JPL Publication 87-22

Guidelines for Preparing Software User Documentation

Diane F. Miller

October 1, 1987

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL Publication 87-22

Guidelines for Preparing Software User Documentation

Diane F. Miller

October 1, 1987

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

Clear, easy-to-use software user's manuals make strong demands on special technical communication techniques. Principles and guidelines are given for analyzing the audience and dealing with wide ranging backgrounds of potential users. Types of information to be included in a complete manual are suggested, with a technique for creating a user-oriented rather than process-oriented organization. Accuracy verification is emphasized. Simple tips are given for formatting for quick comprehension and reference, for deciding on packaging, for creating helpful illustrations and examples, and for setting up clear and consistent conventions. Simple guidelines are offered for writing clearly and concisely and for editing.

TABLE OF CONTENTS

TABLE OF CONTENTS (Contd)

TABLE OF CONTENTS (Contd)

Figures

TABLE OF CONTENTS (Contd)

Figures (Contd)

Tables

SECTION 1

WHAT TO EXPECT FROM THIS GUIDELINE

1.1 PURPOSE AND SCOPE

This document is intended as a guideline to assist anyone responsible for preparing a user's guide for any software product. It is especially oriented toward end-user products intended for a wide range of users. Such software might include management and administrative data maintenance systems, communication software, electronic mail systems, and other general purpose data processing applications.

In order to survive and prosper, companies in the competitive software development business must have as their goal to aggressively seek out and satisfy the requirements of their prospective users. This goal implies that the products provided should be easy to use. Thus, the management of such companies are concerned with the quality and usability not only of the software products released to the user marketplace, but also with the documentation that accompanies them. If the documentation does not promote understanding of and satisfaction with the software product, the product may not succeed. Even if purchased, it may be used incorrectly or not at all. It will attain a poor reputation, as will the company that produced it.

A software user's manual is a technical document with special demands on it. Microcomputers are found in the work place on the desks of everyone from clerks and secretaries to chief executive officers. And now a significant percentage of homes also have microcomputers. Many of these users have little background in computers; yet the jobs they have to do on these computers call for a great deal of learning--fast learning. Because of consumer demand and intense competition, end-user software is becoming increasingly self-explanatory and easy to use. User's manuals, although greatly improved over the last few years, still have a way to go. For the most part, they still make the software application seem more difficult than it really is.

This document is written in recognition of the special demands on the writers of user's manuals. Writing a user's manual that is complete, clear, concise, and easy to use requires consideration of many factors. Software implementors generally have neither the training nor the time to fully do justice to the user documentation task. This guideline is thus addressed to the skilled technical writer or editor already versed in the fundamental principles of good technical writing. We describe here the characteristics of software user's guides most often preferred by users.

1.2 AUTHORITY

The guidelines given in this document are based on

- User preference, as expressed by managers, user consultants, user educators, and users themselves;

- Empirical studies in the fields of technical communication, education, and cognitive psychology;

- Authoritative texts and references on English usage and style.

The following sources were among those used in the preparation of this guideline and are recommended reading or reference for anyone writing software user's manuals:

(1) William Strunk Jr. and E.B. White (1979), The Elements of Style, Third Edition, MacMillan Publishing Co., Inc., New York, NY.

(2) Edmond H. Weiss (1985), How to Write a Usable User Manual, ISI Press, 3501 Market St., Philadelphia, PA 19104. (This book lists many other useful books and resources).

(3) William A. Sabin (1984), The Gregg Reference Manual, Sixth Edition, Gregg Division/McGraw-Hill Book Company, New York, NY.

(4) John Brogan (1973), Clear Technical Writing, McGraw-Hill Book Company, New York, NY.

(5) The American Heritage Dictionary of the English Language, Houghton Mifflin Company, Boston, MA.

(6) Proceedings of the 32nd International Technical Communication Conference, May 19-22, 1985, Houston, TX, sponsored by the Society for Technical Communication, 815 Fifteenth Street Northwest, Washington, DC 20005.

(7) Proceedings of the 33rd International Technical Communication Conference, May 11-14, 1986, Detroit, MI, sponsored by the Society for Technical Communication, 815 Fifteenth Street Northwest, Washington, DC 20005.

(8) Technical Communication: Journal of the Society for Technical Communication, 815 Fifteenth Street Northwest, Washington, DC 20005.

1.3 ORGANIZATION

This document is organized around the basic characteristics of a good user's guide--that is, one which is

(1) Aimed at the right audience
(2) Complete
(3) Accurate
(4) Organized for easy reference
(5) Attractive and handy to use
(6) Formatted for quick comprehension
(7) Generously sprinkled with illustrations and examples
(8) Consistent and clear in use of conventions
(9) Written clearly and concisely
(10) Carefully edited before release

SECTION 2

ANALYZE YOUR AUDIENCE

2.1 WHO ARE THE USERS AND WHAT DO THEY DO?

Every software product created reaches a different set of users, with differing tasks and differing backgrounds. Therein lies the challenge. Before you can write a user's manual, you must find out whom you are addressing. Study the user community. You can learn much by talking with some of the managers and supervisors of the least knowledgeable users. But there is no substitute for talking with a few of these most typical users yourself. Your objectives are to

(1) Find out how much they already know about microcomputers (and other computers, if applicable).

(2) Find out how familiar they are with software similar to or used in connection with the software you are documenting.

(3) Find out how they will use the software system in doing their jobs.

2.2 STATE YOUR ASSUMPTIONS

Let the reader know immediately what it is you expect she or he already knows. For example, suppose you are documenting an electronic mail system but do not feel it is reasonably within your scope to give detailed procedures for using communication hardware and software to access the mail system. In this case, state in the beginning that you assume the user is already set up for communications and knows how to access the appropriate computer. As another example, if you are writing a manual on how to upload an electronic spreadsheet containing budget planning information from the user's microcomputer to the company mainframe, state your assumption that the user already knows how to do financial planning and how to fill in the spreadsheet.

In addition to stating your assumptions about what the user knows, state your assumptions about how the user will use the software system. You may find that users have a wide range of possible uses for the software. Stating these will help users to better understand how the software may serve their own needs.

The assumptions you make about the prior knowledge of the intended audience of a manual, the possible uses of the software product, and the extent of your responsibility as the author of the manual are negotiable. Depending on your organization and how it operates, this agreement may be reached between you and your management, or between your management and the intended users of the software. In any case, these assumptions must be made carefully case by case. When the decision is reached, however, be sure to clearly and boldly state the outcome.

2.3 AIM FOR A RANGE OF USERS

You cannot write one manual that will fit everyone perfectly. Nevertheless, before you begin to write, mentally formulate the characteristics of three groups of users:

(1) The user with the least knowledge who could reasonably be expected to use your system.

(2) The most technically advanced user, who could need considerable technical information about the system.

(3) The most typical user.

It is possible that two or all of these categories could refer to the same user. For example, the technically advanced user could also be the most typical. Usually, though, you will be faced with a range and will have to decide where to put your primary focus.

