading from the VAX is enabled (again this is automated for the HERL site).

NOTE! Access to HERL specific upload and download code is simply disabled by setting the G_Version variable to "PC_" in the Version.DFT system default file. To change the current system version, simply delete the file. When GeneTox Manager is started, it will not find the default file, and will prompt the user for the version information. Once selected, GeneTox Manager will create the file and save the user's choice. However, the VAX interface procedures are specific to the HERL site. It is doubtful if the procedures as they currently exist would work for other sites, even if the complete VAX version of GeneTox was provided. Users may have to modify the source code, or at least the communications script files, to match their systems. If no VAX interaction is desired, the specific code could be removed from the GeneTox Manager system prior to distribution.

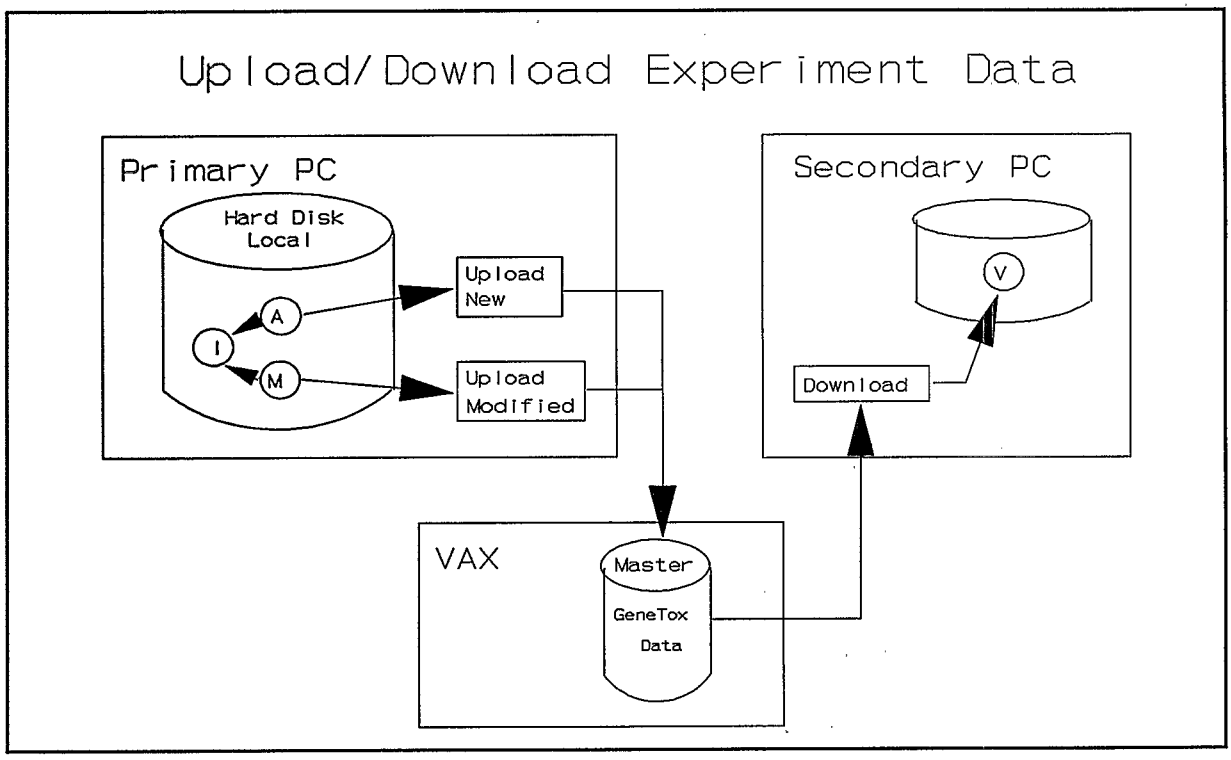
Data Flow Diagrams

Upload/Download Experiment Data

The VAX databases serve as the "master" copy of the data once the experiment has been uploaded. There are two upload choices: upload new and upload modified. Users will use upload new to upload data that has been added to the databases, but which has not been previously uploaded. This data will have a status of "A" (Active). Users will use upload modified data to upload data that has been previously uploaded and subsequently modified. Once data is uploaded, its status flags will be reset to "I" (Inactive).

A second user may choose to download data that has been uploaded to the VAX databases. The user needs only to know the experiment name to perform the download. The data is downloaded as a ReadOnly file: its status is set to "V" (VaxDown). ReadOnly files may not be modified, archived, or re-uploaded; the user may only run reports on ReadOnly data.

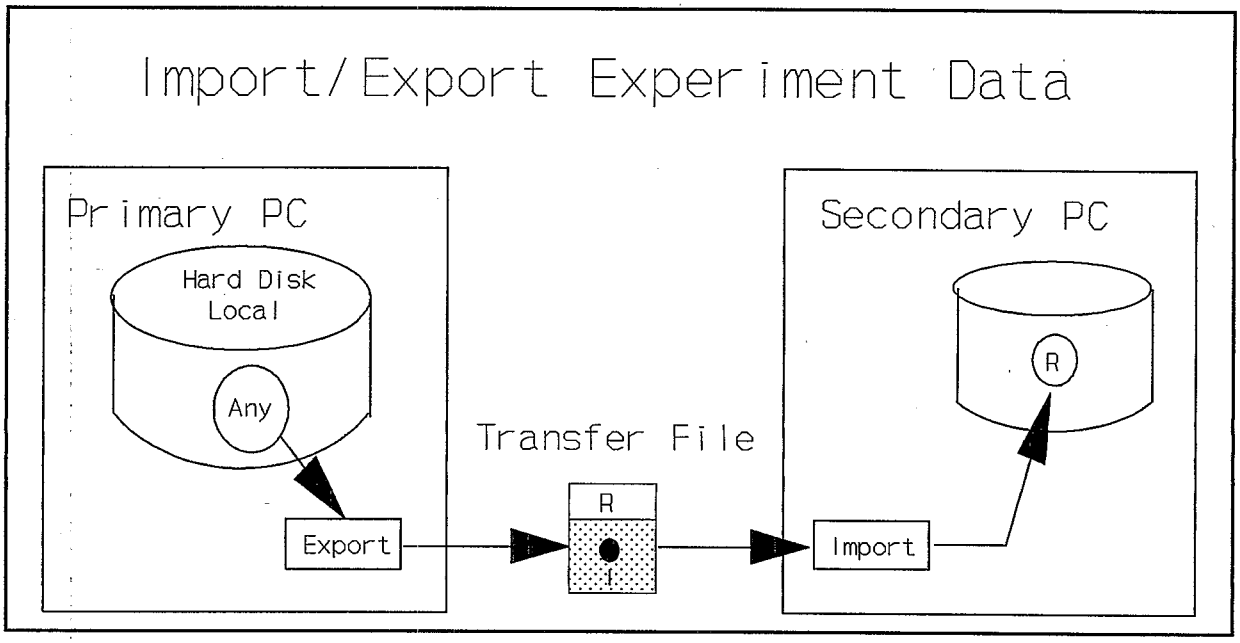
Some special consideration is given to downloading when the incoming experiment name matches a current experiment. If the status of the existing experiment is "A", the user is given the option of renaming the existing experiment (the user could not upload it to the VAX databases anyway, because all experiment names must be unique). If its status is "M" or "I", the user is given the option of overwriting the existing data with the download data.



If its status is "R" or "V" (ReadOnly), the user is given the option of overwriting the existing experiment if the download experiment is actually the same experiment (database keys match) or renaming the download experiment (keys don't match, names just same).

Import/Export Experiment Data

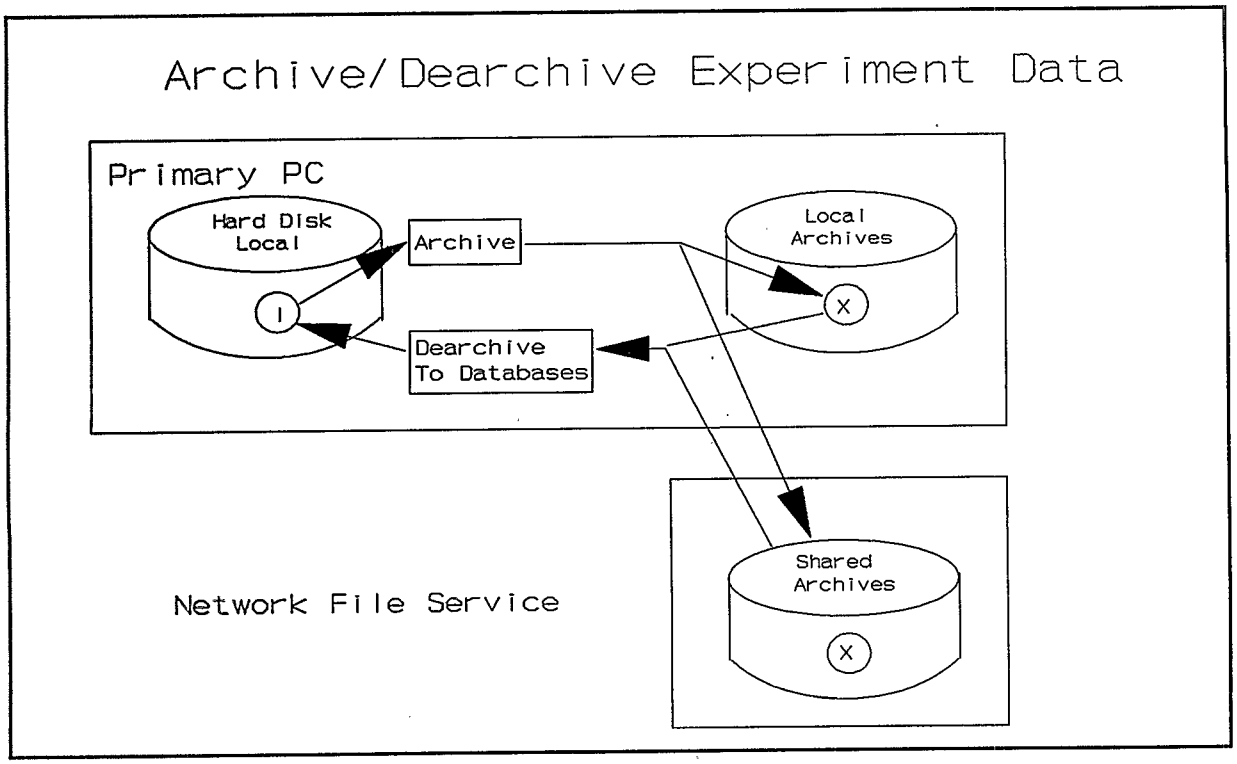
A user may export experiment data in ReadOnly form for use by another user by selecting the Export option. The second user retrieves the data into his databases using the Import option. The imported data is in ReadOnly form ("R" status flag): it may not be modified, archived, or re-uploaded. The receiving user may only run reports on the data. To transfer ownership of the file to another user, users must use the archive/dearchive options (see below).



Special considerations must also be given when importing data if an experiment with the same experiment name already exists. If the status of the existing experiment is "A", "I", or "M", the user may not import the experiment if it was originally created on his PC (database keys match). If it did not come from his PC, he may rename the incoming import experiment name. If the status of the existing experiment is "R", or "V" (ReadOnly), the user is given the choice of overwriting the experiment if it is actually the same experiment (database keys match), or of renaming the incoming import experiment (keys don't match, names just same).

Archive/Dearchive Experiment Data

Users may elect to archive experiment data to reduce the size of their working databases and increase the speed of database accesses. Archiving and dearchiving may also be use to transfer ownership of an experiment to a new user (see below). When archiving data, the data is first written to a flat file in transfer file format (see Chapter 7). This file has a filename of the experiment name + "X". This file is then compressed using the PKZip data compression utility and placed in the designated archive file. The archive filename defaults to "GTMArchv.ZIP" and the location of the archives defaults to the values using "Locate Archives" from the Defaults menu. Both location and filename may be specified by the user when archiving data. The user also selects a password, which must be used to dearchive the data. Once archived, the data is deleted from the users databases.

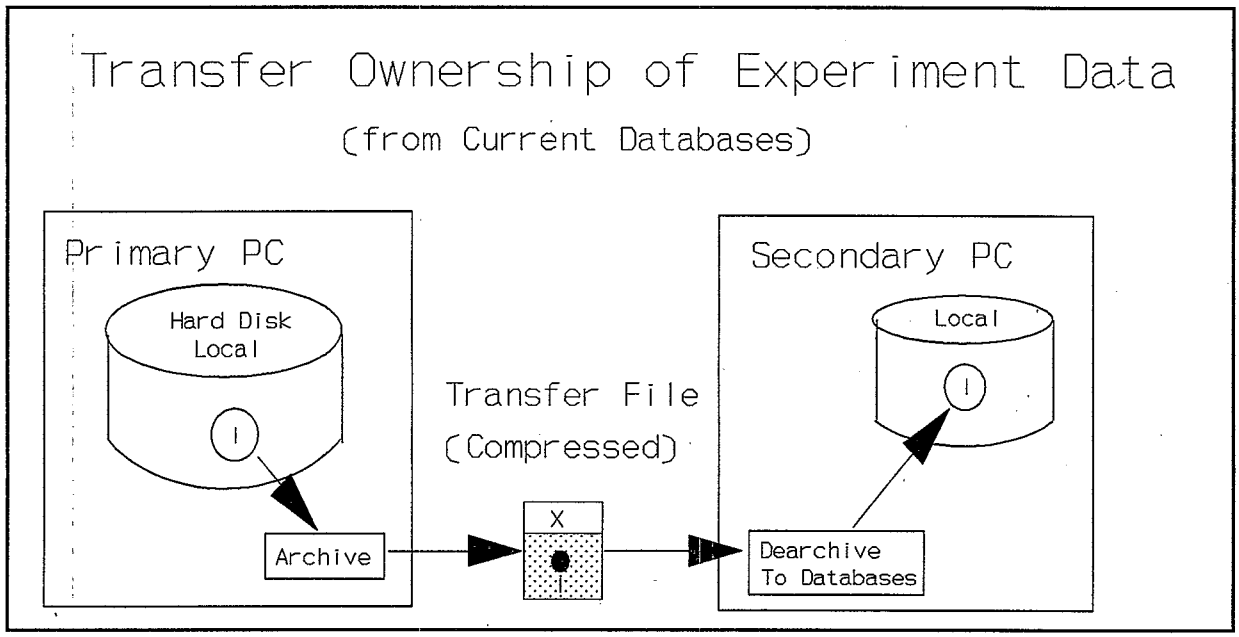


Data can be archived to either local archives residing on the PC's hard disk, or to shared archives residing on a network file service. When data is archived, GeneTox updates a database of archived experiments residing on the user's PC. The database contains the experiment name, location, and password for each experiment that has been archived from the PC. This way, when dearchiving data (use "Dearchive To Databases"), a user does not need to enter the location information if the location information stored in the archived experiments database is correct, only the password.

