United States General Accounting Office

GAO

Information Management and Technology Division

April 1991

Preparing, Documenting, and Referencing Microcomputer Data Base Applications

GAO/IMTEC-11.1.1

Preface

GAO is increasingly using microcomputer data base software for data collection, validation, analysis, and reporting. As with other automated techniques, integrating data base applications into GAO assignments introduces new methods of meeting existing audit and evaluation standards. What are appropriate uses of data base software? How should its use be documented? What, if any, special referencing concerns are raised?

The purpose of this guide, which has been written for those with basic knowledge of data base software concepts, is to help evaluators and supervisors apply and review data base applications on the microcomputer by discussing

- an overview of data base concepts (chapter 1);
- how to choose and plan a data base application (chapter 2);
- data validation techniques, processing controls, and general workpaper requirements (chapter 3);
- how to collect, analyze, and document data using data base software programming modes (chapter 4); and
- technical review and referencing requirements (chapter 5).

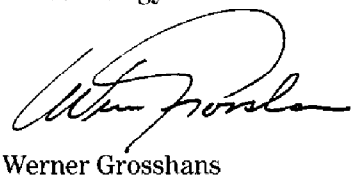
Following these guidelines should improve quality assurance and facilitate the review and referencing processes.

A GAO evaluator with data base software training should be able to reference data base applications. It is important to note, however, that referencers are not required to be familiar with data base software. **GAO policy continues to require that if referencers encounter highly technical material or an unfamiliar methodology (econometric modeling, data base application, etc.), they are to obtain assistance. Referencers are responsible for verifying that a qualified individual from a technical assistance or economic analysis group, or other qualified person independent of**

the assignment, has examined and approved the manner in which data have been developed and used. These examinations should be documented and made part of the workpapers, and such documentation should be acceptable to the referencer.

This document was prepared by a team of headquarters and regional staff. It expands on the guidance provided in GAO's policy manuals and will be incorporated into the automated policy guidance system. Major contributors are shown in appendix VIII. The Information Management and Technology Division (IMTEC) provided project leadership. IMTEC technical staff and Office of Policy staff were involved in review and preparation of the final document.

Ralph V. Carlone
Assistant Comptroller General
Information Management and
Technology Division



Werner Grosshans
Assistant Comptroller
General for Policy

Contents

Appendixes

Bibliography

Abbreviations

EIC	evaluator-in-charge
GAO	General Accounting Office
IMTEC	Information Management and Technology Division
TAG	technical assistance group

Introduction

This document provides guidelines for the use of microcomputer data base software, and for the planning, documenting, testing, and referencing of data base applications. The examples and illustrations used throughout this document refer to dBASE[1] because of its widespread use in GAO. However, methods discussed usually have counterparts in other data base software. This chapter presents

- an overview of data base software,
- when to use a data base management program, and
- definition of terms.

Overview of Data Base Software

Data base software allows staff to create, use, modify, and access information from multiple data bases. It enables staff to

- collect data through automated data entry forms and screen entries for accuracy;
- import existing agency files;
- store and analyze large amounts of data;
- produce custom reports quickly and easily;
- store commands in programs for repeated execution;
- produce stand-alone systems for data collection and analysis; and
- speed execution and save disk space (as compared to spreadsheet processing).

In using data base software, staff should take special care to provide an audit trail. Data are not continuously visible to the user, and the software does not automatically provide a record of data changes and computations. dBASE has two modes of operation—interactive or programming—each requiring different steps for creating an audit trail. The interactive mode allows a user to issue individual, ad hoc commands intermittently to the software. In the

[1] dBASE and dBASE III Plus are registered trademarks of Ashton-Tate.

programming mode, all commands are stored in a file and executed at one time when the program is run.

When to Use a Data Base Management Program

Use data base software's advanced analysis and reporting capabilities when the data volume exceeds what a spreadsheet can efficiently handle. The speed and ease with which data can be used depends on record length and type and capacity of hardware.

Data base software also allows flexibility in data input and retrieval. The software can be used to create data entry screens. For instance, if several hundred records need to be entered, it pays to take the time to write programs setting up data collection screens. These screens can include explanatory text, and programs may contain edit checks to increase data accuracy. (Chapter 3 gives additional information.) If data are collected by different people at various sites, these screens increase data entry consistency. Once the data base is complete, the software provides easy and flexible query and reporting capabilities.

Data base software has many built-in analysis features, providing selection and arithmetic capabilities. More complex analyses can be programmed, as well as a complete system for both data retrieval and analysis.

Terms Defined

Definitions of terms used in this guide are as follows:

Compile: The act of translating a language source program into an object module or program written in machine language. Programs written with data base software must be run while the software is in use. A compiler converts such programs into machine executable code so the data base software is no longer required.

Data base index file: Contains pointers to physical locations of data base records in a particular sorted order. Used in conjunction with the data base, the index "sorts" the data for the user without creating an additional sorted data base, thus saving disk space.

Data dictionary: Explains in detail the source, format, and meaning of the contents of a data base.

Disk directory: A catalog of all files contained on a disk. This includes the file name, size, and date of creation of the file.

Field: A category of information, such as names, cities, birth dates, or salaries.

Flow chart: A diagram showing a process through predefined symbols interconnected to indicate sequential individual steps. Explanatory text within the symbols clarifies the steps.

Record: A collection of related field entries. For example, the name, address, city, state, zip code, and telephone number for an individual would form a record.

Planning

In planning assignments, audit quality and productivity can be improved through appropriate computer technology. Appendix IX to Chapter 10.1 of the Project Manual describes various software packages currently available to GAO staff. Another pertinent document is Information Management and Technology (IMTEC) Division's Technical Guideline titled, Preparing, Documenting, and Referencing Lotus Spreadsheets. dBASE may be one of several software packages used during an assignment. For example, one may choose dBASE to collect and store data, and SAS[2] (a statistical analysis package) for data analysis. This chapter provides guidance for planning the use of data base software, including

- identifying data sources,
- developing a validation plan,
- formulating a data analysis plan,
- data base design considerations,
- creating data bases,
- planning for referencing and technical reviews, and
- defining roles and responsibilities for designing and using data bases.