If a document is well organized and formatted, it can acknowledge and address the needs of the novice without wasting the time of the adept. To serve most needs:

(1) Include complete information for the novice user (but avoid a patronizing tone).

(2) Put technical information of interest to advanced users in an appendix or even a separate volume.

(3) Consider writing two volumes: a simple step-by-step tutorial for the novice and a detailed reference manual for the more experienced.

(4) Write procedural information simply, assuming no problems or exceptions. Then include exceptions, error messages, troubleshooting help, etc., in carefully organized appendixes.

(5) Include an index.

SECTION 3

GATHER COMPLETE INFORMATION

This section is intended to help in the information-gathering stage, before actual writing of the user's manual begins. It covers the types of information you need to consider obtaining for inclusion into the user's guide. How that information is to be organized is addressed in Section 5.

3.1 INFORMATION ABOUT SUPPORTING SOFTWARE

Most users prefer to have all the information they need to operate an application together in the same place. For example, users of an electronic mail system must sign on to the system before doing anything else. However, if the manual for the electronic mail system does not reveal anything about telecommunications packages and how to use them, many users will be frustrated. (If most of the users are already familiar with such concepts, the detailed information for the novice can be placed in an appendix and referenced, out of the way of the majority of users who do not need it.)

Some applications, such as a facility to upload and download information from user workstations to a corporate mainframe computer, require use of other commercial software products such as spreadsheets and special communication packages. The user's manual for such systems should include at least a minimum of relevant information about the supporting commercial package. It is not necessary to reproduce the entire user's manual for the auxiliary product, but to include enough to instruct the user in the operations of the subject system for its basic functions. For more detailed information, the user can be referred to specific portions (sections or pages) of the commercial documentation.

3.2 HOW TO OBTAIN USER AUTHORIZATION

For software operating on a mainframe or other shared computer, find out the complete procedure users are to follow in gaining access to the system and any relevant data set(s). Include a copy of any required user authorization form. Give title (not name), phone number, and mailing address of the person to whom the authorization form should be sent or whom users should call for access. If users must execute a special log-on in order to establish a password, spell out that procedure.

These administrative procedures are often the last items to be established when a new system is being developed. However, they are of primary importance to the user. The writer of the user's manual can provide an impetus to getting the procedures established.

If the user authorization procedure is unstable, the information can be included as a separate handout to accompany the user's manual. Thus, when the procedure changes, the information can be updated and distributed to the users with a minimum of effort.

3.3 REQUIRED USER HARDWARE/SOFTWARE AND CONFIGURATION

Describe the type of computer or terminal required to operate the system. Include required memory, storage capacity (hard disk, floppies, or both), communication hardware, printers or plotters, and any other relevant components. Also describe any other software (including the release or version numbers) required to support or augment the system. Be sure to include the supported version(s) of the disk operating system (DOS) and any necessary communication software.

3.4 WHAT THE SYSTEM DOES AND WHAT IT INCLUDES

Describe the overall function of the system. Conceive a model of the system and use diagrams or other visual aids to help the user see how the system works or is organized. Different or more detailed diagrams can be used later to help the user understand how to use the system.

Describe what is included in the system from the user's point of view. For example, the system may include a floppy diskette, a user's manual, a quick reference card, and a function key template. Or it may contain only a user's manual (as for a mainframe system).

For systems including a floppy diskette with files to be installed on the user's microcomputer, list the files contained on the diskette, briefly describe the function of each file, and tell whether the file is required for the functionality of the system or whether it is optional (e.g., a file used for a tutorial exercise). This information allows the user the opportunity to save hard disk space by not installing optional files.

3.5 HOW TO INSTALL THE SYSTEM (IF APPLICABLE)

Give detailed instructions for installing the system, if applicable. Include all necessary information for linking the system to other installed software packages, for installing print drivers, for configuring for installed communication hardware, and any other relevant considerations.

3.6 HOW TO PERFORM ALL FUNCTIONS OF THE APPLICATION

3.6.1 Explain Overall Operational Conventions

Include detailed information, as applicable, on

(1) How to log on to or start up the system

(2) How to log off or exit the system

(3) Basic screen format(s)

(4) How to navigate within the system; explain how to

(a) give commands
(b) select options
(c) move cursor
(d) add, modify, or delete data
(e) move through screens
(f) cancel a command or action

(5) How to obtain help information

(6) How function keys are used

3.6.2 Explain How To Perform Specific User Functions

Describe in detail and give examples of how the user should accomplish each function designed into the system. The approach you take to this description depends on the system. For simple applications (such as an uploading or downloading function), the user may exercise every function of which the application is capable each time he or she uses it. In such a case, a simple step-by-step procedure may suffice.

For other more complex applications, the individual user may never use all the capabilities, and, because of her or his specific job function, may be primarily concerned with only a few. The functionality of the system must be explained in terms of the tasks facing the user. In gathering information for the user's manual, anticipate how the user will use the system, and judge the completeness of the information accordingly.

3.6.3 Explain How To Exit Gracefully From Any Operation

Sometimes users begin an operation, such as printing, sorting, downloading files, or compiling programs, and then notice something is going wrong or just change their minds. They need to know how to stop the operation without losing data and without having to resort to warm booting or turning off the computer. Include this information in a conspicuous place.

3.7 HOW TO RECOVER FROM ERRORS

In describing step-by-step procedures for operating a system, assume everything happens normally. However, anticipate the worst by including an appendix of all possible error messages and conditions, along with their probable causes and how to correct the problems. Include error messages that might be generated by any auxiliary software product used in conjunction with the system.

3.8 EXAMPLES OF ALL SCREEN DISPLAYS AND PRINTED OUTPUT

Gather examples of all screens generated by the application, using sample data and/or user input. Screen illustrations should show users exactly what they will see on the screen (except, of course, for specific data). Somehow differentiate in screen examples between the computer output (screen titles, field names, prompts, etc.) and the user's input. (See Section 8 for some suggestions.)

If the system includes printed reports, include examples (with descriptions) of each type of report. The report samples must be clearly legible. These may be placed in a separate section on reports or in an appendix.

3.9 EXAMPLES OF TYPICAL USER TRANSACTIONS

If relevant, prepare complete examples of typical transactions a user might have with the system. For complex systems, these may be in the form of complete tutorials, or, for simpler systems, may be simple screen facsimiles which show computer output and user input (somehow differentiated).

In the case of complex systems, a hypothetical (yet realistic) user task situation could be set up, and then the user guided through the task using the computer. A tutorial would include numbered steps to be performed by the user, the associated computer response, and exact facsimiles of the computer displays expected.

3.10 HOW TO OBTAIN AND INSTALL UPDATES

A procedure to distribute updates to the software and the documentation needs to be established for every product, regardless of the computer environment in which the product operates.