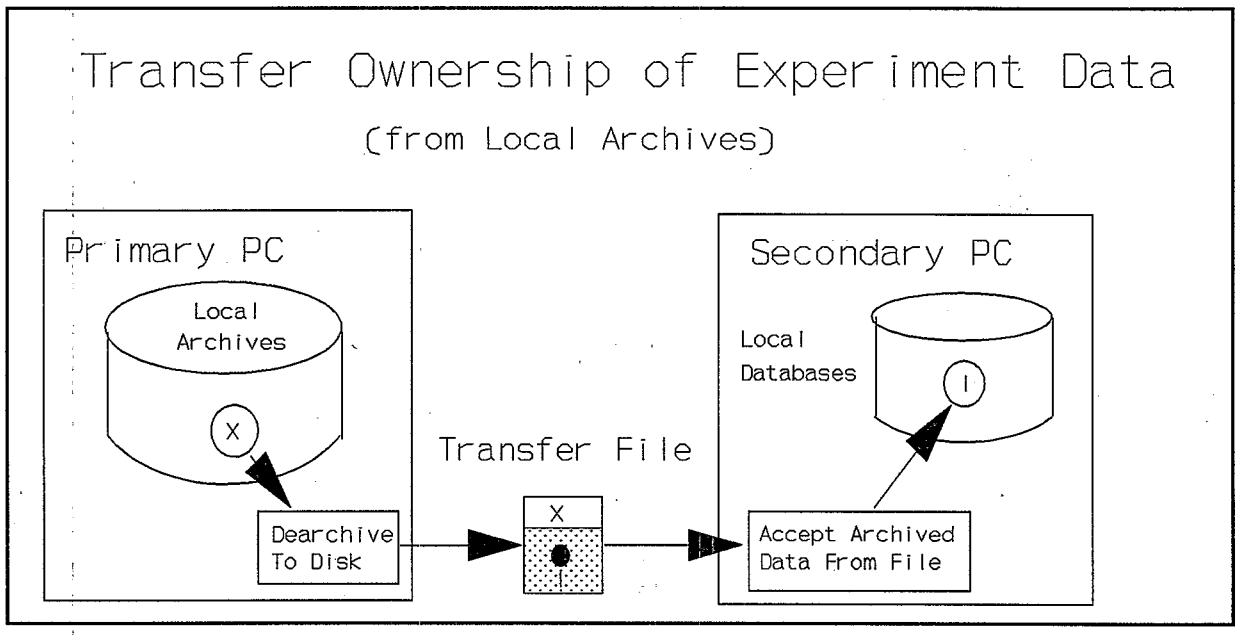
Transfer Ownership (from Databases)

A second usage of the archive and dearchive functions can be to transfer ownership of an experiment from one user to another. In this case, it depends on where the archived experiment is stored, as to how the transfer can be accomplished. If the data is stored in the original users databases, then a transfer may be accomplished by archiving the data to a floppy disk rather than the default archive file at the default location. The receiving user may then retrieve the data by dearchiving the data from the floppy.

To dearchive data that the user did not originally archive, use "Dearchive To

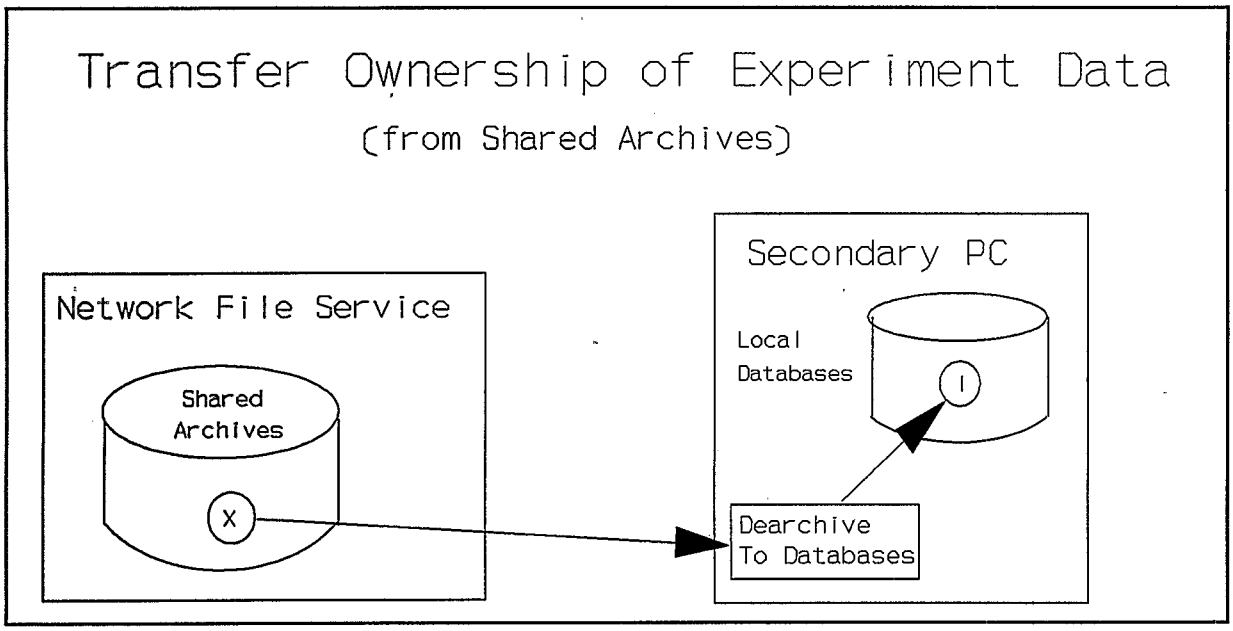


Databases". The receiving user must know the location and filename of the archive file, and the password the experiment was archived under. This is because the receiving users archives database will have no record of the experiment, and will not automatically have the archive file location and name information available. In this case, GeneTox will prompt the user for the archive location and filename information.



Transfer Ownership (from Local Archives)

If a user wishes to transfer ownership of an experiment that has been archived to a local archive residing on their PC, the following process may be used. Select "Dearchive to Disk" from the Utilities/PC Functions menu to create a flat file in transfer file format that is uncompressed. This file can then be brought into a second user's databases using the "Accept Archived Data From File" option. In this case, the receiving user need only know the name of the experiment.



Transfer Ownership (from Shared Archives)

If a user wishes to transfer ownership of an experiment that has been archived to a shared database (that both users can access), the following process may be used. The original owner of the experiment needs to provide the receiving user with the location and archive filename and the password under which the experiment was archived. The receiving user may then use "Dearchive To Disk" to dearchive the data from the shared archives. Because the receiving user's archives database has no record of the experiment, the user must enter the location and filename of the archive file, if the default archives were not used.

Protocol/Instruction Files

The GeneTox Manager system includes the ability to view protocol files and experiment instruction files from within a GeneTox Manager session. Protocol files are text files detailing general methods, procedures, and equipment needed to conduct certain types of experiments. The protocol information is contained in two levels of text files. The first level file can be specified by selecting "Protocol File Name" from the Defaults menu (please specify complete path with filename). The value entered is retained across GeneTox Manager sessions (in the Proto system default file). The specified file will contain several lines of text in the form of a table of contents. Appended to the end of each line (starting with column 81, off screen) is the filename of the text file to display. The second level files are assumed to be in the same location (drive/directory) as the top level file. Pressing return on a particular line causes the text contained in the file indicated at the end of that line to be displayed. These files are static and cannot be modified from within the GeneTox Manager system. There are currently no aids to assist users in generating their own protocol files, or modifying existing protocol files.

Instruction files contain specific instruction for conducting an experiment or set of experiments and are created and modified from within GeneTox Manager. Instructions specify the experiment test type, assay modifications, strains, activation conditions, and sample information for the experiment or set of experiments. The user may select a specific instruction set from the Instructions-Select menu. That instruction set is in effect until another instruction set is selected, even after exiting the program (stored in the G_Instr system default file). The user may choose to append the current instruction set text to the Experimental Remarks section. The protocol text may be viewed at any point from within GeneTox Manager by pressing Shift-F5. The instruction set, if selected, may be viewed only when entering or editing experiment data by pressing Shift-F3.

Spiral Systems Data

A relatively late addition to the GeneTox Manager system has been the ability to read data collected using the Spiral Systems spiral colony counter and control program. The user must provide two data files that the spiral program generates: an .ID file and a .PRN file. The spiral program stores data for each sample analyzed separately under a different filename. The file containing general sample and strain information will have an .ID extension. The dose response data is not stored by the spiral systems program in an ACSII readable format. However, the spiral program provides an export function for transferring dose response data to Lotus-123 spreadsheets, which generates a .PRN file containing the control and dose response data for each strain. Both files must be present for GeneTox to be able to import the data.

Imported data is assigned new experiment keys (as if it had been hand entered) and given a status of "A". Different tests of the same Spiral sample must be saved under different GeneTox experiment names.

Statistical Model Input Format

The statistical models read a text file generated by GeneTox Manager containing dose response data for each sample in the following format:

6 Lines of Text, 72 Characters or Less in Length, followed by any number of dose information lines in the format:

<code> <name> <amount> *<units> <plt1> <plt2> <plt3> <plt4> <plt5>

where an * (in column 36) indicates a dose line is to be ignored
(positive and additional negative controls would be marked with an *)
and an @EOF (in column 36) marks the end of this sample

<code> is a 2 digit numeric field indicating the pos/neg control.
it is left blank for dose responses

<name> is a 20 character text field describing the pos/neg control.
it is left blank for dose responses

<amount> is a numeric field a maximum of 9 characters in length

<units> is a 3 character text field indicating dose amount units

<plt#> is a numeric field a maximun of 5 characters in length

Consider the following example:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456
7890