Identifying Data Sources

Agencies often have established data bases that fulfill audit objectives. Use such data bases whenever possible; verifying their validity is usually cheaper than developing and maintaining an independent data base. If necessary, a dBASE file can be built to accommodate data from other microcomputer software or mainframes.

When external data sources are not available and the staff must collect data, time should be planned to design the data base, data dictionary, and any programmed collection screens. (See app. I for an example of a data base structure and app. II for a sample data dictionary.) Time must also be factored

[2]SAS and SAS-PC are registered trademarks of the SAS Institute Inc.

in for testing data validity and reliability **regardless of its source.**

Validation Plan

The analysis plan should provide for validation of data. (See chapter 3 for validation techniques for microcomputer data bases.) Regardless of the source of the data (agency or GAO), a reliability assessment should also be performed. (See the GAO audit guide, Assessing the Reliability of Computer-Processed Data (GAO/OP-8.1.3.)

Data Analysis Plan

To guide and focus the assignment and strengthen conclusions, the Project Manual (6.2) encourages the development of a detailed data analysis plan explaining how information will be gathered and used.

A process flow chart can supplement a written data analysis plan. By showing what will be done at various points in the process, a flow chart clarifies the analysis plan for the benefit of other staff members. See appendix III for a sample flow chart.

Both a written plan and a flow chart should explain planned data bases, indexes, and their interaction. They should also explain what data analysis steps will be taken and when, as well as the desired reports. This plan should also be included in the workpapers.

The following questions should be used to design the analysis plan:

- What analyses must be performed?
- Will the data base software perform the planned analysis effectively? Perhaps a statistical package such as SAS-PC or SPSS-PC+[3]would be more effective than dBASE in performing part or all of the

[3]SPSS-PC+ is a trademark of SPSS Inc.

analysis. To determine the most appropriate analytical software, staff should consult with their unit's Design, Methodology, and Technical Assistance Group or Technical Assistance Group (TAG).

- What mode of data base processing, interactive or programming, best suits the need? (See below for additional information on this decision.)
- What type of data validation will be done and when?
- How will data analysis steps be tested for accuracy? Who will do it?

Data Base Design Considerations

When designing a data base, one should consider the data, analysis, and planned reports. Specifically:

- **What kind of data will be used?** What are maximum values of numbers? How should character data (e.g., names) be stored, and what length is necessary? Will the data fall within known ranges (e.g., 1 through 5)? If using agency data bases, be sure to get data base structures, data dictionaries, and information on how the data were collected and checked.
- **How will the data be analyzed and sorted?** What calculations will be done? Will there be a need to move data between data bases, or to access data from multiple data bases?
- **What will be produced?** What type of reports, mailing labels, etc., are needed? Will the data base be used with another software package?

Creating Data Bases

When creating a data base both the data base file and fields it contains must be named. Beginning related files with the same prefix can be helpful. For example, a data base containing Environmental Protection Agency data for the state of Ohio could be named EPA-OHIO.DBF and its index EPA-OHIO.NDX. Choose field names that are descriptive of the data they represent: LASTNAME, ZIPCODE, LOSSAMT, etc.

Data base software provides for a variety of field types. For example, dBASE gives a choice of five field types: character, numeric, date, memo, and logical. It is best to use character fields for everything except dates and numbers involved in calculations. Both memo and logical fields are slow and have no advantages over character fields. In fact, memo fields are difficult to edit and impossible to search; logical fields have restricted use and require special logic to analyze.

Make fields large enough to hold the data without wasting space. Make numeric fields large enough for maximum values and for totaling the field. Remember to include decimal points and numeric signs in the field size.

These actions will also help:

- Break up character fields into the smallest possible units. For example, instead of a field called NAME, use LASTNAME and FIRSTNAME. This allows you to search or sort either field.
- Let one field uniquely identify each record to help eliminate duplicates (e.g., Social Security number).
- Reduce errors and save time by not entering numbers that can be calculated.

Interactive vs. Programming Mode

Decide during planning how the data base software will be used. In a typical assignment, it would be common to use both the interactive and programming mode. In the interactive mode, the user issues individual commands from the dBASE "dot" prompt[4] or from a command ("Assist") menu. In the programming mode, all commands are stored in a file and performed in a continuous sequence when the program is executed.

The interactive mode is appropriate when

[4]dBASE uses a period to prompt the user for input at the command line, hence the term "dot" prompt.

- ad hoc queries are necessary,
- deadlines preclude preparing data collection screens,
- data volume is small, or
- users have a thorough understanding of the data base software.

The programming mode is appropriate when

- a stand-alone data collection and analysis system is desired (programs can be compiled, thereby eliminating the need for copies of the data base software);
- users have limited technical knowledge of the data base software, and technical assistants can prepare data collection or analysis systems;
- data will be inputed at multiple sites or by several staff, thereby raising concerns about data consistency;
- programmed data entry checks are desired; or
- the data collection and analysis will be repeated.

Programs should be used for complex analysis that cannot be performed interactively. For example, a program would be used to combine data stored by month into another field designed to hold quarterly data. Another benefit is that a program is itself a record of how the data was analyzed, thus simplifying documentation.

Referencing and Technical Reviews

Planning should include a decision about when and how referencing and technical reviews will be done. (See Chapter 5 for additional information on referencing.) The referencer may request the assistance of an independent, qualified technical reviewer to certify the methodology and accuracy of computer-based applications. Regardless of whether a technical review will be done for the referencer, reviews of computer applications should be done on a timely basis throughout the assignment. Early review is particularly important if findings and conclusions

depend on the data analysis. Thus, any problems can be identified in time to make adjustments.

Defining Roles and Responsibilities

Expectations concerning roles and responsibilities of the evaluators, technical assistance staff, and support staff should be clear. Clearly defined roles and responsibilities help ensure consistency in data entry and analysis, and in keeping track of data base changes. These roles and responsibilities should be fully described in the audit program. In addition, review and approval responsibilities should be built into the expectation statements.

The following responsibilities should be clearly delineated to designated staff:

- developing the data base design and defining the field contents to maintain consistency,
- making and documenting changes to the original data and storing the original data base,
- securing and maintaining the integrity of the data base and programs,
- backing up the data at predefined intervals,
- defining analysis and reporting requirements, and
- reviewing and approving responsibilities.