Systems accessed via a communications link to a central computer must, of course, be configuration controlled, and documentation must be updated as new versions are released. If users need to do anything to obtain those updates, or notify anyone of any changes of mailing address in order to remain on the distribution list, be sure to include this information in the user's guide.

Documentation for microcomputer software must include the procedure for obtaining new releases (diskettes) of the software and how to install those new releases.

3.11 GLOSSARY OF TERMS

It can well be argued that a properly written user's manual will not need a glossary of terms because it will not contain any acronyms, company jargon, computer or other technical terminology, or any but the most common everyday concepts. However, it is almost always necessary to introduce at least a few possibly new words and concepts to the user. These should, of course, be thoroughly explained where they are first introduced.

In addition, these terms should be gathered alphabetically into a glossary of terms, usually as an appendix. Definitions should be clear and complete. Acronyms should not only be spelled out, but defined as well.

3.12 PHONE NUMBER TO CALL FOR HELP

Try as you may, users will come up with problems or questions not covered in your document. Or, more likely, they will be too frustrated or in too big a hurry to consult your document. Every user's guide must include the phone number of a knowledgeable person who is available for help. This specific individual should not be named, but a title (or organization) given and a phone number. Then, of course, the phone number should get results--but that is probably out of your hands.

Because of possible phone number changes, you may wish to avoid embedding the phone number in the document. You may wish to consider such alternatives as placing a sticker on the front cover of each manual or including a separate flier listing phone numbers for both user authorization and help.

3.13 FORM FOR USER FEEDBACK

Include at the back of your user's guide a form for user feedback on both the software system and the documentation. Such a form (as the one included at the back of this document) makes it easy for users to comment on the user's manual, adding any information they have found out through using the system, correcting any errors, or pointing out any other deficiencies or ambiguities. It also, of course, gives them an easy way to give positive feedback; however, do not be surprised or discouraged if this type of comment never appears. In any case, by including a feedback sheet, you again show the user your intention to be as helpful as possible.

SECTION 4

MAKE SURE INFORMATION IS ACCURATE

Completeness, accuracy, and organization for easy reference are the three most desired traits in user documentation. Total technical accuracy is often the most difficult to attain, because it is difficult to anticipate every possible use of the system.

Documentation writers (and editors, if the writer is not a documentation specialist) are urged to take the following steps to assure the accuracy of published user documentation:

(1) **Learn and use the system being described.** Make sure you understand any supporting products, such as the disk operating system (e.g., MS-DOS) and communication systems. Use the system the way a typical user would use it--which means you must also understand the user's application.

(2) **Talk to the developers.** As the writer of the user's manual, you should have almost a running dialogue going with the developers, especially during the information-gathering stage. Keep communication lines open for questions that occur at any time.

(3) **Submit early drafts for technical review by the developers of the software.** The systems analysts and programmers have spent many hours understanding users' needs and developing a design to fulfill them. While not every developer will understand every part of the system, take advantage of the area of specialty of each developer. Insist on written comments on your draft.

(4) **Check all procedures you have written by following your own instructions word for word,** just as a user might. Get someone else--a user would be ideal--to also validate your procedures.

(5) If possible, **distribute your draft to selected knowledgeable managers** in your company (in development and operations organizations, for example) **and users** for comments not only on your procedures, but also on background and conceptual explanations, descriptions of administrative processes, etc.

(6) **Distribute a beta test draft of the document to all system beta users** so that the user's guide can be validated with the software.

(7) **Do not distribute the draft to a wider audience for review until accuracy is thoroughly validated.**

SECTION 5

ORGANIZE FOR EASY REFERENCE

Many devices have been invented to make information easier to find in manuals. The larger the manual and the more complex the system, the more important the use of these devices becomes. The devices described here are adaptable to manuals of all sizes and levels of complexity; therefore, writers and editors are urged to use them whenever possible. A good criterion for a well-organized manual is that the user finds it just as easy to look up the answer in the user's guide as to find the answer by calling the help number listed at the front.

5.1 GROUP INFORMATION INTO USER-ORIENTED CHUNKS

Information "chunking" is a powerful concept in technical communication. It is a way of organizing information into bite-sized groupings. It makes the difference between intimidating or boring the reader, and quickly giving the reader precisely the information sought.

Technical documents are often written using a static, process-oriented, hierarchical outline, with three, four, or more different levels of headings. For most types of documents, this type of outline works fine. However, it should be understood that this type of outline serves primarily the writer, not the reader. A process-oriented, hierarchical outline helps the writer organize a vast amount of material into major groupings, then into subgroupings within each major group, and so on, so that a potentially overwhelming task becomes manageable.

However, we do not read and digest information hierarchically--nor do we think in terms of information hierarchies when we wish to find the answer to a specific question. Rather, we read sequentially. We learn sequentially. We think sequentially. When we want to find an answer, that particular subject is the most important one, and we do not think in terms of where it might be buried in the hierarchy of the writer's outline.

Therefore, a dynamic, user-oriented outline approach, as described in detail by Edmond H. Weiss (Reference 2, listed in Section 1.2 of this document), is recommended for user's manuals. With a user-oriented structure, this type of document is divided up into manageable modules that can be developed separately, and then linked together to form a working whole. It is different from the usual type of static, process-oriented outline in several important ways. The main difference is that the product of a user-oriented outline is a document that is much easier to reference.

Figures 5-1(a) and 5-1(b) show how a typical process-oriented outline has been converted to a dynamic user-oriented outline. Briefly, the following steps describe how to construct a user-oriented outline:

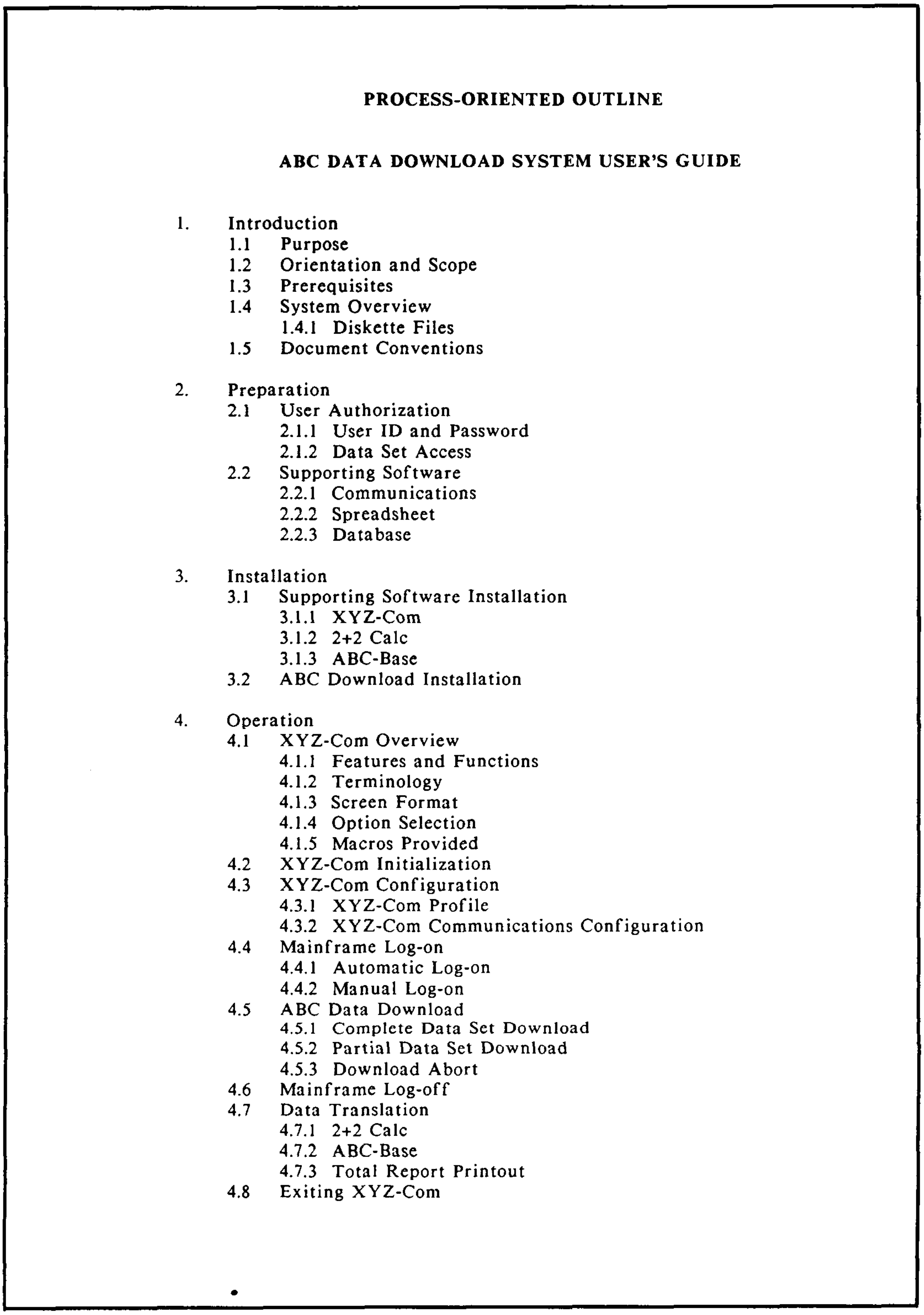
(1) Construct a complete hierarchical outline to help you group details and decide the sequence of topics. As usual, this outline will serve to check the completeness and logic of the presentation.

(2) Examine the outline in terms of user functions. You may wish to group some subjects together and treat them as one or divide others and treat parts of them separately. Think in terms of short modules, rather than sections and subsections. Each module treats a particular subject of interest to the user.

The trick is to create modules that independently treat just one topic or function of the application, while maintaining appropriate links to other material. Modules that are too big lack cohesiveness. Modules that are too small may have to reference many other modules, and the interrelationships with other information may become confusing. Modules will vary in length, but the ideal module is one or two pages long.

(3) Restructure modules into no more than two hierarchical levels.

(4) Give each module a descriptive dynamic headline meaningful to the user. Descriptive headlines often differ from conventional headings, as you can see in Figures 5-1(a) and 5-1(b). Conventional headings are often static, consisting mostly of nouns, often strings of three or four. Sometimes they are vague or abstract, such as "Introduction," "Overview," or "Purpose," and carry no useful information. Concise, descriptive headlines containing action verbs and specific information about what is contained in the module are much more useful in helping the reader locate information. Note some examples from Figure 5-1:



PROCESS-ORIENTED OUTLINE

ABC DATA DOWNLOAD SYSTEM USER'S GUIDE

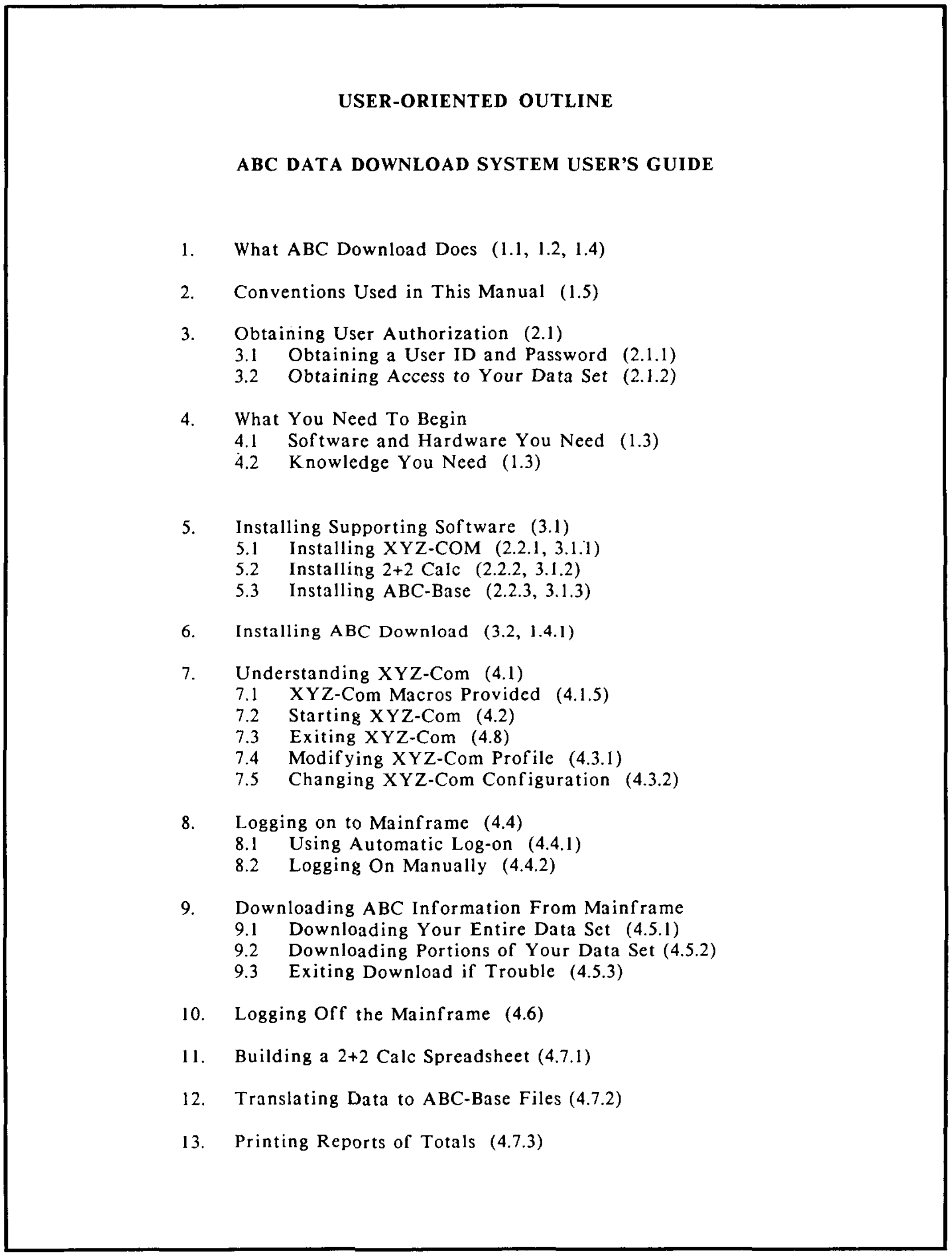
1. Introduction
1.1 Purpose
1.2 Orientation and Scope
1.3 Prerequisites
1.4 System Overview
1.4.1 Diskette Files
1.5 Document Conventions