Sample Id: PARC-91-0017      ResLab..: EHRT        Exp Date.: 11/01/91
Samp Desc: 1-Nitrocarbazole
Exp. Name: PARC910044        Lab Tech: MST         P Invest.: JL
Strain...: TA98              Str Date: 10/29/91    Act Type.: RLA059
Test Type: 01 - Standard Plate Incorporation Assay   Act Level: 0
Time (Hr): 0.0               Temp C..: 37.0        Input by.: MST
01  Sodium Azide             3.000  *ug    344   347   345   (ignore this line, a positive
control)
                        0.000  ul    19    19    24
                        5.000  ug    77    26
                       10.000  ug    29    45
                       25.000  ug    28    30
                            @EOF
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456
7890
```

Procedures and Functions

Procedure and Function files are used heavily in this system and many entire tasks are performed inside of these procedure/function files.

G_Functs	contains system library code.
G_EnFunc	contains enter/modify specific functions.
G_EnProc	contains enter/modify specific procedures.
IVFuncts	contains system procedures.
IVLib	contains the procedures for HELP!.

HELP!

HELP! is a Clipper utility that automatically sets F1 to HELP if one chooses to engage it. HELP! as Clipper designed it can be very specific (to the line of code) but becomes very

large when used this way. Originally, GeneTox Manager's HELP! was like this but it was too large to run. It has been re-written as a general HELP! facility. The original help facility has been saved as .bak which can be restored at a later date.

SNAP!

At the end of this document are the tree diagrams built by SNAP!, a piece of shareware that "documents" dBASE and/or Clipper applications. These show the calling order of program procedures and functions and database files. Each executable file has been documented separately, since they are indeed segments of one larger system, GeneTox Manager.

To use SNAP! type **SNAP** at the DOS prompt. The software will ask for the following information:

System name:	GeneTox Manager
Author:	Whoever wrote the programs being documented
Copyright holder:	Person who owns the software
Copyright date:	1992
Top file:	Starting program
Path for source:	Where the .PRG files are located
Path for data:	Where the .DBF files are located
Path for output:	Where the completed files are to go
Path for SNAP! files:	Where the SNAP! source code is located

Documentation for the SNAP! program can be found on the PC that is being used for GeneTox Manager.

CHAPTER 3

PROGRAM MAINTENANCE

The following sections will explain how to edit, compile and link the files to create the executable file. It also contains some error messages and how to fix the problems.

Editing Program Files

To make a change to the GeneTox Manager software, an existing program file must be edited or a new one created. To make changes to an existing file, first determine which file has to be edited. If an error message has been received the file will be named in the message. If an update is requested, edit the file that performs the function to be modified. If something new has been requested, a new program file will have to be created. Make sure to name the new file with a unique name or unexpected results could occur.

Use an editing package that will save the program file in an ASCII format. This is required because special wordprocessing format characters embedded in the files will cause error messages during compiling and create other problems.

Certain PRG files contain extended graphics characters used to draw the GeneTox Manager screens. Some text editors convert these graphics characters to regular text, and will save the screens in this manner. For this reason, use a text editor that does not destroy the graphics characters when editing these files.

Compiling the Program Files

Once the necessary changes have been made to a program file, it must be compiled. This is done using Clipper, so the path must include a call to the Clipper subdirectory or the CLIPPER.EXE file must be in the same directory as the .PRG files. To compile a program file with line numbers, type:

CLIPPER filename (without .PRG)

Do this when testing the program as the error messages will include line numbers to indicate where an error is occurring. However, this option makes the file larger. When testing is complete, recompile the program file with the following qualifier (a lowercase L):

CLIPPER filename -l

This will compile the program file without line numbers and will reduce the size of the object file.

Linking the Program Files

To link the files, use the Clipper "RTLINK" function. Simply list the names of the object files be linked. These will in turn create the executable file that will be used to run the system. The format is:

RTLINK FILE objectfile,objectfile

If the files are not in the current directory, make sure to include the correct paths in the filenames.

RMake Facility

Clipper 5.01 includes a make facility for automatic executable generation when one of the source files is modified. To run the Rmake facility set up for GeneTox, type RMAKE GTM.
A makefile called GTM.RMK has been created which outlines the dependencies between the source code modules and the procedures for generating a new executable. Only the necessary files will be recompiled and linked. The makefile will make the G_Menu executable, and the GTMLoad and Install system distribution utilities. Note however, that the existing makefile is only valid if existing PRG files are modified. If new files are added to provide additional functionality, the programmer will also have to update the makefile to describe these new PRG files and their relationship to the system.

Right now the multitude of PRG procedure and function files have been divided into categories to group those with similar function together. Data entry/modification files are compiled into DataEntr.OBJ; Utility functions (PC & VAX) are compiled into Utils.OBJ; Reporting functions are compiled into Reports.OBJ; System default functions are compiled into Defaults.OBJ; Data save and storage functions are compiled into DataSave.OBJ; Help functions are compiled into Help.OBJ. G_Functs is compiled into its own OBJ. Reports, Defaults, DataSave, Help, DataEntr, Utils, and G_Functs are linked into G_Menu.EXE.

Error Messages

Most of the error messages will appear on the screen in the following format:

PROC procname LINE lineno, errormsg errobject

where: **procname** is the procedure where the error occurred, **lineno** is the program line number that contains the error (this will be 0 if the file was compiled with the -l

option), **errormsg** tells what is wrong, and **errobject** is the thing (variable name, file, etc.) that caused the problem.

The following are examples of errors that can occur. The important part is the errormsg segment to figure out what is wrong.

PROC MCHOICE LINE 0, open error G_ENIND.NDX

> Could not open the file G_ENIND.NDX - it could be missing, corrupted, or the file limit may have been exceeded. Only 15 files may be open at any given time and with GeneTox Manager, the maximum number of files is open during the Enter, Modify and Reports options. Reindexing the databases usually will take care of this error by recreating the index files or making sure only 15 database or index files are open at one time.

PROC SETGDEF LINE 1213, unidentified identifier G_VAL1

> The variable G_VAL1 was not recognized. Make sure it has been initialized (if needed) or that the right database file has been called.

An OUT OF MEMORY error will occur if the machine has run out of base memory because too many other things are running. Make sure no other processes are running before starting GeneTox Manager to avoid this problem.

Using dBASE

To look at any of the database files, dBASE will have to be used. To invoke dBASE, type **dBASE** at the DOS prompt in the subdirectory that contains the dBASE executable files. Once in dBASE, there are two ways to look at the data - the Assistant or dot prompt commands.

The **Assistant** is an easy-to-use system that guides the user through the process of looking at a database file, by using pop-up menus and prompts.

Dot prompt commands require a little more knowledge of dBASE. To look at the data in a database file, type **USE filename**, where filename is the name of the database file. Then use the dBASE commands to look at the data. A few of the commands are explained below. To use these commands, just type the word at the dBASE dot prompt. Then use the <PgUp> and <PgDn> keys to move between records and the arrow keys and <Return> key move between fields. Use the dBASE manual to get a

more detailed explanation of these commands and other commands that can be used.

EDIT - allows changing the data in the record.

LIST - lists the records on the screen. Some of the forms of this command are : **LIST**, which will list all of the fields for all of the records; **LIST FOR fieldname = "qualifier"**, which will list all of the fields for all of the records that meet the condition specified by **qualifier**; and **LIST fieldname**, which will list **fieldname** for all of the records.

GOTO TOP, GOTO BOTT and **GOTO recno** - goes to the top of the database, to the bottom of the database, or to the record number specified by **recno**.

QUIT - leaves dBASE and returns to the DOS prompt.

Again, see the dBASE manual for more commands.

CHAPTER 4

PROGRAM FILES

The following is an alphabetical listing of the program files used for the GeneTox Manager system. The list includes the name of the program and a very brief description of what the program actually does.

Bbridge.Prg
This routine calls and runs the Bernstein model.

ChkKeys.Prg
Stand alone system utility to report contents of database key default files.

ChkPass.Prg
Stand alone system utility to decrypt and report the current system password.

DSRCH.Prg
This routine searched databases based on selected criteria and sets up the data extraction to be performed.

G_Accept.Prg
This routine allows the user to receive experiment data that has been dearchived by another user to a disk file for the purpose of transferring ownership of the experiment.

G_AcInit.Prg
This routine initializes all variables needed for the Activation System Information Screen.

G_AcScnr.Prg
This routine draws the initial screen for the Activation System Information Screen.

G_ActSys.Prg
This routine allows the user to enter Activation System Information into temporary variables.

G_AdSInt.Prg
This routine initializes the necessary variables before calling G_AdSmpl to add new Samples.

G_AdSmpl.Prg

This routine allows the user to add new Samples to an existing Experiment.

G_AllRpt.Prg

This routine submits selected data to and runs each statistical model.

G_AMAssc.Prg

This routine draws the Assay Type screen.

G_AMAssy.Prg

This routine allows the user to add or modify the Assay Type for the experiment.

G_Archiv.Prg

This routine allows the user to archive an experiment in compressed format to save disk space and decrease database size.

G_ArcMnu.Prg

This routine prompts the user to Archive/Dearchive/Accept experiment data.

G_Assay.Prg

This routine allows the user to select the Assay Type for the experiment.

G_AsScnr.Prg

This routine draws the initial screen for the Assay Type Selection Menu.

G_ChkCnt.Prg

This routine provides header information to the VAX UPLOAD option.

G_ChkTpn.Prg

This routine provides header information to the VAX UPLOAD option.

G_ClrSep.Prg

This routine clears the database G_Sep.DBF. The database is used to make sure that the extended Sample and S9 descriptions are only printed once per RAW DATA output. This is used to clear that database so that on the next run of a Raw Data report the EXPKEY will not be found. This is only a quick fix.

G_Colors.Prg

This routine allows the user to change the color settings used in the system.

G_Commun.Prg

This routine allows the user to select the type of communications procedure to be used.

G_Contnu.Prg

This routine allows the user to decide what action to take.

G_CurStr.Prg

This routine allows the user to enter the Current Strain, Activation Level and Dose Units for the current sample.

G_CuScrn.Prg

This routine draws the screen for the Current Strain and Activation Level Data Entry.

G_DBRInd.Prg

This routine recreates all indexes associated with GeneTox Manager databases.

G_DBZap.Prg

This routine permanently deletes all data from GeneTox Manager databases.

G_Dearch.Prg

This routine allows the user to dearchive experiment data which has been stored in compressed format and return it to their databases.

G_DelExp.Prg

This routine permanently deletes an experiment from the databases.

G_DelSmp.Prg

This routine allows the user to delete a Sample from an Experiment.

G_DelRV.Prg

This routine deletes an experiment that has been imported from another user or downloaded from the VAX.

G_DnLoad.Prg

This routine allows the user to download experiments stored on the VAX.

G_DoPrnt.Prg

This prints the dose responses if they have been entered into memory variables previously.

G_DoScrn.Prg

This routine draws the initial screen for the Dose Data Entry.

G_Dose.Prg

This routine will allow the user to enter an unlimited amount of dose data, but presently only 10 doses are allowed.

G_DoseNu.Prg

This routine draws the correct number of doses to be entered based on NUM_DOSE.

G_EnFunc.Prg

This routine contains all of the functions used by the enter/modify routines.

G_EnProc.Prg

This routine contains all of the procedures used by the enter/modify routines.

G_Enter.Prg

This routine handles all of the data entry for the GeneTox Manager system.

G_ExInit.Prg

This routine initializes all variables needed for the experiment description screen.

G_ExpLod.Prg

This routine allows the user to retrieve backup experiment data should the databases become corrupted.

G_Expmnt.Prg

This procedure allows the user to enter experiment description data.

G_Export.Prg

This routine allows the user to export experiment data for use by other users. Exported data is in a "Read Only" status.

G_ExScrn.Prg

This routine draws the screen for experiment data entry.

G_Extrnl.Prg

This routine declares all the data entry routines for execution.

G_Functs

Contains a library of system functions.

G_GenNps.Prg

This routine generates Negative and Positive control data sets for the VAX UPLOAD option.

G_GenRec.Prg

This routine is used by the VAX UPLOAD option.

G_GenTst.Prg

This routine generates Test plate count data for the VAX UPLOAD option.

G_GeScrn.Prg

This is routine draws the main menu screen for the GeneTox Manager system.

G_GetDP.Prg

This routine allows the user to enter a drive and path and verifies their existence.

G_ImpExp.Prg

This routine allows the user to choose Import/Export of experiment data.

G_Import.Prg

This routine allows the user to import experiment data that was exported by another user. Imported data is in "Read Only" status.

G_InAdd.Prg

This routine allows the user to add instructs to the instruction database.

G_InAScr.Prg

This routine draws the screen for the Instruct add module.

G_Instal.Prg

This routine is part of the Install executable. It loads the GeneTox system onto hard disk from 2 high density floppies that were created using the GTMLoad exe.

G_InstFV.Prg

This routine allows the user to view instructs that have been saved to disk.

G_Instrc.Prg

This routine allows the user to change the instruction help set.

G_IntClr.Prg

This routine defines the default colors to be used with the system.

G_InView.Prg

This routine allows the user to view instructs that have been saved to disk or to modify the instruction set.

G_ListEx.Prg

This routine allows the user to view or print the names of the experiments in the database.

G_LodDat.Prg

Contains the routines used for retrieval of data by G_ExpLod, G_Import, and G_Dearch.

G_LstArc.Prg

This routine allows the user to browse the database of archived experiments.

G_MDeflt.Prg

This routine draws the menu for the defaults menu and prompts the user for a selection.

G_MEnter.Prg

This routine draws the Data Entry menu and loads the Screen Array with the necessary values.

G_Menu.Prg

This routine is the main menu system for the GeneTox Manager system.

G_MoCInt.Prg

This routine initializes the necessary variables before calling G_MoCntl.

G_MoCntl.Prg

This routine allows the user to modify neg/pos control information.

G_Modify.Prg

This routine allows the user to modify existing data in the system.

G_MoDInt.Prg

This routine initializes all of the necessary data for the Dose Data Entry screen.

G_MoDose.Prg

This routine allows the user to modify existing Dose Responses for a given strain, date, and Act.

G_ModSmp.Prg

This routine allows the user to modify Sample data.

G_MoGen.Prg

This routine allows the user to modify general experiment information.

G_MoGInt.Prg
This routine initializes all necessary data before calling G_MoGen.

G_MoSInt.Prg
This routine initializes all necessary data before calling G_MoSmpl.

G_MoSmpl.Prg
This routine allows the user to modify Experiment Specific Sample Data.

G_MoveDo.Prg
This routine controls the cursor movement for G_Dose.Prg.

G_MQuit.Prg
This routine draws the Quit Menu and accepts user response to Quit/Continue.

G_MSvAct.Prg
This routine saves the modified data for the Activation System Information Screen.

G_MSvAsy.Prg
This routine saves the modified data for the Assay Type Selection Screen.

G_MSvCtl.Prg
This routine saves the modified data for the Neg/Pos Control Screen.

G_MSvDos.Prg
This routine saves the modified data for the Dose Response Data Entry Screen.

G_MSvExp.Prg
This routine saves the modified data for the Experiment Description Screen.

G_MSvQlt.Prg
This routine saves the modified data for the Quality Assurance Checks Screen.

G_MSvRmk.Prg
This routine saves the modified data for the Experimental Remarks Screen.

G_MSvSmp.Prg
This routine saves the modified data for the Sample Specific Screen.

G_MSvStn.Prg
This routine saves the modified data for the Final Strain Entry Screen.

G_NegPos.Prg
This routine allows the user to enter Positive and Negative Controls for a given strain.

G_NeScrn.Prg
This routine draws the initial screen for the Neg/Pos Control Entry Screen.

G_NewPas.Prg
This routine allows the user to enter a new password for the Special Options menu.

G_Parse.Prg
This routine parses an input file to make sure all experiment data is in the right format before saving it to the databases using G_RetDat.

G_PC.Prg
This is the main calling program for the PC Utilities function.

G_PiAdd.Prg
This routine allows the user to enter new Principle Investigators into the List.

G_PiMenu.Prg
This routine allows the user to enter, change or delete entries from the Principle Investigator List.

G_PiMod.Prg
This routine allows the user to change entries in the Principle Investigator List.

G_PltItr.Prg
This routine allows the user to modify the number of interactive readings from a colony counter for each plate count.

G_Proto.Prg
This routine allows the user to view the Protocol file anytime.

G_Qualty.Prg
This routine allows the user to enter Quality Assurance Checks into the database.

G_QuitAl.Prg
This routine terminates the GeneTox Manager system and returns to DOS.

G_QuScnr.Prg
This routine draws the initial screen for the Quality Assurance Checks Screen.

G_RawDat.Prg

This routine allows the user to generate a report that includes all data for an experiment.

G_Remark.Prg

This routine allows the user to enter Experimental Remarks into the database.

G_Replac.Prg

This routine replaces the status variable "A(ll)" with "I(gnore)" following upload to VAX.

G_Report.Prg

This routine allows for a parameter setup for a data search over all pertinent data ranges.

G_ReScnr.Prg

This routine draws the initial screen for the Experimental Remarks Screen.

G_RetDat.Prg

This routine saves data in an input file into the databases. It assumes that G_Parse has been called previously to validate the data.

G_RpPage

This routine prints the page number information on each report page.

G_RpRaSH.Prg

This procedure prints the Headers for the Sample description and the S9 description.

G_RpRaw9.Prg

This procedure prints S9 Description and information on separate page.

G_RpRawD.Prg

This procedure prints the header for the Dose amounts.

G_RpRawF.Prg

This procedure prints the Negative control data to the screen or printer if it is the first time this Strain has been printed.

G_RpRawG.Prg

This procedure prints the general information for the Raw Data Report.

G_RpRawH.Prg

This procedure prints the header for the Raw data Report.

G_RpRawO.Prg

This procedure prints the Other Control data if it exists.

G_RpRawR.Prg

This procedure prints the Sample Remarks.

G_RpRawS.Prg

This procedure prints the Raw Data separate pages to the screen or the printer. The separate pages are for the Sample description and the S9 description. The descriptions are printed once per sample id.

G_RpRawT.Prg

This procedure prints the Dose response data and the plate counts to the screen or the printer for the current strain.

G_RptChs.Prg

This routine allows the user to select which Report to run.

G_RptOut.Prg

This routine asks the user for the destination of the statistical reports generated.

G_SaveAl.Prg

This routine calls the appropriate modules to save all temporary variables to the databases.

G_SavExp.Prg

This is routine saves the temporary variables to the G_Expmnt database.

G_SavGen.Prg

This routine save all general data to the G_GenDat database.

G_SavNeg.Prg

This routine saves all negative control information to the G_NegDat database.

G_SavNew.Prg

This routine saves all data for a newly entered experiment.

G_SavNP.Prg

This routine calls either G_SavNeg or G_SavPos depending on the control code stored in the temporary controls database. Each positive and negative control will be saved. NOTE: The first negative control to be used is the control for the statistics model.

G_SavPos.Prg

This routine saves all positive control information to the G_PosDat database.

G_SavSmd.Prg

This routine saves the experiment specific information into the G_Exsmdt database.

G_SavSmp.Prg

This is routine saves all memory variables to the G_Sample database.

G_SavTst.Prg

This routine saves all test data into the G_Test database.

G_SCntrl.Prg

This routine calls the screens as defined by the screen array.

G_SelSug.Prg

This routine sets up the work areas for the suggestion databases.

G_SetFlg.Prg

This routine resets the status flag for a particular experiment to a user specified value.

G_SmpSpc.Prg

This routine allows the user to enter Sample specific data into the system.

G_SmScrn.Prg

This routine draws the initial screen for the Sample Specific Data Entry Screen.

G_SpeOpt.Prg

This routine allows the user to enter special options into the system and is password protected. The backdoor is "LTC".

G_Spiral.Prg

This routine converts data collected from a spiral counter using Spiral Systems data collection program. Data files <SampID>.ID and <SampID>.PRN must be provided to the conversion routine.

G_SpMenu.Prg
This routine allows the user to enter options provided by the Special Options.

G_Strain.Prg
This routine allows the user to enter Strain information into memory variables.

G_StScrn.Prg
This routine draws the initial screen for the Final Strain Entry Screen.

G_SuAct.Prg
This routine allows the user to modify the suggestion box for the Activation System Information Screen.

G_SuAssy.Prg
This routine allows the user to modify text for the Assay Selection Screen.

G_SuCur.Prg
This routine allows the user to modify the suggestion box for the Current Strain Selection Screen.

G_SuDose.Prg
This routine allows the user to modify the suggestion box for the Dose Response Screen.

G_SuExp.Prg
This routine allows the user to modify the suggestion box for the Experiment Description Screen.

G_SuFinl.Prg
This routine allows the user to modify data for the Suggestion List.

G_Sugges.Prg
This routine allows the user to enter help text or a listing for the suggestion box.

G_SugOff.Prg
This routine turns off the suggestion function keys.

G_SugOn.Prg
This routine sets the suggestion function keys to the routines for the suggestion keys.

G_SuHelp.Prg
This routine allows the user to enter help text for the suggestion box.

G_SuHMod.Prg

This routine allows the user to modify existing help text in the suggestion box.

G_SuList.Prg

This routine fills SUGG1.DBF with missing information when there is no listing for a suggestion box.

G_SuLMod.Prg

This routine allows modification of the Suggestion Lists.

G_SuMod.Prg

This routine allows the user to modify the suggestion box for various screens.

G_SuNp.Prg

This routine allows the user to modify the suggestion box for the Negative/Positive Control Screen.

G_SuQual.Prg

This routine allows the user to modify the suggestion box for the Quality Assurance Screen.

G_SuRep.Prg

This routine allows the user to modify the suggestion box for the Experiment Description Screen.

G_SuRExt.Prg

This routine allows the user to modify the suggestion box for the Reports used in the GeneTox Manager System.

G_SuRmrk.Prg

This routine allows the user to modify the suggestion box for the Experimental Remarks Screen.

G_SuSamp.Prg

This routine allows the user to modify the suggestion box for the Sample Specific Screen.

G_SuSel.Prg

This routine allows the user to select which screen to modify the Suggestion text or list for.

G_TermHe.Prg

This routine terminates the modify help mode.

G_Util.Prg
This routine allows the user to choose PC/VAX Functions.

G_Vax.Prg
This routine is the main calling program for the VAX Utilities function.

G_VaxFle.Prg
This routine is the main calling program for procedures used to create the VAX UPLOAD file.

G_Vpubl.Prg
This procedure initializes all of the public variables for the GeneTox Manager system.

G_Welcom.Prg
This routine displays a welcome message to the user for 20 seconds or until a key is pressed.

G_Xtrct.Prg
This is routine prompts the user for the experiment name for which to extract data.

GTMLoad.Prg
Generates a set of installation diskettes from functioning GeneTox Manager system.

GTMRest.Prg
Restores GeneTox Manager systems converted to version 2.2 database formats using the GeneTox Manager Upgrade program to their original condition.

Hbridge.Prg
This routine calls and runs the Histidine model.

Help.Prg
This routine displays the help text on the screen.

HelpText.Prg
Stand Alone system utility to print contents of IVHelp memo fields.

InsLib.Prg
Library routines used by GeneTox Manager install program.

Install.Prg

Installation program for GeneTox Manager.

IVFuncts.Prg

Library routines used by GeneTox Manager system.

IVLib.Prg

Library routines used by GeneTox Manager system.

Qbridge.Prg

This routine calls and runs the Quasi-Stead model.

Sbridge.Prg

This routine calls and runs the Stead model.