Data Validation, Documentation, and Testing of Data Analysis

GAO products rely on quality evidence and sound analysis to support the findings, conclusions, and recommendations. Therefore, data validation, documentation, and testing of data analysis are extremely important in ensuring quality products.

This chapter discusses

- validating original data,
- validating data transferred between computers and software packages,
- documenting interactive work sessions, and
- documenting and testing data analysis procedures.

Procedures discussed in this chapter apply to both dBASE interactive and programming modes.

Data Validation

Government Auditing Standards requires checking the validity of original data. In addition, it is good practice to establish checks to ensure the integrity of data when it is transferred between computers or different software packages. These checks generally include reasonableness checks, record counts, data totals, hash totals, and data verification.

- A **reasonableness check** is a common sense look at the data. For instance:
  - Are amounts too small (cost per mile to operate a 1-ton truck = $.004)?
  - Are amounts too large (accrued annual leave balance for a non-SES employee = 3,000 hours; a student loan for $1,500,000)?
  - Are the data fields complete (data field for patient name is blank; Social Security data field has only eight numbers)?
  - Does the data seem logical (a child is older than her mother)?
- **Record counts** should match the original number of computerized records.

- **Data totals** are the sum of numeric data, such as dollars or quantities. These should be the same in the original and newly transferred or created data bases.
- **Hash totals** are the sum of numbers not normally added, such as Social Security numbers or grade levels. As with data totals, they should be the same for both files.
- **Data verification** compares the contents of the data base with its source.

Data Transfer

Data bases are often transferred from

- one software package to another,
- one microcomputer to another, or
- a mainframe computer to a microcomputer.

In this process it is important that procedures be developed to ensure that data transfer will be successful.

In whatever way data are transferred, data accuracy should be verified before any data analysis is performed. The extent of review depends on how critical the data are and the risk data transfer entails. Test the data received using the validation techniques given in the previous section.

Minimum documentation of data transfers should include

- printed copies of the disk directories of both the original and the transferred files;
- an explanation of how the transfer was done, including names of software used; and
- a description of tests used to verify data accuracy and completeness.

Documenting Interactive Work Sessions

The interactive mode is often used to analyze agency data bases, to perform ad hoc queries, and to run simple analyses. To use the interactive mode, one must know dBASE and keep a clear audit trail.

When using the interactive mode, enter commands from the dot prompt, not through dBASE command ("Assist") menus. Procedures accomplished through such menus cannot be adequately documented.

Using Alternate Files

dBASE "alternate" files create an audit trail of interactive sessions. They are text files that record most commands and responses. This file can later be edited and printed, or kept in an electronic file. Appendix IV is an example of an alternate file.

Here are tips on how to set up and use alternate files.

- Type these commands at the dot prompt:

  SET ALTERNATE TO logfile

  SET ALTERNATE ON

  This creates a text file called "logfile," that will capture most commands and their results. These files can be named by date (e.g., "LOG1022A" for the text file created during the first work session on October 22).

- Then, display the current date and time to record this information in the alternate file. The dBASE commands are

  ? DATE( )

  ? TIME( )

- The alternate file will not capture results of commands like CREATE, APPEND, and EDIT. This can be compensated for by doing appropriate checks,

such as displaying the data base structure before and after each procedure. When necessary, comments can be added to the alternate file after the session is complete.

- When opening a data base, type **"DISPLAY STATUS"** and **"DISPLAY STRUCTURE"** to show information about the data base.

- Alternate files will not capture reports created by a report format file. To document these, print the report format screens or annotate the report.

- To close the alternate file, type **"SET ALTERNATE OFF"** before using the **"QUIT"** command. Because the results of each session can be quite lengthy, an edit of the file using a word processor may be useful. Delete mistyped or erroneous commands. Keep commands and results that create or change data or data bases, or are part of the analysis. After editing, the alternate files can be printed or stored on diskettes and included in the workpapers.

  Since alternate files can be edited with a word processor, they are not a perfect audit trail; they do, however, illustrate the steps taken. If an independent person could repeat the steps and achieve the same results, the audit trail is sufficient.

Documentation and Testing of Data Analysis

As part of documentation, include logs of interactive work sessions (alternate files), any programs, and results.

Perform control checks after each step of the analysis to ensure that no data have been lost or altered and the analysis went as planned. Such control checks include

- comparing present and previous record counts;
- totaling numeric fields before and after each step;
- checking for duplicate or blank records; and

- hand-calculating computations for a small, diverse group of records, and comparing the results with computer-generated totals.

Workpapers should provide a clear trail of all work done and include

- an analysis plan;
- all interactive session logs, with results (such as reports, data listings, etc.) in chronological sequence (app. IV is an example);
- a written description or flow chart of the process, cross-referenced to programs and session logs (see app. V);
- hierarchy flow charts of any programmed systems (see app. VI);
- all programs (see app. VII) and their results;
- printed copies of data entry screens;
- a summary of any problems found in the data and their disposition;
- a summary of work performed; and
- evidence of timely supervisory review.

Additional Considerations

Here are additional practices to keep in mind:

- Provisions of the Computer Security Act of 1987 should be followed as they apply to the processing and storing of sensitive and proprietary data. See Project Manual (Chapter 11.1-9).
- The supervisor must review all data collection procedures and workpapers, and ensure that the data were verified against the source.
- **Staff should back up data, programs, and other related files** (e.g., indexes, report files) **periodically.**
- Use the dBASE command **"QUIT"** to exit the program. **Do not remove the data diskette before exiting the system because you risk losing the data.**
- Close data bases as soon as use of them is finished, ("CLOSE DATABASES"). dBASE files are vulnerable to damage while the data base is open.

- dBASE calculations treat blank numeric fields as zeroes. Plan for a way to indicate missing data to distinguish it from fields left blank in error, (e.g., entering "999999" for missing data).

Programming With Data Base Software

Data base programs allow the user to do complex or repetitive tasks quickly and accurately. These programs can do one basic task or automate an entire job. This chapter offers guidance for

- using programmed data entry screens,
- writing programs,
- using data base software with complex programs, and
- testing programs.