2. Preparation
2.1 User Authorization
2.1.1 User ID and Password
2.1.2 Data Set Access
2.2 Supporting Software
2.2.1 Communications
2.2.2 Spreadsheet
2.2.3 Database

3. Installation
3.1 Supporting Software Installation
3.1.1 XYZ-Com
3.1.2 2+2 Calc
3.1.3 ABC-Base
3.2 ABC Download Installation

4. Operation
4.1 XYZ-Com Overview
4.1.1 Features and Functions
4.1.2 Terminology
4.1.3 Screen Format
4.1.4 Option Selection
4.1.5 Macros Provided
4.2 XYZ-Com Initialization
4.3 XYZ-Com Configuration
4.3.1 XYZ-Com Profile
4.3.2 XYZ-Com Communications Configuration
4.4 Mainframe Log-on
4.4.1 Automatic Log-on
4.4.2 Manual Log-on
4.5 ABC Data Download
4.5.1 Complete Data Set Download
4.5.2 Partial Data Set Download
4.5.3 Download Abort
4.6 Mainframe Log-off
4.7 Data Translation
4.7.1 2+2 Calc
4.7.2 ABC-Base
4.7.3 Total Report Printout
4.8 Exiting XYZ-Com

Figure 5-1(a). Example of a Typical Process-Oriented Outline. Note the three levels of information and the vague and static headings. Also note that most of the information the user will frequently reference is in a single major section.



USER-ORIENTED OUTLINE

ABC DATA DOWNLOAD SYSTEM USER'S GUIDE

1. What ABC Download Does (1.1, 1.2, 1.4)

2. Conventions Used in This Manual (1.5)

3. Obtaining User Authorization (2.1)
   3.1 Obtaining a User ID and Password (2.1.1)
   3.2 Obtaining Access to Your Data Set (2.1.2)

4. What You Need To Begin
   4.1 Software and Hardware You Need (1.3)
   4.2 Knowledge You Need (1.3)

5. Installing Supporting Software (3.1)
   5.1 Installing XYZ-COM (2.2.1, 3.1.1)
   5.2 Installing 2+2 Calc (2.2.2, 3.1.2)
   5.3 Installing ABC-Base (2.2.3, 3.1.3)

6. Installing ABC Download (3.2, 1.4.1)

7. Understanding XYZ-Com (4.1)
   7.1 XYZ-Com Macros Provided (4.1.5)
   7.2 Starting XYZ-Com (4.2)
   7.3 Exiting XYZ-Com (4.8)
   7.4 Modifying XYZ-Com Profile (4.3.1)
   7.5 Changing XYZ-Com Configuration (4.3.2)

8. Logging on to Mainframe (4.4)
   8.1 Using Automatic Log-on (4.4.1)
   8.2 Logging On Manually (4.4.2)

9. Downloading ABC Information From Mainframe
   9.1 Downloading Your Entire Data Set (4.5.1)
   9.2 Downloading Portions of Your Data Set (4.5.2)
   9.3 Exiting Download if Trouble (4.5.3)

10. Logging Off the Mainframe (4.6)

11. Building a 2+2 Calc Spreadsheet (4.7.1)

12. Translating Data to ABC-Base Files (4.7.2)

13. Printing Reports of Totals (4.7.3)

Figure 5-1(b). Example of a User-Oriented Outline Developed From the Process-Oriented Outline in Figure 5-1(a). Information is modularized so that no more than two levels of information are needed. Notice the dynamic descriptive headlines. (Numbers in parentheses refer to corresponding sections of the process-oriented outline in Figure 5-1(a).)

Static Heading	Dynamic Headline
Introduction	What ABC Download Does
Purpose	
User Authorization	Obtaining User Authorization
Prerequisites	Software and Hardware You Need
	Knowledge You Need

(5) Revise the user-oriented outline as you write. You may find that modules, as conceived directly from a process-oriented outline, actually contain more than one idea and are thus too complicated to present in one or two pages. There is no practical limit to the number of modules you can have as long as headlines are descriptive and easily visible and you have a good index and table of contents.

(6) Try to use at least one visual (such as a screen facsimile or a diagram) for each module. Generally, the more visuals, the better.

Note in the Figure 5-1(b) example that the user-oriented outline contains numerous first- and second-level topics, but no third, fourth, or further broken-down topics.

By "modularizing" the material in this fashion, you take a big step toward making your manual more usable. However, even with this approach as a beginning, you may be tempted to make "submodules" and even "sub-submodules" because of the old habits instilled through years of working from static, process-oriented, hierarchically complex outlines. If you keep in mind the needs of the user, you will realize that even your "submodule" deserves the status of a full-fledged "module" when the user is anxiously looking for the information it contains.

For tutorial material--exercises using sample user tasks--modules can be used to break the overall task into smaller tasks, with the headline for each module describing specifically that particular part of the task.

For explanations of commands, consider making each command a separate module.

5.2 USE INDEX TABS

Index tabs are useful to separate major parts of a manual. For example, you might separate the types of modules as follows:

Index Tab	Modules
INSTALLATION	Hardware and Software You Need List of Files on Diskette Copying the Files to Your Hard Disk Configuring Software to Your Needs Installing Driver for Your Printer
GETTING STARTED	(All Tutorial Modules)
COMMANDS	(All Modules Explaining Commands)
ERROR MESSAGES	(All Modules on Error Recovery)
INDEX	Index to the Manual

If you have separate index tabs for each one- to three-page module, the tabs become more of a hindrance than a help.

The decision to use index tabs and how to use them must depend on the type and complexity of the application being described.

5.3 INCLUDE AN INDEX

Unless you can somehow automate the process, creating an index is a huge task. Nonetheless, users continue to demand indexes and feel that even superficial ones are better than none at all.

Ideally, an index has the following characteristics:

(1) Entries are given for all key words as they appear in the manual. Keep in mind that the greater the proportion of verb forms to nouns as index entries, the more user task oriented is the manual. For example, think in terms of "installing," rather than "installation."

(2) Cross-referenced entries are also given in the terminology of the user when it is likely to differ from the terminology used in the manual. For example, if the manual refers to the "log-on" procedure, you might

consider additional entries in the index for "sign-on," "accessing mainframe," and "connecting to mainframe," all referring the reader to the entries for "log-on."