UpgLib.Prg

Library routines used by GeneTox Manager upgrade program.

Upgrade.Prg

Database upgrade program of upgrading to GeneTox Manager v2.2 database format.

CHAPTER 5

SYSTEM DEFAULT FILES

The following default files are used by the GeneTox Manager system. They are standard Clipper MemVar files that have been given a .DFT extension. Most are initialized by G_VPubl for the duration of the application, although a few are initialized and saved by individual procedures. They are generally not saved unless the user changes a system default setting.

MemVar files have .MEM extension and are located in the application directory (directory where system executables reside).

MemVar File	Variable	Remarks
DrvPath	DftStg	GeneTox database location (<drv>:\<path>).
GTMArchv	ArchStg	Archives location (<drv>:\<path>).

Default files have .DFT extension and reside in database directory (specified in the DrvPath.MEM MemVar file).

Default File	Variables	Remarks
ColorHel	ColorHelp	Color string used when displaying system help.
ColorsH	Header	Color string used when displaying screen headings.
ColorsN	Normal	Color string used for main menu, first level menus, and data e n t r y screens.
Edit_Col	Edit_Color	Color string used for second level and lower menus.
Err_Colo	Err_Color	Color string used when displaying system error messages using the Error_Msg() function.
Exp_Colo	Exp_Color	Color string used when displaying expanded suggestion boxes containing no selectable items.
Exp_Edit	Exp_Edit	Color string used when displaying expanded suggestion boxes containing selectable items.
Func_Col	Func_Color	Color string used to display currently active function keys.
G_Array	Inc_Array	Allow addition of new assays? (T/F)
G_Assays	Assay1..26, AssayEnter, AssaySave	Contains user entered assay information.
G_Commun	CommunType	String containing communications link type to VAX databases: "1200" or "2400" baud modem; "TERM"inal server; or "PCSA" DEC Pathworks. NOTE! This feature not implemented.

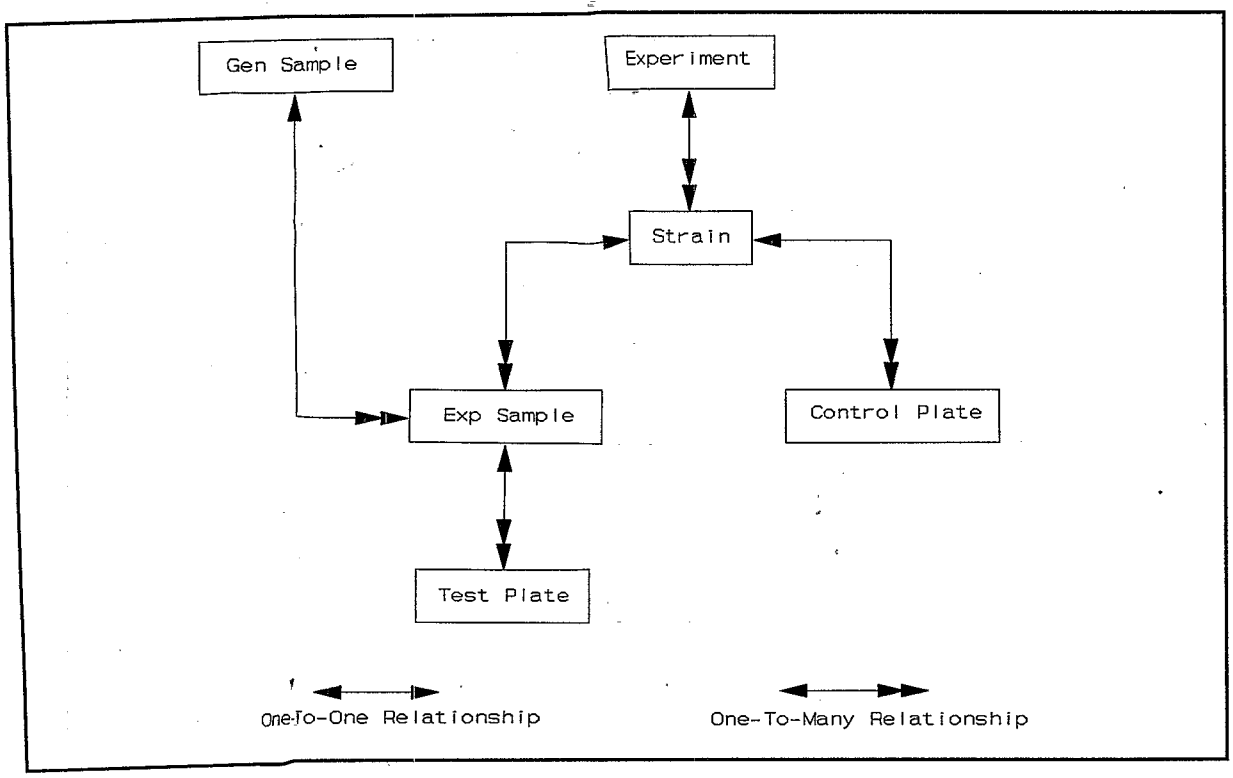
Default File	Variables	Remarks
G_Coords	Win_Coord,	Window coordinates and starting position for ACHOICE() of Win_Pos assays array.
G_ExpKey	T_ExpKeys	Contains next available experiment key.
G_Gene (encrypted).	Enc_Pas	Contains system password for access to Defaults-Set Special Options Menu
G_Instr	InstSelect	String containing filename of currently selected instruction set.
G_ItCnt	RdCnt	Number of times to count each plate when using colony counter interface.
G_Keys	S_Key	Next available suggestion index key value.
G_ModHel	ModHelp	Color String used when modifying system help.
G_NegKey	T_NegKey	Next available negative control key.
G_PosKey	T_Poskey	Next available positive control key.
G_SamKey	TSampKey	Next available sample key.
Proto	ProtoNam	Drive/Path for Protocol files, top level filename.
Sugg_Col	Sugg_Color	Color string used when displaying suggestion box text.
Version	G_Version	String indicating version information ("PC_" or "VAX").
QStead	(text file)	Presence of this file in the database directory causes GeneTox Manager to block execution of the Quasi-Stead model.

CHAPTER 6

DATABASE STRUCTURES

This chapter contains the logical and physical structures of the various databases used with the GeneTox Manager system.

LOGICAL STRUCTURE



Entity	Description
Gen Sample	General sample information independent of the experiment(s) in which the sample was run (G_Sample).
Experiment	General experiment information (G_GenDat).
Strain	Information about each strain used in the experiment (G_Expmnt).
Exp Sample	Experiment specific sample information related to the specific strain the sample was analyzed under (G_ExSmDt).
Control	Positive and negative controls for each strain used in the experiment.
Test Plate	Test plate for a particular dosage, 5 Plates/Dose (G_Test).
Control Plat	Positive or negative control plate, 5 Plates/Dose (G_Neg/PosDat).

The following sections contain the name, alias, and indices used with each database, as well as the name, type, width, number of decimal places and contents of each field in the database.

Name: G_GENDAT.DBF - general experiment information
Alias: GENERAL
Keys: ExpNam
Index: G_GENIND (ExpNam)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPNAM	Character	10		Experiment name
2	STATUS	Character	1		Upload status flag
3	RESLABID	Character	4		Research lab ID
4	EXPDATE	Date	8		Date conducted
5	TESTYPCOD	Numeric	2		Type of test conducted
6	S9TYPLOT	Character	6		S9 type and lot number
7	LABTECH	Character	4		Person conducting exp
8	TESTYPDES	Character	35		Test type description
9	TIMEMI	Numeric	3		Minutes
10	TIMEHR	Numeric	6	1	Hours
11	TEMPERATRC	Numeric	6	1	Temperature Celsius
12	TEMPERATRF	Numeric	6	1	Temperature Fahrenheit
13	LABBOOK	Character	25		Labbook number
14	ASYMODF	Character	35		Assay modification
15	ENTERDATE	Date	8		Date data entered
16	ENTRYTECH	Character	4		Person entering data
17	P_INVEST	Character	4		Principal Investigator
18	S9_DESC	Memo	10		S9 description
19	S9STRB	Numeric	1		Sterility before
20	S9STRA	Numeric	1		Sterility after
21	S9LEVEL1	Character	6		Activation level
22	S9LEVEL2	Character	6		Activation level
23	S9UNITS1	Character	4		Units of measurement
24	S9UNITS2	Character	4		Units of measurement

Name: G_SAMPLE.DBF - sample specific data
Alias: SAMPLE
Keys: SampKey
Index: G_SAKIND (Str(SampKey,9))
G_SAMIND (Samplabid+Str(GbbSmplyr,2)+Str(GbbSmpln,4)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	SAMPKEY	Numeric	9		Sample key
2	SM_STATUS	Character	1		Sample upload status
3	RESLABID	Character	4		Research lab ID
4	SAMPLABID	Character	4		Sample lab ID
5	GBBSMPLYR	Numeric	2		Year info recorded
6	GBBSMPLN	Numeric	4		Sample ID
7	DS_STATUS	Character	1		Dose upload status
8	SAMPDESC	Memo	10		Sample description
9	GS_STATUS	Character	1		Overall upload status
10	SC_BAT_LOT	Character	51		Sample source/ batch/lot
11	CAS_NUM	Character	15		Sample CAS number

Name: G_EXPMNT.DBF - experiment information
Alias: EXPMNT
Keys: ExpKeys, ExpNam
Index: G_EXPIND (Expnam+StrainAbr+Dtoc(StrnDate)+Str(ActLevel,1))
G_EXKIND (ExpKeys)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPKEYS	Numeric	9		Experiment key
2	STATUS	Character	1		Upload status flag
3	EXPNAM	Character	10		Experiment name
4	STRAINABR	Character	6		Strain abbreviation
5	STRNDATE	Date	8		Strain date
6	ACTLEVEL	Numeric	1		Activation level
7	CULTDNSY	Character	10		Culture density

Name: G_EXSMDT.DBF - experiment specific data
Alias: EXSPECIFIC
Keys: ExpKeys, SampKey
Index: G_EXSIND (Str(ExpKeys,9)+Str(SampKey,9))
G_ESSIND (Str(SampKey,9))
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPKEYS	Numeric	9		Experiment key
2	SAMPKEY	Numeric	9		Sample key
3	STATUS	Character	1		Upload status flag
4	PHENOCOD	Numeric	1		Pheno code check
5	SMPLSTRB	Numeric	1		Sterility before
6	SMPLSTRA	Numeric	1		Sterility after
7	DOSUNITS	Character	24		Units for dose
8	REMARK	Memo	10		Remark field
9	OLD_EXPKEY	Numeric	9		Trash field
10	SOLV_DESC	Character	25		Solvent description
11	SOLV_AMTPT	Numeric	4		Solvent amount
12	SOLV_UNITS	Character	3		Units of measure
13	INCUBAT_HR	Numeric	6	2	Hours of incubation

Name: G_TSTRN.DBF
Alias: TSTRAIN
Keys:
Index: G_TSKIND (G_SKey)
G_TSTIND (G_Strain+Dtoc(G_StrDate)+Str(G_ActLevel,1))
Type: TEMPORARY

Field	Field name	Type	Width	Dec	Contents
1	G_STRAIN	Character	6		Strain used
2	G_STRDATE	Date	8		Date
3	G_ACTLEVEL	Numeric	1		Activation level
4	G_DENSITY	Character	10		Culture density
5	G_EXPMNT	Character	1		Experiment name
6	G_EXSMDT	Character	1		Experiment date
7	G_SKEY	Numeric	9		Sample key

Name: G_HCNTRL.DBF
Alias: HCONTROL
Keys: G_Key, NPKey
Index: G_HCNIND (G_Key)
G_HCCIND (Str(G_Code,2)+Str(G_Amnt,9)+G_Unit)
Type: TEMPORARY

Field	Field name	Type	Width	Dec	Contents
1	G_KEY	Numeric	9		Experiment key
2	G_NPKEY	Numeric	9		Neg/Pos control key
3	G_STATUS	Character	1		Upload status flag
4	G_CODE	Numeric	2		Neg/Pos control code
5	G_NAME	Character	20		Name of control
6	G_AMNT	Numeric	6	2	Amount used
7	G_UNIT	Character	3		Unit of measurement
8	G_MEAN	Numeric	10	4	Calculated mean
9	G_STD	Numeric	9	4	Calculated standard dev
10	G_PLATE1	Numeric	5		Temp plate counts
11	G_BKGRND1	Character	1		Temp background code
12	G_PLATE2	Numeric	5		Temp plate counts
13	G_BKGRND2	Character	1		Temp background code
14	G_PLATE3	Numeric	5		Temp plate counts
15	G_BKGRND3	Character	1		Temp background code
16	G_PLATE4	Numeric	5		Temp plate counts
17	G_BKGRND4	Character	1		Temp background code
18	G_PLATE5	Numeric	5		Temp plate counts
19	G_BKGRND5	Character	1		Temp background code

Name: G_TCNTRL.DBF
Alias: TCONTROL
Keys: G_Key, G_NPKey
Index: G_TCNIND (G_Key)
G_TCCIND (Str(G_Code,2)+Str(G_Amnt,9)+G_Unit)
Type: TEMPORARY

Field	Field name	Type	Width	Dec	Contents
1	G_KEY	Numeric	9		Experiment key
2	G_NPKEY	Numeric	9		Neg/Pos control key
3	G_STATUS	Character	1		Upload status flag
4	G_CODE	Numeric	2		Neg/Pos control code
5	G_NAME	Character	20		Name of control
6	G_AMNT	Numeric	6	2	Amount used
7	G_UNIT	Character	3		Units of measurement
8	G_MEAN	Numeric	10	4	Calculated mean
9	G_STD	Numeric	9	4	Calculated standard dev
10	G_PLATE1	Numeric	5		Temp plate counts
11	G_BKGRND1	Character	1		Temp background code
12	G_PLATE2	Numeric	5		Temp plate counts

Field	Field name	Type	Width	Dec	Contents
13	G_BKGRND2	Character	1		Temp background code
14	G_PLATE3	Numeric	5		Temp plate counts
15	G_BKGRND3	Character	1		Temp background code
16	G_PLATE4	Numeric	5		Temp plate counts
17	G_BKGRND4	Character	1		Temp background code
18	G_PLATE5	Numeric	5		Temp plate counts
19	G_BKGRND5	Character	1		Temp background code

Name: G_NEGDAT.DBF - negative control data
Alias: NEGATIVE
Keys: ExpKeys, NegKey
Index: G_NEGIND (Str(ExpKeys,9)+Str(NegKey,9))
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPKEYS	Numeric	9		Experiment key
2	NEGKEY	Numeric	9		Negative control key
3	STATUS	Character	1		Upload status flag
4	NEGCODE	Numeric	2		Negative control code
5	NEGCTRLNA	Character	20		Negative control name
6	NEGAMT	Numeric	6	2	Negative control amount
7	NEGUNT	Character	3		Negative control units
8	NEGMEAN	Numeric	10	4	Negative control mean
9	NEGSTDEV	Numeric	9	4	Negative control sd
10	NEGPLTCNT1	Numeric	5		Negative plate counts
11	NEGBKGRND1	Character	1		Background code
12	NEGPLTCNT2	Numeric	5		Negative plate counts
13	NEGBKGRND2	Character	1		Background code
14	NEGPLTCNT3	Numeric	5		Negative plate counts
15	NEGBKGRND3	Character	1		Background code
16	NEGPLTCNT4	Numeric	5		Negative plate counts
17	NEGBKGRND4	Character	1		Background code
18	NEGPLTCNT5	Numeric	5		Negative plate counts
19	NEGBKGRND5	Character	1		Background code

Name: G_POSDAT.DBF - positive control data
Alias: POSITIVE
Keys: ExpKeys, PosKey
Index: G_POSIND (Str(ExpKeys,9)+Str(PosKey,9))
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPKEYS	Numeric	9		Experiment key
2	POSKEY	Numeric	9		Positive control key
3	STATUS	Character	1		Upload status flag
4	POSCODE	Numeric	2		Positive control code
5	POSCTRLNAM	Character	20		Positive control name
6	POSAMT	Numeric	6	2	Positive control amount
7	POSUNT	Character	3		Positive control units
8	POSMEAN	Numeric	10	4	Positive control mean
9	POSSTDEV	Numeric	9	4	Positive control sd
10	POSPLTCNT1	Numeric	5		Positive plate counts
11	POSBKGRND1	Character	1		Background code
12	POSPLTCNT2	Numeric	5		Positive plate counts
	POSBKGRND2	Character	1		Background code
14	POSPLTCNT3	Numeric	5		Positive plate counts
15	POSBKGRND3	Character	1		Background code
16	POSPLTCNT4	Numeric	5		Positive plate counts
17	POSBKGRND4	Character	1		Background code
18	POSPLTCNT5	Numeric	5		Positive plate counts
19	POSBKGRND5	Character	1		Background code

Name: G_TEST.DBF - test dose data
Alias: TEST
Keys: ExpKeys, SampKey
Index: G_TESIND (Str(ExpKeys,9)+Str(SampKey,9))
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPKEYS	Numeric	9		Experiment key
2	SAMPKEY	Numeric	9		Sample key
3	TD_STATUS	Character	1		Test dose upload status
4	DOSE	Numeric	9	3	Dose amount used
5	TESTMEAN	Numeric	10	4	Test mean
6	TESTDEV	Numeric	10	4	Test standard dev
7	TP_STATUS	Character	1		Test plate upload status
8	TSTPLTCNT1	Numeric	5		Test plate counts
9	TSTBKGRND1	Character	1		Background code
10	TSTPLTCNT2	Numeric	5		Test plate counts
11	TSTBKGRND2	Character	1		Background code
12	TSTPLTCNT3	Numeric	5		Test plate counts
13	TSTBKGRND3	Character	1		Background code
14	TSTPLTCNT4	Numeric	5		Test plate counts
15	TSTBKGRND4	Character	1		Background code
16	TSTPLTCNT5	Numeric	5		Test plate counts
17	TSTBKGRND5	Character	1		Background code

Name: G_DOSE.DBF - dose response data
Alias:
Keys: G_Dose
Index: None
Type: TEMPORARY

Field	Field name	Type	Width	Dec	Contents
1	G_DOSE	Numeric	9	3	Dose amount
2	G_DOSEPLT1	Numeric	5		Plate counts
3	G_DOSEBKG1	Character	1		Background code
4	G_DOSEPLT2	Numeric	5		Plate counts
5	G_DOSEBKG2	Character	1		Background code
6	G_DOSEPLT3	Numeric	5		Plate counts
7	G_DOSEBKG3	Character	1		Background code
8	G_DOSEPLT4	Numeric	5		Plate counts
9	G_DOSEBKG4	Character	1		Background code
10	G_DOSEPLT5	Numeric	5		Plate counts
11	G_DOSEBKG5	Character	1		Background code

Name: G_SUGG1.DBF
Alias: SUGG1
Keys: Var_Name
Index: G_SUGG1 (Var_Name)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	VAR_NAME	Character	10		Variable name
2	LIST	Logical	1		Selectable/Help text flag
3	SUGG_INDEX	Numeric	5		Suggestion key
4	NUM_SUGG	Numeric	5		Suggestion box number
5	REQUIRED	Logical	1		Required field tag
6	VERIFIED	Logical	1		Verified field tag

Name: G_SUGG2.DBF
Alias: SUGG2
Keys: Sugg_Index
Index: G_SUGG2 (Sugg_Index)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	SUGG_INDEX	Numeric	5		Suggestion key
2	SUGGESTION	Character	48		String displayed
3	RETURN_VAL	Character	48		Selectable field return value

Name: G_SEP.DBF - used for the raw data report
Alias: SEPERATE
Index: G_SEPIND (Sep_Nam)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	SEP_NAM	Character	10		Flagged names for raw data report

Name: G_PINVST.DBF - principal investigator list
Alias: PINVST
Keys: P_Invest
Index: G_PININD (P_Invest)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	P_INVEST	Character	4		Principal investigator

Name: G_PITMP.DBF - principal investigator modification
Alias: PITMP
Keys: G_PI
Index: None
Type: TEMPORARY

Field	Field name	Type	Width	Dec	Contents
1	G_PI	Character	4		Principal investigator

Name: G_MODSUG.DBF - suggestion modification
Alias: MODSUG
Keys: Sugg_Index
Index: None
Type: TEMPORARY

Field	Field name	Type	Width	Dec	Contents
1	SUGG_INDEX	Numeric	5		Suggestion key
2	SUGGESTION	Character	48		String displayed
3	RETURN_VAL	Character	48		Selectable field return value

Name: IVHELP.DBF
Alias:
Keys: ProcName
Index: IVHELP (ProcName)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	PROCNAME	Character	10		Procedure name (not presently used)
2	HELPTEXT	Memo	10		Help text shown

Name: GTMDIST.DBF
Alias:
Keys: DistKey
Index: G_DSTIND (DistKey)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	DISTKEY	Numeric	9		Distribution Key (Millions number for all keys generated by this GeneTox distribution copy)
2	DISTSITE	Character	35		Person and/or place to receive GeneTox distribution copy

Name: GTMARCHV.DBF
Alias: ArchiveDB
Keys: ExpNam
Index: G_ARCIND (ExpNam)
Type: STORAGE

Field	Field name	Type	Width	Dec	Contents
1	EXPNAM	Character	10		Experiment name
2	PASSWORD	Character	20		Archive Password
3	DRIVE	Character	1		Drive Letter of Archive Location
4	PATH	Character	24		DOS Path of Archive Location
5	FILENAME	Character	12		Archive File DOS Filename
6	ARCHDATE	Date	8		Date Experiment Archived

CHAPTER 7

TRANSFER FILE FORMATS

GeneTox manager uses flat file formats for storing experiment and sample data for backup, uploading, archiving, and sharing files between machines. Two distinct file formats exist: the "old" file format (particular to the database structures before version 2.2) and the "new" format (used in version 2.2). GeneTox has the ability to read transfer files of either type, but generates files only in the "new" format. New format files may be identified by a character "N" in the 56th column of the STO1 header record.

Record Type ST01 - Required with all associated records below for EACH experiment/strain combination.

Col	Field	Data Type	Len	Description	
1-4	RecType	Character	4	ST01 - Master record for an experiment strain	
5-13	ExpKeys	Integer	9	Experiment strain key	
14	RecStat	Character	1	Data record status in PC databases: (A)ctive, (I)nactive, (M)odified, (R)ead Only,(V)ax Download, (X)Archived	
15-24	ExpNam	Character	10	Experiment Name	
25-30	StrnAbr	Character	6	Strain Abbreviation	
31-36	StrnDate	Date	6	Strain Date (YYMMDD)	
37	ActLevel	Character	1	Activation Level	
38	GenRec	Character	1	General Experiment Info included (Y/N)	
39-41	NumSmp	Integer	2	Number of Samples included for this experiment strain	analysis
42-43	NumNeg	Integer	2	Number of negative controls for this experiment strain	analysis
44-45	NumPos	Integer	2	Number of positive controls for this experiment strain	analysis
46-55	CultDnsy	Character	10	Culture Density	
** New File Format Only **					
56	TFFormat	Character	1	Transfer file format flag (N = new)	

End-of-record

Record Type GE01 - Required if GenRec field in ST01 record = Y.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	GE01 - General experiment information - Rec 1
5-13	ExpKeys	Integer	9	Experiment strain key
14	RecStat	Character	1	Data record status in PC databases: (A)ctive, (I)nactive, (M)odified, (R)ead Only, (V)ax Download, (X)Archived
15-20	ExpDate	Date	6	Experiment Date (YYMMDD)
21-22	TesTypCod	Integer	2	Test type code
23-28	S9TypLot	Character	6	S9 Type and Lot
29-32	LabTech	Character	4	Lab Technician initials
33-67	TesTypDes	Character	35	Test type description

End-of-record

Record Type GE02 - Required if GenRec field in ST01 record = Y.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	GE02 - General experiment information - Rec 2
5-13	ExpKeys	Integer	9	Experiment strain key
14-16	TimeMi	Integer	3	Experiment time in minutes
17-22	TimeHr	Floating	6.1	Experiment time in hours
23-28	TempF	Floating	6.1	Temperature in Farenheit
29-34	TempC	Floating	6.1	Temperature in Centigrade
35-59	LabBook	Character	25	LabBook Number
60-63	ResLabID	Character	4	Research Lab ID

End-of-record

Record Type GE03 - Required if GenRec field in ST01 record = Y.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	GE03 - General experiment information - Rec 3
5-13	ExpKeys	Integer	9	Experiment strain key
14-48	AsyModf	Character	35	Assay Modification
49-54	EnDate	Date	6	Data entry date (YYMMDD)
55-58	P_Invest	Character	4	Principal investigator initials
59-62	EntryTech	Character	4	Data entry technician initials

End-of-record

Record Type GE04 - Required if GenRec field in ST01 record = Y.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	GE04 - General experiment information - Rec 4
5-13	ExpKeys	Integer	9	Experiment strain key
14	S9StrB	Integer	1	S9 sterility before
15	S9StrA	Integer	1	S9 sterility after
16-21	S9Level1	Character	1	S9 level 1
22-27	S9Level2	Character	1	S9 level 2
28-31	S9Units1	Character	4	S9 units 1
32-35	S9Units2	Character	4	S9 units 2
36	S9Used	Character	1	S9 used (+/-)
37-40	NmS9Desc	Integer	4	Number of S9 description records to follow (i.e., number of GE05s)

End-of-record

Record Type GE05 - Number of GE05 records must correspond to value in NmS9Desc field in GE04 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	GE05 - General experiment information - Rec 5
5-13	ExpKeys	Integer	9	Experiment strain key
14-78	S9Desc	Character	65	S9 description line

End-of-record

Record Type NP01 - Number of NP01/NP02 record sets must correspond to value in NumNeg field in ST01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	NP01 - Negative control general information
5-13	NegKey	Integer	9	Negative control key
14	RecStat	Character	1	Data record status in PC databases: (A)ctive, (I)nactive, (M)odified, (R)ead Only, (V)ax Download, (X)Archived
15-16	NegCode	Integer	2	Negative control code
17-36	NegName	Character	20	Negative control name
37-45	NegAmt	Floating	9.3	Negative control amount
46-48	NegUnit	Character	3	Negative control units
49-58	NegMean	Floating	10.4	Negative control mean
59-68	NegStDev	Floating	10.4	Negative control standard deviation
69	NmNegCnt	Integer	1	Number of negative control plate counts to follow in NP02 record

End-of-record

Record Type NP02 - Number of NP01/NP02 record sets must correspond to value in NumNeg field in ST01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	NP02 - Negative control plate counts
5-13	NegKey	Integer	9	Negative control key
14-18	NegPlCnt1	Integer	5	Plate count 1
19	NegBkgnd1	Character	1	Background code 1
20-24	NegPlCnt2	Integer	5	Plate count 2
25	NegBkgnd2	Character	1	Background code2
26-30	NegPlCnt3	Integer	5	Plate count 3
31	NegBkgnd3	Character	1	Background code 3
32-36	NegPlCnt4	Integer	5	Plate count 4
37	NegBkgnd4	Character	1	Background code 4
38-42	NegPlCnt5	Integer	5	Plate count 5
43	NegBkgnd5	Character	1	Background code 5

End-of-record

Record Type PP01 - Number of PP01/PP02 record sets must correspond to value in NumPos field in ST01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	PP01 - Positive control general information
5-13	PosKey	Integer	9	Positive control key
14	RecStat	Character	1	Data record status in PC databases: (A)ctive, (I)nactive, (M)odified, (R)ead Only, (V)ax Download, (X)Archived
15-16	PosCode	Integer	2	Positive control code
17-36	PosName	Character	20	Positive control name
37-45	PosAmt	Floating	9.3	Positive control amount
46-48	PosUnit	Character	3	Positive control units
49-58	PosMean	Floating	10.4	Positive control mean
59-68	PosStDev	Floating	10.4	Positive control standard deviation
69	NmPosCnt	Integer	1	Number of positive control plate counts to follow in PP02 record

End-of-record

Record Type PP02 - Number of PP01/PP02 record sets must correspond to value in NumPos field in ST01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	PP02 - Positive control plate counts
5-13	PosKey	Integer	9	Positive control key
14-18	PosPlCnt1	Integer	5	Plate count 1
19	PosBkgnd1	Character	1	Background code 1
20-24	PosPlCnt2	Integer	5	Plate count 2
25	PosBkgnd2	Character	1	Background code 2
26-30	PosPlCnt3	Integer	5	Plate count 3
31	PosBkgnd3	Character	1	Background code 3
32-36	PosPlCnt4	Integer	5	Plate count 4
37	PosBkgnd4	Character	1	Background code 4
38-42	PosPlCnt5	Integer	5	Plate count 5
43	PosBkgnd5	Character	1	Background code 5

End-of-record

Record Type SM01 - Number of SM01 record sets must correspond to value in NumSmp field in ST01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM01 - Sample analysis header information
5-13	SampKey	Integer	9	Sample key
14	RecStat	Character	1	Data record status in PC databases: (A)ctive, (I)nactive, (M)odified, (R)ead Only, (V)ax Download, (X)Archived
15-18	Filler			Not currently used
19-22	SmpLabID	Character	4	Sample lab ID
23-24	SmpYr	Integer	2	Sample year
25-28	SmpNo	Integer	4	Sample number
29	PhenoCod	Integer	1	Phenocode check
30	SmplStrB	Integer	1	Sample sterility before
31	SmplStrA	Integer	1	Sample sterility after
32-55	DosUnits	Character	24	Dose units
56-57	NumDose	Integer	2	Number of doses
58	GenSm	Character	1	General sample information included (Y/N) (i.e., do SM03-SM04 records follow?)
59-62	RemCt	Integer	4	Number of remarks records that follow (SM05s)
63-66	NumSDc	Integer	4	Number of sample description records that follow (SM03)
67-75	OldKey	Integer	9	Old sample key if the SampKey field is a replacement

End-of-record

In the "old" transfer file format, SM02, SM03 and SM04 records have the following format:

Record Type SM02 - Number of SM02 records must correspond to value in NumSDc field in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM02 - Sample description records
5-13	SampKey	Integer	9	Sample key
14-64	SampDesc	Character	51	Sample description line

End-of-record

Record Type SM03 - Required if GenSm field = Y in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM03 - General sample information
5-13	SampKey	Integer	9	Sample key
14-38	SolvDesc	Character	25	Solvent description
39-42	SolvAmtPt	Integer	4	Solvent amount per plate
43-45	SolvUnits	Character	3	Solvent units
46-60	CasNum	Character	15	CAS number

End-of-record

Record Type SM04 - Required if GenSm field = Y in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM04 - General sample information
5-13	SampKey	Integer	9	Sample key
14-64	Sc_Bat_Lot	Character	51	Source/Batch/Lot
65-70	IncubatHr	Floating	6.2	Incubation hours

End-of-record

In the "new" transfer file format SM02, SM03, and SM04 records have the following format:

Record Type SM02 - Required for each SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM02 - Experiment specific sample information
5-13	SampKey	Integer	9	Sample key
14-38	SolvDesc	Character	25	Solvent description
39-42	SolvAmtPt	Integer	4	Solvent amount per plate
43-45	SolvUnits	Character	3	Solvent units
46-51	IncubatHr	Floating	6.2	Incubation hours

End-of-record

Record Type SM03 - Number of SM03 records must correspond to value in NumSDc field in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM03 - Sample description records
5-13	SampKey	Integer	9	Sample key
14-64	SampDesc	Character	51	Sample description line

End-of-record

Record Type SM04 - Required if GenSm field = Y in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM04 - General sample information
5-13	SampKey	Integer	9	Sample key
14-64	Sc_Bat_Lot	Character	51	Source/Batch/Lot
65-79	CasNum	Character	15	CAS number

End-of-record

Record Type SM05 - Number of SM05 records must correspond to value in RemCt field in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	SM05 - Sample remarks records
5-13	SampKey	Integer	9	Sample key
14-73	Remark	Character	60	Sample remark line

End-of-record

Record Type TD01 - Number of TD01/TD02 record sets must correspond to value in NumDose field in SM01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	TD01 - Dose analysis information
5-13	ExpKeys	Integer	9	Experiment strain key
14-22	SampKey	Integer	9	Sample key
23	RecStat	Character	1	Data record status in PC databases: (A)ctive, (I)nactive, (M)odified, (R)ead Only, (V)ax Download, (X)Archived
24-32	Dose	Floating	9.3	Dose amount
33-42	TestMean	Floating	10.4	Test dose mean
43-52	TestDev	Floating	10.4	Test dose standard deviation
53	TestCnt	Integer	1	Number of test plate counts to follow in TD02 record

End-of-record

Record Type TD02 - Number of TD01/TD02 record sets must correspond to value in NumDose field in ST01 record.

Col	Field	Data Type	Len	Description
1-4	RecType	Character	4	TD02 - Test plate counts
5-13	SampKey	Integer	9	Sample key
14-18	TstPlCnt1	Integer	5	Plate count 1
19	TstBkgnd1	Character	1	Background code 1
20-24	TstPlCnt2	Integer	5	Plate count 2
25	TstBkgnd2	Character	1	Background code 2
26-30	TstPlCnt3	Integer	5	Plate count 3
31	TstBkgnd3	Character	1	Background code 3
32-36	TstPlCnt4	Integer	5	Plate count 4
37	TstBkgnd4	Character	1	Background code 4
38-42	TstPlCnt5	Integer	5	Plate count 5
43	TstBkgnd5	Character	1	Background code 5

End-of-record

CHAPTER 8

POSSIBLE ENHANCEMENTS

1. Modify GeneTox Manager code and databases to allow for replication of Samples within an Experiment. The current database structure is:

```
Exp ===>          Strain ===>          ExpSamp ===> Test Dose
                        ^
GenSamp ==============+
```

(where GenSamp is general sample information shared between Experiments, and ExpSamp is experiment specific sample information.)

The altered structure could be (for example):

```
Exp ===>          Strain ===>  ExpSamp ===>  RepSamp ===> Test Dose
                        ^
GenSamp ==============+
```

(where GenSamp is general sample information shared between Experiments, ExpSamp is experiment specific sample information, and RepSamp is information specific to each replication of the sample in the experiment.)

Once this is done, GeneTox Manager spiral data import code could be modified to save Spiral Systems multiple test data (tests A-D) under the same Experiment name and Sample ID (with replications of the sample) rather than requiring the user to choose a different Experiment name for each test.

2. The Defaults menu option "Communications Setup" could be fully implemented. The current choices include 1200/2400 baud modem, terminal server, or DEC Pathworks (PCSA). The terminal server and PCSA options have been implemented for the HERL site, although both are not used interchangeably. The terminal server connections was used prior to networking PCs with PCSA. PCSA is now used exclusively. However, both terminal server and PCSA connection practices for GeneTox Manager have been created.

For terminal server connections, the XTALK communications program was used. Connection was automated using script files. Specific script files were created for the following operations: Execute VAX GeneTox Manager System (VAXONLY); Upload New Data to VAX Databases

(VAXT); Upload Modified Data to VAX Databases (VAXM); and Download VAX Data to Databases (VAXDOWN). The XTALK .XTK files contain connection settings, and all these files were identical, while the .XTS files contain connection commands. XTALK could probably be used for MODEM (1200/2400 baud) connections as well, although this was not implemented for GeneTox Manager at this site.

For connections through DEC Pathworks (PCSA), the SETHOST terminal emulation program that comes with Pathworks was used. This program provides better VAX terminal emulation than XTALK. Connection was again automated using script files. Specific script files were created for the following operations: Execute VAX GeneTox Manager System (GENETOX); Upload New Data to VAX Databases (UPLNEW); Upload Modified Data to VAX Databases (UPLMOD); and Download VAX Data to Databases (DNLOAD). These SETHOST .SRC files contain connection settings and commands.

The VAX Databases for the HERL site were FOCUS databases. Data was uploaded/downloaded in Transfer File Format (see Programmer's Guide, Chapter 7 for details). GeneTox Manager users may obtain copies of the FOCUS database structures and code, and the FORTRAN source code used for statistical models, from Dr. Claxton on request. Users may also wish to use other mainframe or mini-computer databases for storing long term data. The details of the Transfer File Format are provided for this purpose. Users will have to modify the communications files to execute the specific upload/download procedures used at their site.

GeneTox Manager could be modified, for distribution, to use one of several communication methods, and to allow the user to select between them, as the "Communication Setup" option seems to suggest. One could also modify GeneTox Manager to allow users to save the name of the communication program called and the script files used in Clipper default files, rather than hardwiring them into the code, which would allow users to tailor the communications system (from the PC side) without reprogramming.

3. GeneTox Manager source code could be reorganized into larger files where the individual procedures are specifically named and grouped according to function. Currently, many small .PRG files exist and these files correspond to the procedure names of the procedures they contain. Reorganization would necessitate rewriting the system make file (GTM.RMK) as well. CA-Clipper technicians have recommend that .OBJ files produced for use in CLIPPER 5.0 executables be 64K or less (the swap segment size of the VMM). GeneTox Manager as currently organized and compiled does not conform to that recommendation.

4. A system similar to GeneTox Manager but specifically tailored to Mutation Frequency calculations (named MUTAGEN FREQUENCY) exists. This system is almost identical to GeneTox Manager. Mutagen Frequency will additionally store survivor as well as mutations data for each control/sample, and produces a mutation frequency calculations report. It also allows for more that five plate counts per dose, up to maximum of 100 plates and 50 doses. This feature has not been brought into GeneTox Manager, although it could be added (without adding the mutation frequency specific parts). Also, most of the GeneTox Manager code is duplicated for use in the Mutagen Frequency application with few changes or additions. However, two separate copies of the code exist, and both copies must be updated for bug fixes, changes, etc. These two applications could be merged into one source code pool, with minor changes to the current code so that a majority of the code is exactly the same for each application. This would be especially true if the additional "unlimited" plates per dose feature was brought into GeneTox Manager. Some changes to the code to isolate application specific routines would be necessary.

5. Currently the GeneTox Manager system backs up data by saving it in Transfer File Format to the database directory. There are no other provisions for backing up GeneTox Manager data. Users may backup database data using the DOS BackUp utility. Files with extensions of .DBF and .DBT contain experiment data which users may want to back up. .DFT files contain system default settings, and users may also want to back these up. A strategy for implementing automatic backups as part of the GeneTox Manager system was developed, but never implemented (see below).

Automatic Backups for GeneTox Manager:

-- It would eliminate the current practice of saving flat file backups with each data entry/modification and upload. (A flat file will still have to be generated for uploading/exporting/archiving, however. This can't be helped.)

-- Instead we would provide a "Backup" utility that would backup the current databases, indexes, and system default settings all at the same time. This utility could be accessed manually (probably from the Utilities-PC Functions menu), or be automated. A manual "Restore" utility would also be provided to recover to the last backup (probably from the Defaults-Set Special Options menu).

-- Automated backups could be controlled from the Special Options menu also. The user could specify the drive [and path] to backup to, whether to automatically backup, and the frequency of backups if automated. The settings menu would look like this:

GENETOX DATA BACKUP SETTINGS

Drive: _ [Path: _____]
Automatic Backups (Y/N) _
Backup Every __ Day(s) (0-99)

-- Backups could be done at either entry or exit of the program. Specifying a backup every 0 days would cause the system to backup every time the program was run.

-- The backup will backup only modified files. However, since almost all files are affected by entry of new data, the majority of the files would be backed up each time the backup was run, assuming at least one experiment was entered/modified during the backup interval.

-- Using PKZip to "write" the backup utility. PKZip compresses files and allows specification of a path (and filename).

-- Testing of backups on large sample databases machine yielded the following results:

	Time*	Disk Space Used**	Compression Ratio	
PKZip		3:30	0.28Mb	89%

* times are in minutes:seconds and are approximated
** disk space is measured in Megabytes and rounded to the nearest 0.01Mb

6. The current distribution strategy for GeneTox Manager is to provide the means to generate additional distribution diskettes with each copy of the system (GTMLoad.EXE). Other distributors may not wish to allow users to generate additional distribution copies, and may wish to modify the load/install code accordingly.

7. GeneTox Manager code could be modified to make it networkable. CLIPPER contains all the necessary procedures for implementing a shared file database system (file locking, record locking, etc.), but the GeneTox Manager code does not use these facilities. With minor additions to the use of databases during data writes, a network version of the system could be produced.

NOTE: See also Appendix E of the GeneTox Manager User's Guide, "Known Problems and Limitations" for details of other problems and limitations within GeneTox Manager that may be corrected.

CHAPTER 9

SNAP! DIAGRAMS

The following tree diagram has been created using the SNAP! shareware mentioned earlier in this document. The documentation for SNAP! is located on the PC in the SNAP subdirectory. The programmer can also retrieve other useful outputs through SNAP!.

System: GeneTox Mutagen Frequency System
Author: CSC/EPA
06/24/93 15:29:30

Tree Diagram