Using Programmed Data Entry Screens

Data base software's programming capabilities allow the user to design screens for data entry and editing. Such screens simplify and clarify data entry, while providing the capability to limit input errors. They provide the user with more control than interactive data entry allows. For example, when adding or editing records in the interactive mode, dBASE will prompt for input using field names. Being limited to 8 to 10 characters can make these field names unrecognizable. Here is an example of a dBASE interactive input screen.

```
FACILITY    [     ]
SAMPLENUM   [    ]
REGSTRNUM   [         ]
SSN_FMP     [             ]
PATCATGRY   [ ]
TYPETRTMNT  [ ]
CHARGES     [           .   ]
TRTMNTDATE  [  /  /   ]
INJCAUSE    [      ]
DISCHARGE   [  /  /   ]
```

The user enters data to the right of the field name (illustrated here with brackets).

In contrast, programmed entry screens can be easier to understand, and allow data checking to increase consistency and accuracy. Here is a sample data collection screen.

```
          MEDICAL COST RECOVERY -- JOB CODE 123456
        INPUT SCREEN FOR MEDICAL FACILITY PORTION
-------------------------------------------------------------

 Facility:  WHMC    Sample Number:   3   Type of treatment:  2

 Register number:   12345678      SSN/FMP:         11122333344

 Patient category:  4             Charges:            80000

 Admission date:    08/16/89      Discharge date:  08/20/89

-------------------------------------------------------------
```

Possible data checks that can be programmed are shown in the following examples:

- Limit the values of numbers. For example, the "Patient category" above may be limited to values greater than 0 and less than 6. This can be done with either the dBASE "range" command or the Clipper[5] "valid" command.
- Limit the responses of character fields, such as "T" and "F" for a true/false question.
- Compare the answers of related questions. For example, a child's age cannot be greater than the mother's.

When practical, use memory variables for data entry, run error/logic checks on them, and then

[5]Clipper is a trademark of Nantucket Corporation. It can be used to compile dBASE III Plus programs, but also has its own programming commands.

replace the appropriate field with the memory variable contents.

Writing Programs

Rather than writing the entire program at once, it is easier to write and test individual tasks separately. Liberal use of blank lines and indentations will also make programs easier to read. Before writing a program, answer these questions.

- What is its purpose? Divide a large program into its logical units and state the purpose of each.
- What data will be used?
- What analysis will be done?
- What reports will be produced?

Internal Documentation

Comments within the program tell the user what the program does and how it works. Comments should be added as the program is written. They should include

- a header stating the program name, programmer, when written, purpose, and when last modified (see app. VII); and
- explanations of tasks.

Workpaper Documentation

The documentation should allow a reviewer or referencer to independently replicate the work done (see the General Policy Manual, 11.0-1). For programs, this should include

- a list of related files and a brief description of each;
- a list of data base structures and a brief description of each field's purpose;
- the printed programs;
- a list of memory variables with the purpose of each, as well as the possible values they could take;
- a description of any report forms used. (In dBASE, use MODIFY REPORT to display the format, then use SHIFT-PRTSC to print the column contents. An

alternative is to mark a printed report with fields and calculations.);
- a list cross-referencing related files and memory variables; and
- a system flow chart showing how programs and data bases interact (app. VI is an example).

Tips for Use With More Complex Programs

The following tips apply to larger and more complex programs.

- Consider compiling the system. It leads to faster program execution, ensures consistent analysis and results, allows more error checks, and does not require a copy of dBASE for subsequent execution.
- Provide a menu option for reindexing files. Index files can become damaged or outdated.
- Provide a way to recover from printer problems, such as paper jams and running out of paper.

Using Compilers With dBASE

dBASE is an "interpreter," meaning every time a command is given, it is changed into machine code. This occurs whenever a program is run. Compilers (such as Clipper) are more efficient. They convert programs to machine code that never needs to be interpreted again.

Compilers have several advantages. They identify programming errors and create faster, "executable" programs that can be run from DOS (Disk Operating System), eliminating the need for a copy of dBASE. Users cannot modify program code and need not know dBASE.

Be aware that compilers may not be 100 percent compatible with the data base software. Interactive commands like APPEND, EDIT, etc., cannot be used in a compiled program (though forms of these commands, such as "APPEND BLANK," may be used).

Testing

Test programs to ensure desired results and to handle unusual or erroneous data. Run a sample of data and compare it with predetermined results. Test to see how error conditions are handled. Then document the testing with

- listings of test data,
- description and extent of tests, and
- actual results and comparisons with predetermined results.

Referencing

The Communications Manual (12.13-1) requires that each GAO product be referenced to ensure accuracy and adequate support. The evaluator-in-charge (EIC) and referencer are responsible for the quality of the report; the technical reviewer's role, if needed, is more limited. This chapter sets out the responsibilities of

- the EIC,
- the referencer, and
- the technical reviewer.

EIC's Responsibilities

The EIC is responsible for ensuring the product is ready to be referenced. Beforehand, the EIC or a designee must review all workpapers, ensuring that they meet GAO standards. This includes using appropriate summary schedules and cross-indexing related workpapers to form an audit trail.

The EIC should document the steps taken to ensure source data validity. Was a reliability assessment performed? If so, does the product explain the steps GAO took to perform this assessment? If a reliability assessment was not done, does the product say so and give appropriate cautions concerning the results?

It is the EIC's responsibility to ensure the product identifies any serious data limitations. The workpapers should show what is included in the data, the period covered, the source, and any information that is excluded and why.

The EIC should ensure that the workpapers have enough instructions so the referencer or a technical reviewer (see below) can duplicate the work done. In addition, the EIC should ensure that hard copies of key computer-assisted schedules that support the facts and figures in the draft product have been printed, reviewed, and approved. (See Communications Manual (12.13-3.) The EIC should also ensure that an independent, qualified person has examined

and approved the data analysis methodology. This technical review is a part of the assignment process, not the technical assistance associated with referencing as discussed below. As such, the review should be documented and made part of the workpapers.

Referencer's Responsibilities

The referencer is responsible for ensuring that

- required supervisory and technical reviews were performed and proper quality control procedures were followed;
- the product presents data accurately and fairly; and
- information, findings, and conclusions are supported by sufficient evidence.