(3) Entries are given at both major and minor levels, where appropriate. For example,

Installing
  Infogate **5**
  Lotus 123 **5**
  Printer for spreadsheet **10**
  RSR Download **6**, *7*
  SuperCalc3 **5**

(4) The page number of the primary place where the term is defined or explained is printed in boldface type. This is where the user is most likely to find the main point regarding the topic.

Some word-processing packages have built-in index generators. The best ones allow you to define two levels of entries, and designate the desired page number format. Writers and editors of user's guides are urged to consider using one of these word-processing packages or a separate index-generating package.

5.4 INCLUDE SUMMARIZING APPENDIXES

Appendixes are ideal places to put reference material of all kinds. Any details that have something in common and can be alphabetized or otherwise logically ordered in some way can be gathered into an appendix. In this way, technical details are easily referenced by the more advanced user, while staying out of the way of the novice.

In addition to providing a forum for greater detail, an appendix can be used to summarize information given in greater detail in the body of the manual. For example, where commands are explained in user-task-oriented modules in the body, they can be summarized and alphabetized in an appendix. After becoming familiar with a system, most users will remember the name of the command needed to perform a particular task, but may not remember such details as command parameters. They need a quick way to look up that command, without having to wade through a lot of detail they already know.

Examples of materials appropriate for appendixes are as follows:

(1) Glossary of terms and acronyms

(2) Detailed definitions of data elements (or data field names), with maximum field lengths, default values, possible values, codes, mapping of data element names to field names on printed reports, etc.

(3) Lists of field or data element names, with their definitions, maximum field length, default values, other possible values, etc.

(4) Samples of printed reports generated by the system

(5) Summaries of commands

(6) Lists of error messages and recovery procedures

5.5 INCLUDE A TABLE OF CONTENTS

Compared to an index, a table of contents is easy to prepare, and there is no excuse for leaving it out. However, a table of contents is only as good as the overall organization and headlining of the entire document. If the modular headlines are concise, descriptive, and written with the user's tasks in mind, a table of contents can be very useful.

The table of contents should have right-flushed page numbers with leader dots to assist the eye.

5.6 CONSIDER A QUICK REFERENCE GUIDE

Many users appreciate a quick reference card or booklet they can separate from the rest of the user's manual and leave in a handy place near their computer. A quick reference card helps the user already familiar with the system to recall exact commands, parameters, sequences, codes, or other information.

Quick reference guides may take a number of different physical forms.

(1) If carefully organized and laid out, folded reference cards work well. These would include, for example, an 8-1/2- x 11-inch card folded into thirds, with information printed on both sides, or an even wider page, fan folded several times. However, some users find these confusing if the layout is weak.

(2) A flat card with well-organized and labeled information printed on both sides is preferred by some users over the folded format.

(3) For a complex system, a small booklet, folded and stapled in the center, may be required in order to include all the vital information.

(4) A function key template or other type of keyboard overlay is often an indispensable form of quick reference documentation.

Whatever the physical form, information selected for the quick reference guide should be limited to procedural details not easily remembered by infrequent users. Information should be structured into numbered steps, tables, or other concise formats for easy reference. Headings for the procedures, tables, lists, diagrams, and other aids should be prominent and descriptive. The diminutive size of the reference card should not compromise the legibility of the type.

SECTION 6

DESIGN THE PACKAGE TO FIT

6.1 FAVOR SMALL PACKAGES FOR LARGE DISTRIBUTIONS

Microcomputer users have quickly become accustomed to the small user document format used for many popular commercial software packages. From the user's point of view, a document in this small format (pages 5-1/2 inches wide by 8-1/2 inches high) has several advantages over the standard 8-1/2- x 11-inch format:

(1) It is handier to use at the computer.

(2) It has enough body to stand up on shelves without needing a hard binder, yet takes little space.

(3) With less information on each page, information is easier to find and pages are less intimidating.

Items 1 and 2 may seem frivolous in view of the additional production and reproduction constraints. (For example, this size document cannot be easily published using a standard photocopying machine; usually it must be printed on a press and then the sheets cut to size.) However, Item 3 is significant and may be worth the sacrifice of some ease of production, especially for documents that will have a large distribution and will have to be printed on a press anyway.

For documents to be distributed to 50 or more users, the smaller format is recommended.

Section 7 elaborates on page design to make the most of the 5-1/2- x 8-1/2-inch page.

6.2 BIND FOR USER'S CONVENIENCE

The choice of binding method may depend on the distribution of the document, its thickness, whether it will be updated with change pages (and how often), whether the package includes diskettes, the document production budget, and other considerations. Table 6-1 compares five different binding methods. All these methods allow for custom-printed covers on heavy, colored stock.

Table 6-1. Comparison of Binding Techniques

Method	Advantages	Disadvantages
Loose-leaf ring binder. Includes: **Linen-covered binder with custom silk screening;** **vinyl with custom printing;** **lip overlay with printed insert**	Allows easy insertion of updates. Allows easy, clear identification of spine. Lays flat when open. Can also hold and protect diskettes in plastic sleeve.	Takes up extra shelf space. Cannot be folded under for convenience at computer.
	* * * * *	
Plastic comb binding	Efficiently uses shelf space. Lays flat when open and can be folded over. Spine can be identified with a stick-on label or spine imprint.	Special "Combo" device needed to open binding and insert updates. Many users do not have access to device.
	* * * * *	
Acco-fastened (3-hole punched)	Efficiently uses shelf space. Updates can be inserted fairly easily. User can insert into own loose-leaf binder, if desired.	Usually will not lay flat when open and cannot be folded under. Difficult to identify spine. Fasteners may be troublesome, possibly causing small cuts:

Table 6-1. Comparison of Binding Techniqnes (Contd)

Method	Advantages	Disadvantages
Saddle stitching	Efficiently uses shelf space. Lays flat when open and can be folded under.	Cannot be used for documents over 96 pages (copied two-sided). Does not allow insertion of updates. Does not accommodate use of index tabs. Does not hold and protect diskettes.
	* * * * *	
Perfect binding (wrap-around cover, glued pages, similar to a paper-bound book)	Attractive, professional looking. Efficiently uses shelf space. Title can be printed on spine.	Does not lay flat when open and cannot be folded under. Updates cannot be inserted.