```
G_MENU.PRG
+----HELP.PRG
|   +----HELP.PRG
*** Truncating tree for recursive call to HELP.PRG
+----G_VPUBL.PRG
|   +----G_INTCLR.PRG
+----G_WELCOM.PRG
+----G_PROTO.PRG
|   +----G_PROTO.PRG
*** Truncating tree for recursive call to G_PROTO.PRG
+----G_TERMHE.PRG
+----G_SELSUG.PRG
+----G_GESCRN.PRG
+----G_MENTER.PRG
|   +----G_ENTER.PRG
|   |   +----G_INSTFV.PRG
|   |   +----G_PROTO.PRG
|   |   |   +----G_PROTO.PRG
*** Truncating tree for recursive call to G_PROTO.PRG
|   |   +----G_TERMHE.PRG
|   |   +----SET_INST.PRG
|   |   +----G_SCNTRL.PRG
|   |       +----G_EXTRNL.PRG
|   |           +----G_EXPMNT.PRG
|   |           |   +----G_EXINIT.PRG
|   |           |   +----G_EXSCRN.PRG
|   |           |   +----G_SUGON.PRG
|   |           |   |   +----EDIT_SUG.PRG
|   |           |   |   +----EXPAND_SUG.PRG
|   |           |   |   +----UP_BOX.PRG
|   |           |   |   +----DOWN_BOX.PRG
|   |           |   +----G_SELSUG.PRG
|   |           |   +----MOD_SAVE.PRG
|   |           |   +----G_SUGOFF.PRG
|   |           +----G_ASSAY.PRG
```