The referencer need not necessarily be familiar with data base applications or be a technical expert. GAO policy provides that if the referencer encounters highly technical material or a very complex data base application, technical assistance can be requested, if necessary, to ensure the accuracy of computations or the soundness of the data analysis methodology.

In checking for required reviews and quality control procedures, the referencer should ask:

- Is there evidence that the supervisor has reviewed each workpaper indexed in the final product?
- Is there documentation that an independent technical reviewer has verified the computations and approved the data analysis methodology? (See Communications Manual, 12.13-5.)
- Does the product contain a statement regarding the validity of the data used and how this was assessed?
- Is there documentation that a reliability assessment was performed? If not, is this reported?
- Is there evidence of data entry or transfer verification?

- Do the workpapers document changes to the source data as a result of data entry or transfer verification?
- If data have been transmitted, do the workpapers describe the steps performed?

The referencer must also verify that figures or statements of fact agree with the support provided. Generally, it should not be necessary to go beyond the top schedules of the workpapers, although selected critical items should be checked back to the detailed supporting workpapers. This should be done to ensure that supervisory reviews are effective and that cross-indexing is accurate. Tested items should be marked on the workpaper and on the referenced draft to indicate they have been verified. The referencer should sign and date the workpaper to indicate that he or she entered these marks. The referencer is not expected to check computer programs or verify that computerized data have been keyed correctly, (these are responsibilities of the technical reviewer and the EIC, respectively).

When graphs are used, the referencer should check the axes and legends to be sure they reasonably represent the data from the workpapers and that the graphs are not misleading.

The referencer must determine the adequacy of the supporting sources and the logical soundness of the facts and evidence provided.

Technical Reviewer's Responsibilities

As discussed above, the referencer may request technical assistance. The technical reviewer, if needed, is an assistant to the referencer, and must not have been involved in developing or operating the data base application. The reviewer should examine the workpapers to ensure proper methods and correct use of automated techniques. (The referencer is responsible for proper use of data in the product.) The technical reviewer is responsible for

verifying the accuracy of figures or statements of fact based on computer-generated data. The reviewer does not have to repeat the analysis but must evaluate the soundness of the methodology, including checking dBASE program files or commands.

The reviewer should generally check each analysis that produced figures used in the product, or do sufficient tests to assure the accuracy of the data. A well-documented analysis will make this easier and quicker. The reviewer must understand the purpose of each step in the analysis, i.e., what the staff was trying to do. Then the reviewer must check the data base software commands used to determine whether they would achieve the intended result. In the example that follows, the reviewer would note that the program does not count budget items equal to $50, $100, or $250.

```
*********************************************************
* PROG1.PRG                                             *
* Counts the budget items by dollar categories          *
* Written in dBASE III PLUS v. 1.1                      *
* Written by John Smith                                 *
* Last modified on 12/15/88                             *
*********************************************************
USE Budget
COUNT FOR Budgetitem < 50
COUNT FOR Budgetitem > 50 .AND. Budgetitem < 100
COUNT FOR Budgetitem > 100 .AND. Budgetitem < 250
COUNT FOR Budgetitem > 250
*EOP (Prog1.Prg)
```

The technical reviewer should mark each figure verified. When finished, the reviewer should write a brief statement for the referencing notes describing the technical review and summarizing conclusions on the use of the computer-generated data. Any cautions or problems should be clearly explained. The reviewer should sign and date the document. If

as a result of the technical review a problem was identified but not corrected, the referencer must determine the impact of the problem on the accuracy of computations or the soundness of the data analysis methodology.

Introduction to the Appendixes

Appendixes I through VII, taken from a review of the Department of Defense's Medical Cost Recovery, are intended to illustrate the tasks and documentation necessary to use a data base application. Since every assignment has unique data processing needs, these examples should not be considered inclusive.

The assignment evaluated whether Defense had recouped expenses at military medical facilities for civilian care. The GAO staff transferred an agency data base into dBASE, worked with data interactively, and programmed a data collection and reporting system. The application has been simplified for illustration.

Data Base Structure

Data Base Structure

The following example of a dBASE data base structure was designed for patient data at various medical facilities. It shows field names, the data type for each field (character, numeric, date, logical, or memo), field lengths, and decimals. dBASE produces this report with the command LIST STRUCTURE.

```
Structure for data base:  C:recovery.dbf
Number of data records:     1401
Date of last update   :  12/08/89

Field  Field Name  Type        Width    Dec

    1  FACILITY    Character       4
    2  SAMPLENUM   Numeric         3
    3  REGSTRNUM   Numeric         7
    4  SSN_FMP     Character      11
    5  PATCATGRY   Numeric         1
    6  TYPETRTMNT  Numeric         1
    7  CHARGES     Numeric        12      2
    8  TRTMNTDATE  Date            8
    9  DISCHARGE   Date            8

**  Total  **                     56
```

Data Base Dictionary

A data base dictionary explains the source, format, and meaning of the data. It describes acceptable values and may provide sample entries. Its purpose is to explain the data base and ensure proper interpretation.

Data Dictionary for the Recovery Data Base

Source of data: the "Record of Inpatient Treatment" in the inpatient medical file.

1. **Facility** (field name: FACILITY; character, 4 long) abbreviation of medical facility providing service e.g., WHMC — Wilford Hall Medical Center; BAMC — Brooks Army Medical Center

2. **Sample Number** (field name: SAMPLENUM; numeric, 3 long) sample number assigned by GAO; can contain numbers from 1 to 999; uniquely identifies record

3. **Register Number** (field name: REGSTRNUM; numeric, 7 long) number assigned by medical facility at time of admission (inpatient only); unique for each case

4. **SSN/FMP** (field name: SSN_FMP; character, 11 long) Social Security Number (positions 1-9) and Family Member Prefix (FMP) (positions 10-11); the Social Security Number is that of the enlisted/retired person; the FMP identifies the actual member of that person's family who is receiving the medical care

5. **Patient Category** (field name: PATCATGRY; numeric, 1 long) patient category represents the patient's military status
   1 - active duty (AD)
   2 - dependent AD
   3 - dependent of deceased
   4 - retiree
   5 - dependent retiree
   6 - civilian