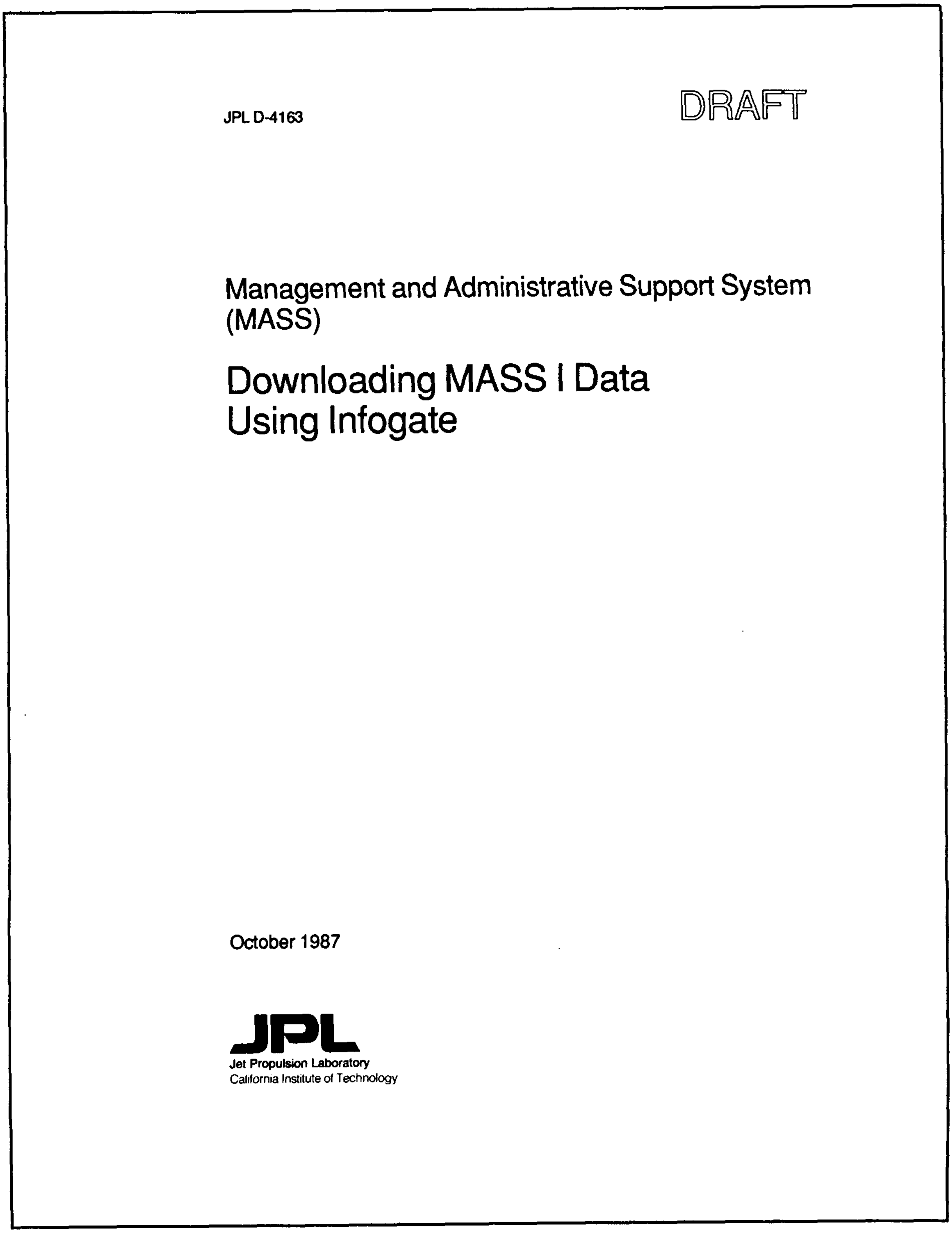
Your choice of binding method depends at least partly on cost considerations. If the document will be a selling tool for the software product, a well-designed, professional-looking cover is more important than it would be for a product developed for internal use in an organization. However, in the process of designing a cover for form, do not forget the functionality of the document.

6.3 USE STANDARD COVER AND TITLE PAGE FORMAT

Organizations developing user's guides for internal use or commercial software companies desiring to create a trademark image should consider developing a standard format for all their user's guides. Figure 6-1 shows a sample cover for a software user's guide for a system developed at the Jet Propulsion Laboratory (JPL). A wide variety of software is developed for internal use at JPL. Therefore, the size format of the documentation is not dictated, but rather left to be decided case by case.

The title page can be identical to the cover, except for the addition of any signature blocks for the originator and approvers, as in the JPL example in Figure 6-2.

Figure 6-3 shows a format that can be used for identifying the spine of the document if you are using one of the binding methods allowing such identification (i.e., loose-leaf binders or perfect binding).



JPL D-4163

DRAFT

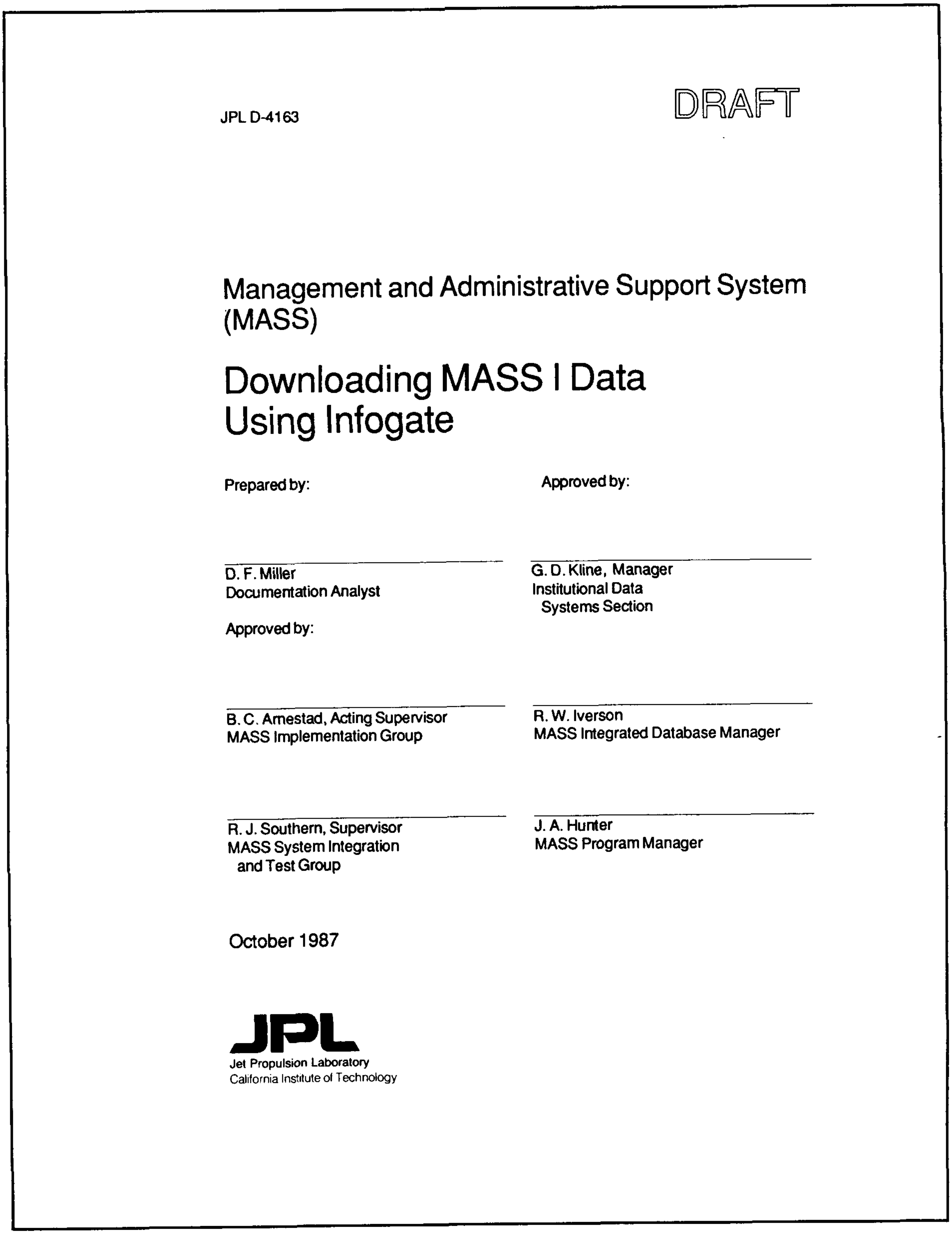
Management and Administrative Support System
(MASS)