```
|   |           |   +---G_ASSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           +---G_STRAIN.PRG
|   |           |   +---FAKE_LEFT.PRG
|   |           |   +---G_STSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           |   +---STRN_MOVE.PRG
|   |           |   +---G_MSVSTN.PRG
|   |           |   |   +---G_SELSUG.PRG
|   |           |   +---G_SUGOFF.PRG
|   |           +---G_DILFAC.PRG
|   |           |   +---FAKE_LEFT.PRG
|   |           |   +---CHANGESTRAIN (procedure in G_DILFAC.PRG)
|   |           |   |   +---STRAINLIST.PRG
|   |           |   |   +---CHANGESTRAIN (procedure in G_DILFAC.PRG)
*** Truncating tree for recursive call to CHANGESTRAIN
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---SET_INST.PRG
|   |           |   |   +---G_PROTO.PRG
|   |           |   |       +---G_PROTO.PRG
*** Truncating tree for recursive call to G_PROTO.PRG
|   |           |   +---G_DFSCRN.PRG
|   |           |   +---STRAINLIST.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           +---G_NPMUT.PRG
|   |           +---G_NPSRV.PRG
|   |           +---G_ACTSYS.PRG
|   |           |   +---ENTER_COMM.PRG
|   |           |   +---G_ACINIT.PRG
|   |           |   +---G_ACSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           +---G_SMPSPC.PRG
|   |           |   +---G_SMSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---ENTER_COMM.PRG
```

```
|  |        |  +----MOD_SAVE.PRG
|  |        |  +----G_SUGOFF.PRG
|  |        +----G_CURSTR.PRG
|  |        |  +----G_CUSCRN.PRG
|  |        |  +----STRAINLIST.PRG
|  |        |  +----G_SUGON.PRG
|  |        |  |  +----EDIT_SUG.PRG
|  |        |  |  +----EXPAND_SUG.PRG
|  |        |  |  +----UP_BOX.PRG
|  |        |  |  +----DOWN_BOX.PRG
|  |        |  +----G_SELSUG.PRG
|  |        |  +----CUR_STRAIN.PRG
|  |        |  +----MOD_SAVE.PRG
|  |        |  +----G_SUGOFF.PRG
|  |        +----G_QUALTY.PRG
|  |        |  +----G_QUSCRN.PRG
|  |        |  +----G_SUGON.PRG
|  |        |  |  +----EDIT_SUG.PRG
|  |        |  |  +----EXPAND_SUG.PRG
|  |        |  |  +----UP_BOX.PRG
|  |        |  |  +----DOWN_BOX.PRG
|  |        |  +----G_SELSUG.PRG
|  |        |  +----MOD_SAVE.PRG
|  |        +----G_MUTDOSE.PRG
|  |        +----G_SRVDOSE.PRG
|  |        +----G_REMARK.PRG
|  |        |  +----G_SUGON.PRG
|  |        |  |  +----EDIT_SUG.PRG
|  |        |  |  +----EXPAND_SUG.PRG
|  |        |  |  +----UP_BOX.PRG
|  |        |  |  +----DOWN_BOX.PRG
|  |        |  +----G_SELSUG.PRG
|  |        |  +----G_RESCRN.PRG
|  |        |  +----APNDINST (procedure in G_REMARK.PRG)
|  |        |  |  +----G_INSTFV.PRG
|  |        |  |  +----G_APPEND (procedure in G_REMARK.PRG)
|  |        |  |  +----G_INVIEW.PRG
|  |        |  +----MOD_SAVE.PRG
|  |        +----G_CONTNU.PRG
|  |           +----G_SAVEAL.PRG
|  |           |  +----G_SAVEXP.PRG
|  |           |  +----G_SAVSMP.PRG
|  |           |  +----G_SAVGEN.PRG
|  |           |  +----G_SAVNP.PRG
|  |           |  |  +----G_QUITAL.PRG
|  |           |  +----G_SAVSMD.PRG
|  |           |  |  +----G_QUITAL.PRG
|  |           |  +----G_SAVTST.PRG
|  |           |  |  +----G_QUITAL.PRG
|  |           |  +----G_SELSUG.PRG
|  |           +----G_SAVNEW.PRG
|  |           |  +----G_SAVSMP.PRG
|  |           |  +----G_SAVSMD.PRG
|  |           |  |  +----G_QUITAL.PRG
|  |           |  +----G_SAVTST.PRG
|  |           |     +----G_QUITAL.PRG
|  |           +----G_VAXFLE.PRG
|  |              +----G_CHKCNT.PRG
|  |              +----G_GENREC.PRG
```

```
|   |                 |   +---G_RPPAGE.PRG
|   |                 +---G_GENNPS.PRG
|   |                     +---G_CHKTPN.PRG
|   |                     +---G_GENREC.PRG
|   |                     |   +---G_RPPAGE.PRG
|   |                     +---G_GENTST.PRG
|   +---G_MODIFY.PRG
|   |   +---G_SUGON.PRG
|   |   |   +---EDIT_SUG.PRG
|   |   |   +---EXPAND_SUG.PRG
|   |   |   +---UP_BOX.PRG
|   |   |   +---DOWN_BOX.PRG
|   |   +---G_SELSUG.PRG
|   |   +---G_INSTFV.PRG
|   |   +---G_PROTO.PRG
|   |   |   +---G_PROTO.PRG
*** Truncating tree for recursive call to G_PROTO.PRG
|   |   +---FAKE_LEFT.PRG
|   |   +---G_MOGEN.PRG
|   |   |   +---G_MOGINT.PRG
|   |   +---G_MOCNTL.PRG
|   |   |   +---G_MOCINT.PRG
|   |   +---G_MOSMPL.PRG
|   |   |   +---G_MOSINT.PRG
|   |   |       +---G_SUGON.PRG
|   |   |       |   +---EDIT_SUG.PRG
|   |   |       |   +---EXPAND_SUG.PRG
|   |   |       |   +---UP_BOX.PRG
|   |   |       |   +---DOWN_BOX.PRG
|   |   |       +---G_SELSUG.PRG
|   |   |       +---PRNT_STRAI.PRG
|   |   |       +---CUR_STRAIN.PRG
|   |   +---G_MODOSE.PRG
|   |   |   +---G_SUGON.PRG
|   |   |   |   +---EDIT_SUG.PRG
|   |   |   |   +---EXPAND_SUG.PRG
|   |   |   |   +---UP_BOX.PRG
|   |   |   |   +---DOWN_BOX.PRG
|   |   |   +---G_SELSUG.PRG
|   |   |   +---PRNT_STRAI.PRG
|   |   |   +---CUR_STRAIN.PRG
|   |   +---G_ADSMPL.PRG
|   |   |   +---G_ADSINT.PRG
|   |   |       +---G_SUGON.PRG
|   |   |       |   +---EDIT_SUG.PRG
|   |   |       |   +---EXPAND_SUG.PRG
|   |   |       |   +---UP_BOX.PRG
|   |   |       |   +---DOWN_BOX.PRG
|   |   |       +---G_SELSUG.PRG
|   |   |       +---CUR_STRAIN.PRG
|   |   +---G_DELSMP.PRG
|   |   |   +---G_SUGON.PRG
|   |   |   |   +---EDIT_SUG.PRG
|   |   |   |   +---EXPAND_SUG.PRG
|   |   |   |   +---UP_BOX.PRG
|   |   |   |   +---DOWN_BOX.PRG
|   |   |   +---G_SELSUG.PRG
|   |   |   +---PRNT_STRAI.PRG
|   |   |   +---CUR_STRAIN.PRG
```

```
|   |   +----G_SCNTRL.PRG
|   |       +----G_EXTRNL.PRG
|   |           +----G_EXPMNT.PRG
|   |           |   +----G_EXINIT.PRG
|   |           |   +----G_EXSCRN.PRG
|   |           |   +----G_SUGON.PRG
|   |           |   |   +----EDIT_SUG.PRG
|   |           |   |   +----EXPAND_SUG.PRG
|   |           |   |   +----UP_BOX.PRG
|   |           |   |   +----DOWN_BOX.PRG
|   |           |   +----G_SELSUG.PRG
|   |           |   +----MOD_SAVE.PRG
|   |           |   +----G_SUGOFF.PRG
|   |           +----G_ASSAY.PRG
|   |           |   +----G_ASSCRN.PRG
|   |           |   +----G_SUGON.PRG
|   |           |   |   +----EDIT_SUG.PRG
|   |           |   |   +----EXPAND_SUG.PRG
|   |           |   |   +----UP_BOX.PRG
|   |           |   |   +----DOWN_BOX.PRG
|   |           |   +----G_SELSUG.PRG
|   |           |   +----MOD_SAVE.PRG
|   |           +----G_STRAIN.PRG
|   |           |   +----FAKE_LEFT.PRG
|   |           |   +----G_STSCRN.PRG
|   |           |   +----G_SUGON.PRG
|   |           |   |   +----EDIT_SUG.PRG
|   |           |   |   +----EXPAND_SUG.PRG
|   |           |   |   +----UP_BOX.PRG
|   |           |   |   +----DOWN_BOX.PRG
|   |           |   +----G_SELSUG.PRG
|   |           |   +----MOD_SAVE.PRG
|   |           |   +----STRN_MOVE.PRG
|   |           |   +----G_MSVSTN.PRG
|   |           |   |   +----G_SELSUG.PRG
|   |           |   +----G_SUGOFF.PRG
|   |           +----G_DILFAC.PRG
|   |           |   +----FAKE_LEFT.PRG
|   |           |   +----CHANGESTRAIN (procedure in G_DILFAC.PRG)
|   |           |   |   +----STRAINLIST.PRG
|   |           |   |   +----CHANGESTRAIN (procedure in G_DILFAC.PRG)
*** Truncating tree for recursive call to CHANGESTRAIN
|   |           |   |   +----EDIT_SUG.PRG
|   |           |   |   +----EXPAND_SUG.PRG
|   |           |   |   +----SET_INST.PRG
|   |           |   |   +----G_PROTO.PRG
|   |           |   |       +----G_PROTO.PRG
*** Truncating tree for recursive call to G_PROTO.PRG
|   |           |   +----G_DFSCRN.PRG
|   |           |   +----STRAINLIST.PRG
|   |           |   +----MOD_SAVE.PRG
|   |           +----G_NPMUT.PRG
|   |           +----G_NPSRV.PRG
|   |           +----G_ACTSYS.PRG
|   |           |   +----ENTER_COMM.PRG
|   |           |   +----G_ACINIT.PRG
|   |           |   +----G_ACSCRN.PRG
|   |           |   +----G_SUGON.PRG
|   |           |   |   +----EDIT_SUG.PRG
```

```
|  |        |  |  +---EXPAND_SUG.PRG
|  |        |  |  +---UP_BOX.PRG
|  |        |  |  +---DOWN_BOX.PRG
|  |        |  +---G_SELSUG.PRG
|  |        |  +---MOD_SAVE.PRG
|  |        +---G_SMPSPC.PRG
|  |        |  +---G_SMSCRN.PRG
|  |        |  +---G_SUGON.PRG
|  |        |  |  +---EDIT_SUG.PRG
|  |        |  |  +---EXPAND_SUG.PRG
|  |        |  |  +---UP_BOX.PRG
|  |        |  |  +---DOWN_BOX.PRG
|  |        |  +---G_SELSUG.PRG
|  |        |  +---ENTER_COMM.PRG
|  |        |  +---MOD_SAVE.PRG
|  |        |  +---G_SUGOFF.PRG
|  |        +---G_CURSTR.PRG
|  |        |  +---G_CUSCRN.PRG
|  |        |  +---STRAINLIST.PRG
|  |        |  +---G_SUGON.PRG
|  |        |  |  +---EDIT_SUG.PRG
|  |        |  |  +---EXPAND_SUG.PRG
|  |        |  |  +---UP_BOX.PRG
|  |        |  |  +---DOWN_BOX.PRG
|  |        |  +---G_SELSUG.PRG
|  |        |  +---CUR_STRAIN.PRG
|  |        |  +---MOD_SAVE.PRG
|  |        |  +---G_SUGOFF.PRG
|  |        +---G_QUALTY.PRG
|  |        |  +---G_QUSCRN.PRG
|  |        |  +---G_SUGON.PRG
|  |        |  |  +---EDIT_SUG.PRG
|  |        |  |  +---EXPAND_SUG.PRG
|  |        |  |  +---UP_BOX.PRG
|  |        |  |  +---DOWN_BOX.PRG
|  |        |  +---G_SELSUG.PRG
|  |        |  +---MOD_SAVE.PRG
|  |        +---G_MUTDOSE.PRG
|  |        +---G_SRVDOSE.PRG
|  |        +---G_REMARK.PRG
|  |        |  +---G_SUGON.PRG
|  |        |  |  +---EDIT_SUG.PRG
|  |        |  |  +---EXPAND_SUG.PRG
|  |        |  |  +---UP_BOX.PRG
|  |        |  |  +---DOWN_BOX.PRG
|  |        |  +---G_SELSUG.PRG
|  |        |  +---G_RESCRN.PRG
|  |        |  +---APNDINST (procedure in G_REMARK.PRG)
|  |        |  |  +---G_INSTFV.PRG
|  |        |  |  +---G_APPEND (procedure in G_REMARK.PRG)
|  |        |  |  +---G_INVIEW.PRG
|  |        |  +---MOD_SAVE.PRG
|  |        +---G_CONTNU.PRG
|  |           +---G_SAVEAL.PRG
|  |           |  +---G_SAVEXP.PRG
|  |           |  +---G_SAVSMP.PRG
|  |           |  +---G_SAVGEN.PRG
|  |           |  +---G_SAVNP.PRG
|  |           |  |  +---G_QUITAL.PRG
```