7 - other
9 - missing record

6. **Type of Treatment** (field name: TYPETRTMNT; numeric, 1 long) type of treatment refers to whether the patient received treatment as an inpatient, in the emergency room, or a clinic
1 - inpatient
2 - emergency room clinic
3 - physical therapy clinic
4 - orthopedic clinic
5 - neurosurgery

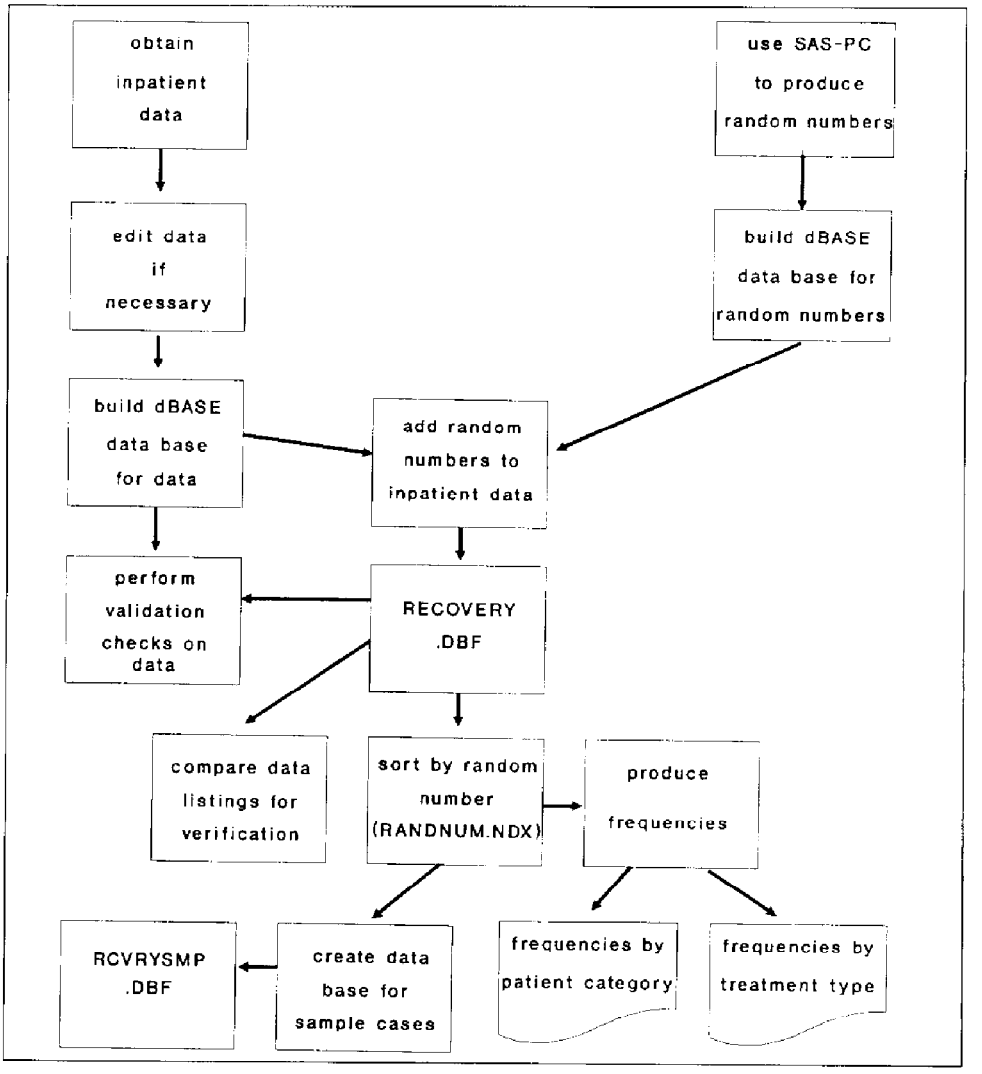
7. **Charges** (field name: CHARGES; numeric, 12 long, 2 decimals) amount charged by the medical facility for the patient visit

8. **Admission Date** (field name: TRTMNTDATE; date —mm/dd/yy) date patient was admitted (inpatient) or received treatment (outpatient)

9. **Discharge Date** (field name: DISCHARGE; date —mm/dd/yy) date patient was discharged from medical facility (inpatient)

Analysis Plan Flow Chart

An analysis plan flow chart shows planned data collection and analysis steps. It begins with obtaining the source data and ends with the desired reports.



Alternate File of a dBASE Interactive Work Session

An alternate file records dBASE commands and their results and saves them to a disk file. To clarify the work done, add narrative comments with word processing software. The command SET ALTERNATE TO creates the disk file; SET ALTERNATE ON/OFF begins or ends the recording. The following is an example of an alternate file.

```
Note: This is the alternate file of a dBASE work session on
12/08/89.  All notes added later will be in bold print.
Programmer:  J. Smith

Note:  The purpose of this work session was to
   --  copy random numbers created by a SAS-PC program to dBASE
       data base; and
   --  attach the random numbers to the RECOVERY data base.

Note:  First, create an empty data base for the random numbers.

   . ? DATE()
   12/08/89
   . ? TIME()
   15:48:30
   . CREATE RANDOM

   . LIST STRUCTURE

   Structure for database : B:random.dbf
   Number of data records :        0
   Date of last update    : 12/08/89
   Field  Field name  Type       Width     Dec
       1  RANDNUMBER  Numeric       10       8
   ** Total **                      11

Note:  Then, copy the random numbers created by SAS-PC (contained
in RANDOM.TXT) to RANDOM data base.

   . APPEND FROM RANDOM.TXT SDF

      1401 records added
   . CLOSE DATABASES
```

```
Note:  Modify the RECOVERY data base to accommodate the random
numbers.

   . USE RECOVERY

   . LIST STRUCTURE
   Structure for database:  C:recovery.dbf
   Number of data records:     1401
   Date of last update    :  12/08/89
   Field  Field Name  Type       Width     Dec
       1  FACILITY    Character      4
       2  SAMPLENUM   Numeric        3
       3  REGSTRNUM   Numeric        7
       4  SSN_FMP     Character     11
       5  PATCATGRY   Numeric        1
       6  TYPETRTMNT  Numeric        1
       7  CHARGES     Numeric       12       2
       8  TRTMNTDATE  Date           8
       9  DISCHARGE   Date           8

   **  Total  **                    56

   . MODIFY STRUCTURE

   . LIST STRUCTURE

   Structure for database : recovery.dbf
   Number of data records :   1401
   Date of last update    : 12/08/89
   Field  Field name  Type       Width     Dec
       1  FACILITY    Character      4
       2  SAMPLENUM   Numeric        3
       3  REGSTRNUM   Numeric        7
       4  SSN_FMP     Character     11
       5  PATCATGRY   Numeric        1
       6  TYPETRTMNT  Numeric        1
       7  CHARGES     Numeric       12       2
       8  TRTMNTDATE  Date           8
       9  DISCHARGE   Date           8
      10  RANDNUMBER  Numeric       10       8

   ** Total **                      66

   . CLOSE DATABASES
```

Note: A program was written (see WP F-A6) to replace the empty random number fields in the RECOVERY data base with the actual random numbers in the RANDOM data base.

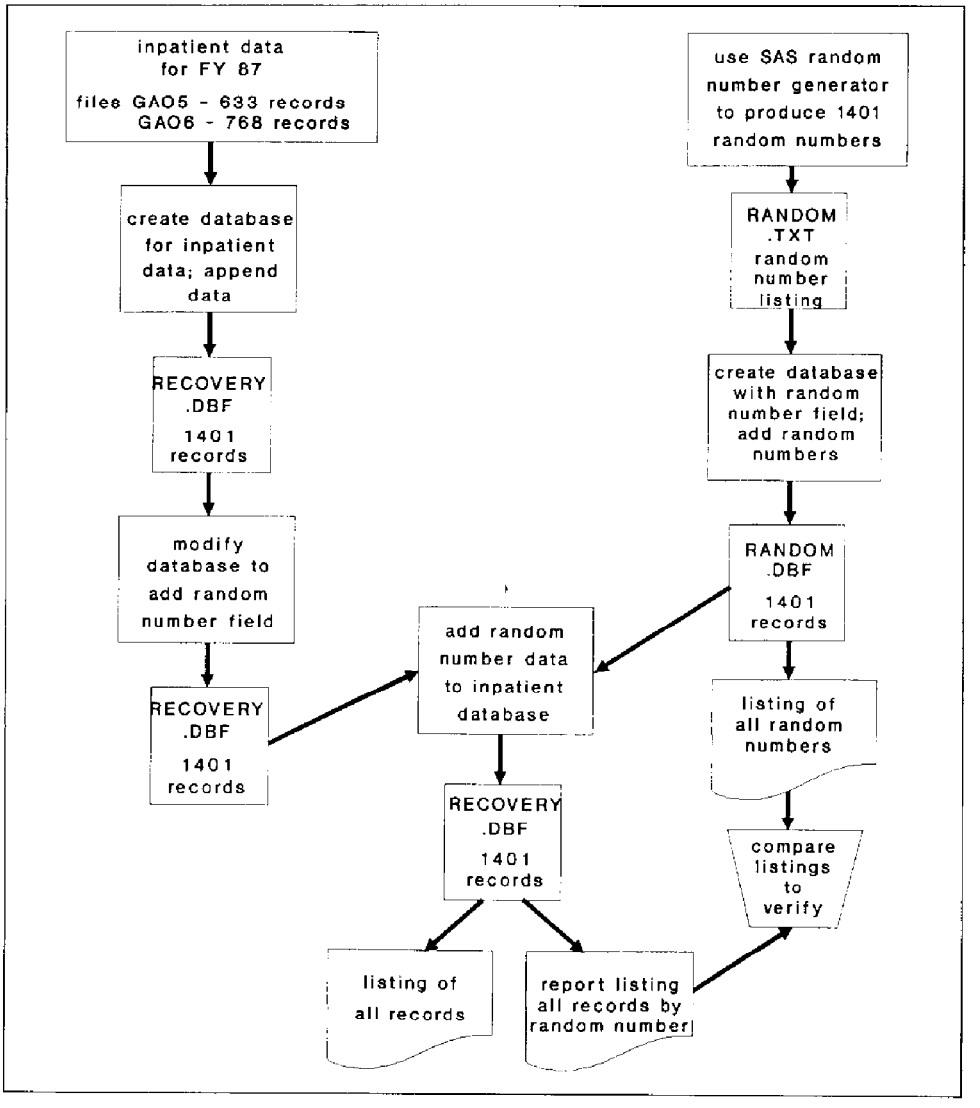
```
. MODIFY COMMAND SKIPIT

. DO SKIPIT

1401 records replaced
```

Note: A listing of all the random numbers from the RECOVERY data base was prepared to compare with the SAS-PC listing of random numbers to ensure that the replacement process worked.

```
. LIST ALL RANDNUMBER

Record#  RANDNUMBER
      1  0.76882922
      2  0.64278364
      3  0.35448651
      4  0.68823158
      5  0.86214386
      6  0.71195812
      7  0.16187964
      8  0.69224314
      9  0.34548115
     10  0.44520226

*** INTERRUPTED ***

CLOSE DATABASES

SET ALTERNATE OFF
```