```
|  |          |  +----G_SAVSMD.PRG
|  |          |  |   +----G_QUITAL.PRG
|  |          |  +----G_SAVTST.PRG
|  |          |  |   +----G_QUITAL.PRG
|  |          |  +----G_SELSUG.PRG
|  |          +----G_SAVNEW.PRG
|  |          |  +----G_SAVSMP.PRG
|  |          |  +----G_SAVSMD.PRG
|  |          |  |   +----G_QUITAL.PRG
|  |          |  +----G_SAVTST.PRG
|  |          |      +----G_QUITAL.PRG
|  |          +----G_VAXFLE.PRG
|  |             +----G_CHKCNT.PRG
|  |             +----G_GENREC.PRG
|  |             |   +----G_RPPAGE.PRG
|  |             +----G_GENNPS.PRG
|  |                +----G_CHKTPN.PRG
|  |                +----G_GENREC.PRG
|  |                |   +----G_RPPAGE.PRG
|  |                +----G_GENTST.PRG
|  +----G_MODSMP.PRG
|  |  +----G_SCNTRL.PRG
|  |     +----G_EXTRNL.PRG
|  |        +----G_EXPMNT.PRG
|  |        |  +----G_EXINIT.PRG
|  |        |  +----G_EXSCRN.PRG
|  |        |  +----G_SUGON.PRG
|  |        |  |  +----EDIT_SUG.PRG
|  |        |  |  +----EXPAND_SUG.PRG
|  |        |  |  +----UP_BOX.PRG
|  |        |  |  +----DOWN_BOX.PRG
|  |        |  +----G_SELSUG.PRG
|  |        |  +----MOD_SAVE.PRG
|  |        |  +----G_SUGOFF.PRG
|  |        +----G_ASSAY.PRG
|  |        |  +----G_ASSCRN.PRG
|  |        |  +----G_SUGON.PRG
|  |        |  |  +----EDIT_SUG.PRG
|  |        |  |  +----EXPAND_SUG.PRG
|  |        |  |  +----UP_BOX.PRG
|  |        |  |  +----DOWN_BOX.PRG
|  |        |  +----G_SELSUG.PRG
|  |        |  +----MOD_SAVE.PRG
|  |        +----G_STRAIN.PRG
|  |        |  +----FAKE_LEFT.PRG
|  |        |  +----G_STSCRN.PRG
|  |        |  +----G_SUGON.PRG
|  |        |  |  +----EDIT_SUG.PRG
|  |        |  |  +----EXPAND_SUG.PRG
|  |        |  |  +----UP_BOX.PRG
|  |        |  |  +----DOWN_BOX.PRG
|  |        |  +----G_SELSUG.PRG
|  |        |  +----MOD_SAVE.PRG
|  |        |  +----STRN_MOVE.PRG
|  |        |  +----G_MSVSTN.PRG
|  |        |  |  +----G_SELSUG.PRG
|  |        |  +----G_SUGOFF.PRG
|  |        +----G_DILFAC.PRG
|  |        |  +----FAKE_LEFT.PRG
```

```
|   |           |   +---CHANGESTRAIN (procedure in G_DILFAC.PRG)
|   |           |   |   +---STRAINLIST.PRG
|   |           |   |   +---CHANGESTRAIN (procedure in G_DILFAC.PRG)
*** Truncating tree for recursive call to CHANGESTRAIN
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---SET_INST.PRG
|   |           |   |   +---G_PROTO.PRG
|   |           |   |       +---G_PROTO.PRG
*** Truncating tree for recursive call to G_PROTO.PRG
|   |           |   +---G_DFSCRN.PRG
|   |           |   +---STRAINLIST.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           +---G_NPMUT.PRG
|   |           +---G_NPSRV.PRG
|   |           +---G_ACTSYS.PRG
|   |           |   +---ENTER_COMM.PRG
|   |           |   +---G_ACINIT.PRG
|   |           |   +---G_ACSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           +---G_SMPSPC.PRG
|   |           |   +---G_SMSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---ENTER_COMM.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           |   +---G_SUGOFF.PRG
|   |           +---G_CURSTR.PRG
|   |           |   +---G_CUSCRN.PRG
|   |           |   +---STRAINLIST.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---CUR_STRAIN.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           |   +---G_SUGOFF.PRG
|   |           +---G_QUALTY.PRG
|   |           |   +---G_QUSCRN.PRG
|   |           |   +---G_SUGON.PRG
|   |           |   |   +---EDIT_SUG.PRG
|   |           |   |   +---EXPAND_SUG.PRG
|   |           |   |   +---UP_BOX.PRG
|   |           |   |   +---DOWN_BOX.PRG
|   |           |   +---G_SELSUG.PRG
|   |           |   +---MOD_SAVE.PRG
|   |           +---G_MUTDOSE.PRG
```

```
|   |         +----G_SRVDOSE.PRG
|   |         +----G_REMARK.PRG
|   |         |   +----G_SUGON.PRG
|   |         |   |   +----EDIT_SUG.PRG
|   |         |   |   +----EXPAND_SUG.PRG
|   |         |   |   +----UP_BOX.PRG
|   |         |   |   +----DOWN_BOX.PRG
|   |         |   +----G_SELSUG.PRG
|   |         |   +----G_RESCRN.PRG
|   |         |   +----APNDINST (procedure in G_REMARK.PRG)
|   |         |   |   +----G_INSTFV.PRG
|   |         |   |   +----G_APPEND (procedure in G_REMARK.PRG)
|   |         |   |   +----G_INVIEW.PRG
|   |         |   +----MOD_SAVE.PRG
|   |         +----G_CONTNU.PRG
|   |           +----G_SAVEAL.PRG
|   |           |   +----G_SAVEXP.PRG
|   |           |   +----G_SAVSMP.PRG
|   |           |   +----G_SAVGEN.PRG
|   |           |   +----G_SAVNP.PRG
|   |           |   |   +----G_QUITAL.PRG
|   |           |   +----G_SAVSMD.PRG
|   |           |   |   +----G_QUITAL.PRG
|   |           |   +----G_SAVTST.PRG
|   |           |   |   +----G_QUITAL.PRG
|   |           |   +----G_SELSUG.PRG
|   |           +----G_SAVNEW.PRG
|   |           |   +----G_SAVSMP.PRG
|   |           |   +----G_SAVSMD.PRG
|   |           |   |   +----G_QUITAL.PRG
|   |           |   +----G_SAVTST.PRG
|   |           |       +----G_QUITAL.PRG
|   |           +----G_VAXFLE.PRG
|   |               +----G_CHKCNT.PRG
|   |               +----G_GENREC.PRG
|   |               |   +----G_RPPAGE.PRG
|   |               +----G_GENNPS.PRG
|   |                   +----G_CHKTPN.PRG
|   |                   +----G_GENREC.PRG
|   |                   |   +----G_RPPAGE.PRG
|   |                   +----G_GENTST.PRG
|   +----G_INSTRC.PRG
|       +----G_INADD.PRG
|       |   +----G_INASCR.PRG
|       +----G_INVIEW.PRG
|       +----G_INSTFV.PRG
+----G_UTIL.PRG
|   +----G_LISTEX.PRG
|   +----G_DBRIND.PRG
|   +----G_EXPLOD.PRG
|   |   +----DP_SCRN (procedure in G_EXPORT.PRG)
|   |   +----RMVEXP.PRG
|   +----G_IMPEXP.PRG
|   |   +----G_IMPORT.PRG
|   |   |   +----EXIT_PROC (procedure in G_IMPORT.PRG)
|   |   |   +----RMVEXP.PRG
|   |   +----G_EXPORT.PRG
|   |       +----DP_SCRN (procedure in G_EXPORT.PRG)
|   |       +----G_VAXFLE.PRG
```

```
|   |         +---G_CHKCNT.PRG
|   |         +---G_GENREC.PRG
|   |         |   +---G_RPPAGE.PRG
|   |         +---G_GENNPS.PRG
|   |             +---G_CHKTPN.PRG
|   |             +---G_GENREC.PRG
|   |             |   +---G_RPPAGE.PRG
|   |             +---G_GENTST.PRG
|   +---G_DELRV.PRG
|   |   +---RMVEXP.PRG
|   +---G_ARCMNU.PRG
|       +---G_ARCHIV.PRG
|       |   +---G_VAXFLE.PRG
|       |   |   +---G_CHKCNT.PRG
|       |   |   +---G_GENREC.PRG
|       |   |   |   +---G_RPPAGE.PRG
|       |   |   +---G_GENNPS.PRG
|       |   |       +---G_CHKTPN.PRG
|       |   |       +---G_GENREC.PRG
|       |   |       |   +---G_RPPAGE.PRG
|       |   |       +---G_GENTST.PRG
|       |   +---ARCH_SCRN (procedure in G_ARCHIV.PRG)
|       |   +---RMVEXP.PRG
|       +---G_DEARCH.PRG
|       |   +---ARCH_SCRN2 (procedure in G_DEARCH.PRG)
|       |   +---DP_SCRN2 (procedure in G_DEARCH.PRG)
|       |   +---EXIT_PROC (procedure in G_IMPORT.PRG)
|       |   +---RMVEXP.PRG
|       |   +---G_RESET (procedure in G_ACCEPT.PRG)
|       +---G_ACCEPT.PRG
|           +---EXIT_PROC (procedure in G_IMPORT.PRG)
|           +---RMVEXP.PRG
|           +---G_RESET (procedure in G_ACCEPT.PRG)
+---G_REPORT.PRG
|   +---G_RPTCHS.PRG
|       +---G_RAWDAT.PRG
|       |   +---G_CLRSEP.PRG
|       |   +---G_RPRAWH.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   +---G_RPRWHM.PRG
|       |   +---G_RPRAWD.PRG
|       |   +---G_RPRWFM.PRG
|       |   +---G_RPRWTM.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   +---G_RPRWOM.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   +---G_RPRWHS.PRG
|       |   +---G_RPRWFS.PRG
|       |   +---G_RPRWTS.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   +---G_RPRWOS.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   +---G_RPRAWG.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   +---G_RPRAWR.PRG
|       |   |   +---G_RPPAGE.PRG
|       |   |   +---G_GENREC.PRG
|       |   |       +---G_RPPAGE.PRG
|       |   +---G_RPRAWS.PRG
```

```
|       |   |   +----G_RPRASH.PRG
|       |   |   |   +----G_RPPAGE.PRG
|       |   |   +----G_GENREC.PRG
|       |   |       +----G_RPPAGE.PRG
|       |   +----G_RPRAW9.PRG
|       |       +----G_RPRASH.PRG
|       |       |   +----G_RPPAGE.PRG
|       |       +----G_GENREC.PRG
|       |           +----G_RPPAGE.PRG
|       +----G_STATS.PRG
|       |   +----DSRCH.PRG
|       |   +----BBRIDGE.PRG
|       |   |   +----G_RPTOUT.PRG
|       |   +----HBRIDGE.PRG
|       |   |   +----G_RPTOUT.PRG
|       |   +----SBRIDGE.PRG
|       |   |   +----G_RPTOUT.PRG
|       |   +----G_ALLRPT.PRG
|       |       +----G_RPTOUT.PRG
|       +----CSRCH.PRG
+----G_MDEFLT.PRG
|   +----G_COMMUN.PRG
|   +----G_COLORS.PRG
|   +----G_INTCLR.PRG
|   +----G_SPEOPT.PRG
|       +----G_QUITAL.PRG
|       +----G_NEWPAS.PRG
|       +----G_SPMENU.PRG
|           +----G_NEWPAS.PRG
|           +----G_SUGGES.PRG
|           |   +----G_SELSUG.PRG
|           |   +----G_SUSEL.PRG
|           |   +----G_SUEXP.PRG
|           |   |   +----G_SUMOD.PRG
|           |   |       +----G_SULIST.PRG
|           |   |       |   +----G_SULMOD.PRG
|           |   |       +----G_SUHELP.PRG
|           |   |       +----G_SULMOD.PRG
|           |   |       +----G_SUHMOD.PRG
|           |   |           +----G_SUHELP.PRG
|           |   +----G_SUASSY.PRG
|           |   |   +----G_SUMOD.PRG
|           |   |       +----G_SULIST.PRG
|           |   |       |   +----G_SULMOD.PRG
|           |   |       +----G_SUHELP.PRG
|           |   |       +----G_SULMOD.PRG
|           |   |       +----G_SUHMOD.PRG
|           |   |           +----G_SUHELP.PRG
|           |   +----G_SUFINL.PRG
|           |   |   +----G_SUMOD.PRG
|           |   |       +----G_SULIST.PRG
|           |   |       |   +----G_SULMOD.PRG
|           |   |       +----G_SUHELP.PRG
|           |   |       +----G_SULMOD.PRG
|           |   |       +----G_SUHMOD.PRG
|           |   |           +----G_SUHELP.PRG
|           |   +----G_SUNP.PRG
|           |   |   +----G_SUMOD.PRG
|           |   |       +----G_SULIST.PRG
```

```
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SUACT.PRG
|       |  |   +---G_SUMOD.PRG
|       |  |       +---G_SULIST.PRG
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SUSAMP.PRG
|       |  |   +---G_SUMOD.PRG
|       |  |       +---G_SULIST.PRG
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SUCUR.PRG
|       |  |   +---G_SUMOD.PRG
|       |  |       +---G_SULIST.PRG
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SUQUAL.PRG
|       |  |   +---G_SUMOD.PRG
|       |  |       +---G_SULIST.PRG
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SUDOSE.PRG
|       |  |   +---G_SUMOD.PRG
|       |  |       +---G_SULIST.PRG
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SURMRK.PRG
|       |  |   +---G_SUMOD.PRG
|       |  |       +---G_SULIST.PRG
|       |  |       |   +---G_SULMOD.PRG
|       |  |       +---G_SUHELP.PRG
|       |  |       +---G_SULMOD.PRG
|       |  |       +---G_SUHMOD.PRG
|       |  |           +---G_SUHELP.PRG
|       |  +---G_SUREP.PRG
|       |  +---G_SUREXT.PRG
|       |      +---G_SUMOD.PRG
|       |          +---G_SULIST.PRG
|       |          |   +---G_SULMOD.PRG
|       |          +---G_SUHELP.PRG
```

```
|          |          +----G_SULMOD.PRG
|          |          +----G_SUHMOD.PRG
|          |             +----G_SUHELP.PRG
|          +----G_PIMENU.PRG
|          |   +----G_PIADD.PRG
|          |   |   +----PI_SAVE (procedure in G_PIADD.PRG)
|          |   +----G_PIMOD.PRG
|          |       +----G_PIADD.PRG
|          |          +----PI_SAVE (procedure in G_PIADD.PRG)
|          +----G_DBZAP.PRG
|          +----G_AMASSY.PRG
|          |   +----G_AMASSC.PRG
|          |   +----G_SELSUG.PRG
|          |   +----G_SUGOFF.PRG
|          +----G_PLTITR.PRG
|          +----HELP.PRG
|          |   +----HELP.PRG
*** Truncating tree for recursive call to HELP.PRG
|          +----G_LSTARC.PRG
|          +----G_DELEXP.PRG
|             +----RMVEXP.PRG
+----G_MQUIT.PRG
+----G_QUITAL.PRG
```


☆U.S. GOVERNMENT PRINTING OFFICE: 1994 - 550-001/80379