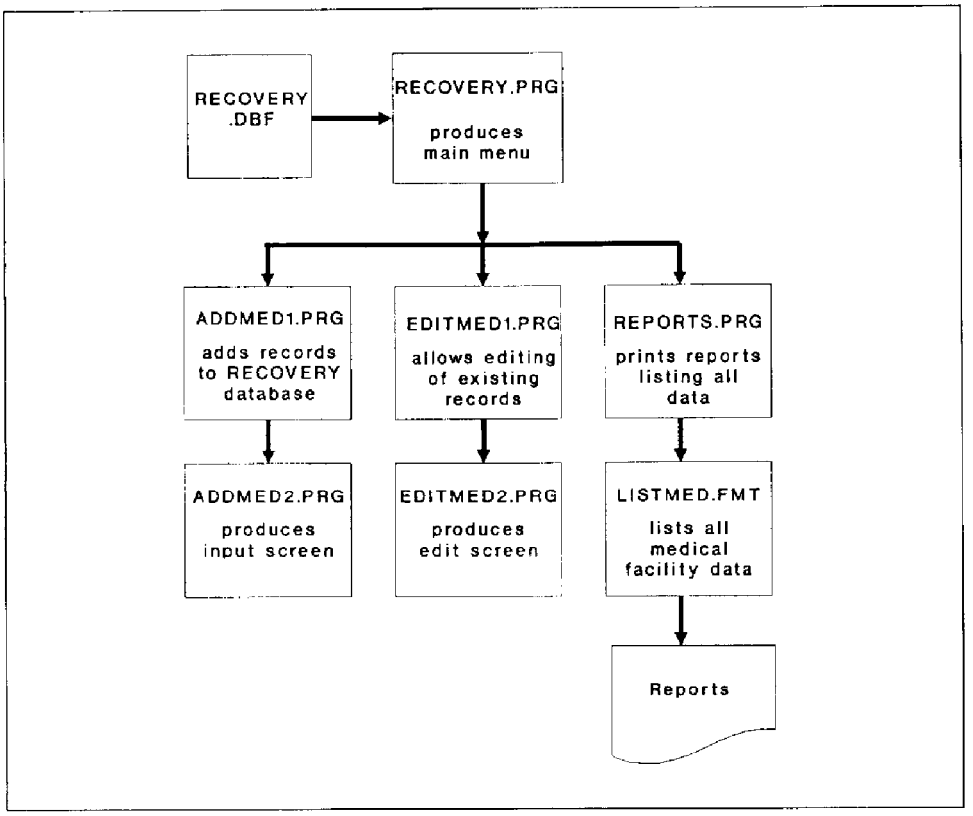
Process Flow Chart

A process flow chart shows actual steps in data collection, verification, and analysis.



Data Collection and Reporting System Flow Chart

A system flow chart illustrates how programs and data bases interact.



Sample Program

Here is an example of a dBASE program that stores commands to a disk file where they can be executed at any time. Asterisks denote comments that are not commands; hence, they cannot be executed.

```
*********************************************************
* PROGRAM NAME: ADDMED1.PRG                             *
* PROGRAMMER  : J. SMITH                                *
* DATE WRITTEN: 12/8/89; LAST MODIFICATION: 12/20/89    *
* PURPOSE: This program creates a screen to enter       *
*  data into the RECOVERY database                      *
*********************************************************
SET SAFETY OFF
SET TALK OFF
DO WHILE .T.
  STORE SPACE (3) TO MSAMPLE
  CLEAR
*
* SET UP SCREEN TO ACCEPT SAMPLE NUMBER
*
  @ 1,14 SAY "************************************************"
  @ 2,14 SAY "*              MEDICAL COST RECOVERY           *"
  @ 3,14 SAY "*                 JOB CODE 123456              *"
  @ 4,14 SAY "*   INPUT SCREEN FOR MEDICAL FACILITY PORTION   *"
  @ 5,14 SAY "************************************************"
  @ 8,10 SAY "If you wish to enter information on a new sample"
  @ 8,59 SAY "number,"
  @ 10,10 SAY "enter the new sample number below.  If you have"
  @ 10,59 SAY "finished"
  @ 12,10 SAY "all entries for the current session, press the"
  @ 12,57 SAY "Enter key"
  @ 14,10 SAY "and you will return to the main menu."
  @ 17,25 SAY "Sample Number:"
  @ 17,41 GET MSAMPLE
  READ
*
* IF A SAMPLE NUMBER WAS NOT ENTERED, RETURN TO THE MAIN MENU
*
  IF LEN(TRIM(MSAMPLE))=0
    RETURN
  ENDIF
*
* APPEND A BLANK RECORD TO THE RECOVERY DATABASE FOR ADDITION
*  OF DATA ON NEXT SAMPLE CASE
```

```
*
  USE RECOVERY INDEX SAMPLE
  APPEND BLANK
  REPLACE SAMPLENUM WITH VAL(MSAMPLE)
  CLEAR
*
* MOVE ON TO THE NEXT INPUT SCREEN WHERE MORE DATA WILL BE ADDED
*
  DO ADDMED2
  REINDEX
ENDDO
RETURN
* END OF PROGRAM
```

Major Contributors to This Publication

Dallas Regional Office	Barbara A. Johnson, Senior Evaluator David W. Irvin, Senior Evaluator/TAG Manager
Los Angeles Regional Office	Barbara Y. House, Senior Evaluator
Resources, Community, and Economic Development Division, Washington, D.C.	Annette Wright, Senior Evaluator
Information Management and Technology Division, Washington, D.C.	Daniel C. White, Director, Planning and Reporting William C. Oelkers, Associate Director Dennis L. O'Connor, Reports Analyst

Bibliography

dBASE III Plus Manual, Vol. II, Torrance, CA: Ashton-Tate, 1985.

Castro, Luis, Jay Hanson, and Tom Rettig. "Program Documentation," Advanced Programmers' Guide, Culver City, CA: Ashton-Tate, 1985, pp. 141-151.

Liskin, Miriam. Advanced dBASE III Plus Programming And Techniques, Berkeley, CA: Osborne McGraw-Hill, 1987, pp. 717-725.

Seman, Stewart O. "Controlling the Quality of Electronic Work Papers," The GAO Review, 21, (No. 4, Fall 1986), pp. 11-14, 33.

U.S. General Accounting Office. Information Management and Technology Division. Assessing the Reliability of Computer-Processed Data, (8.1.3), Washington, D.C., 1990.

— Preparing, Documenting, and Referencing Lotus Spreadsheets, (11.1.3), Washington, D.C., 1987.

— Project Manual, Washington, D.C., 1989, chapter 10.1.

Ordering Information

The first five copies of each GAO report are free. Additional copies are $2 each. Orders should be sent to the following address, accompanied by a check or money order made out to the Superintendent of Documents, when necessary. Orders for 100 or more copies to be mailed to a single address are discounted 25 percent.

U.S. General Accounting Office
P.O. Box 6015
Gaithersburg, MD 20877

Orders may also be placed by calling (202) 275-6241.

United States
General Accounting Office
Washington, D.C. 20548

Official Business
Penalty for Private Use $300

First-Class Mail
Postage & Fees Paid
GAO
Permit No. G100