Downloading MASS I Data
Using Infogate

October 1987

JPL
Jet Propulsion Laboratory
California Institute of Technology

Figure 6-1. Sample Cover Page for a User's Guide



JPL D-4163

DRAFT

Management and Administrative Support System (MASS)

Downloading MASS I Data Using Infogate

Prepared by:

D. F. Miller
Documentation Analyst

Approved by:

G. D. Kline, Manager
Institutional Data
Systems Section

Approved by:

B. C. Arnestad, Acting Supervisor
MASS Implementation Group

R. W. Iverson
MASS Integrated Database Manager

R. J. Southern, Supervisor
MASS System Integration
and Test Group

J. A. Hunter
MASS Program Manager

October 1987

JPL
Jet Propulsion Laboratory
California Institute of Technology

Figure 6-2. Sample Title/Approval Page for a User's Guide

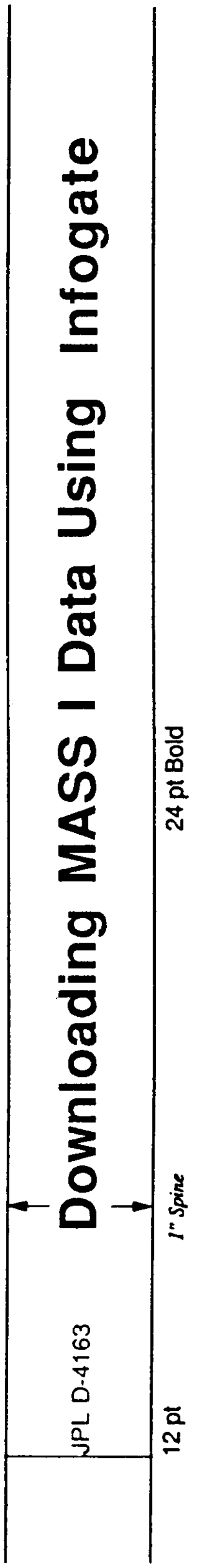


Figure 6-3. Sample Format for Spine Identification

SECTION 7

FORMAT FOR QUICK COMPREHENSION

This section elaborates on the following basic principle of technical writing, derived from extensive studies of how we learn and remember (a field called cognitive psychology):

> **Readers understand written information more easily and quickly when they are given visual clues to meaning, organization, and conceptual relationships.**

We are not talking about visuals, per se, just yet. This principle refers only to the visual shapes of groupings of words (phrases, sentences, and paragraphs) and their relationship to the white space remaining on the page.

7.1 LIMIT AMOUNT OF INFORMATION ON A PAGE

Naturally, to create a visual effect with words, you must have white space on the page for background or contrast. If words on the page were notes in a symphony, the white spaces would be the rests, the time between movements, the silences that give the music its form in time.

The first step in using this principle is to present a limited amount of information on each page. Only then will you have the white space you need on which to manipulate the shapes of your writing. If you follow the guidelines in Section 5.1 and divide your material into bite-sized modules, the task of designing your pages will be simpler.

Use of the small format (5-1/2- x 8-1/2-inch) pages described in Section 6.1 forces you to put only a small amount of information on each page. Many users become intimidated or impatient with full-sized pages carrying wall-to-wall type, even if the writing itself is clear and concise. Users do not mind turning more pages if the information they seek shouts right out at them, because it is surrounded by silence.

7.2 USE DEVICES TO CREATE SHAPE

Several simple ideas can be adapted to any kind of technical or expository writing to greatly improve the reader's ease of reading and comprehension:

(1) **Use a serif typeface of at least 10 points.** Serif typefaces are generally considered easier to read than are sans serif typefaces. The fine strokes at the bottom of the letters form a kind of imaginary line on which the type rests, thus aiding the reader's eye across the line of words. The size is important, because readers may be continually focusing from screen to document and back again. The document may be placed on the table beside the computer at a greater than normal reading distance. Sans serif type, however, is acceptable for large headings.

(2) **Use upper- and lower-case letters rather all upper case.** Because of the ascenders and descenders, lower-case letters have more distinctive shapes than do upper-case letters. Use all upper-case only for headings, names of keys or commands, or emphasis of no more than three or four words in a row within the text.

(3) **Use a ragged rather than a justified right margin.** A ragged right margin has a meaningful shape that a right-justified margin does not have. When lines vary in length, their differences help the reader's eye find the beginning of the next line. Another disadvantage of right-justification is that often large spaces are left between words, creating "rivers" meandering down the page. These spaces are meaningless and distracting.

(4) **Treat items in a series as list items.** Rather than allowing groups of words describing items in a series to remain buried in paragraph form, set each one apart on a line of its own. Figure 7-1 is a before-and-after example. Note also the use of line spacing to indicate groupings and separations in the microstructure of the paragraph. Or, these groupings of words may each be several lines long and treated visually as indented paragraphs, as shown in Figure 7-2.

(5) **Sequentially number items in a series.** Although not strictly visual, this device is related to Item 4. Numbering list items often helps readers to learn and remember them. The fact that there are four different types of screens in an application may not be significant in itself, but it may help in recalling what those types are if the reader can first recall that there are four of them. The order of the items in the list is usually arbitrary. See Figure 7-3 for a before-and-after example.

BEFORE USE OF LIST ITEMS:

The Contract Management System at JPL contains sponsor (or customer) contract management information. It is linked to the Financial Management Chart of Accounts, which contains the valid charts of job numbers used to control all work being done internally at JPL. The Contracts Management System contains information on sponsors, sponsor contracts, sponsor account numbers, funds, and funding agreements, while the Financial Management Chart of Account contains detailed sponsor accounts (JPL-determined) and job numbers.

AFTER USE OF LIST ITEMS:

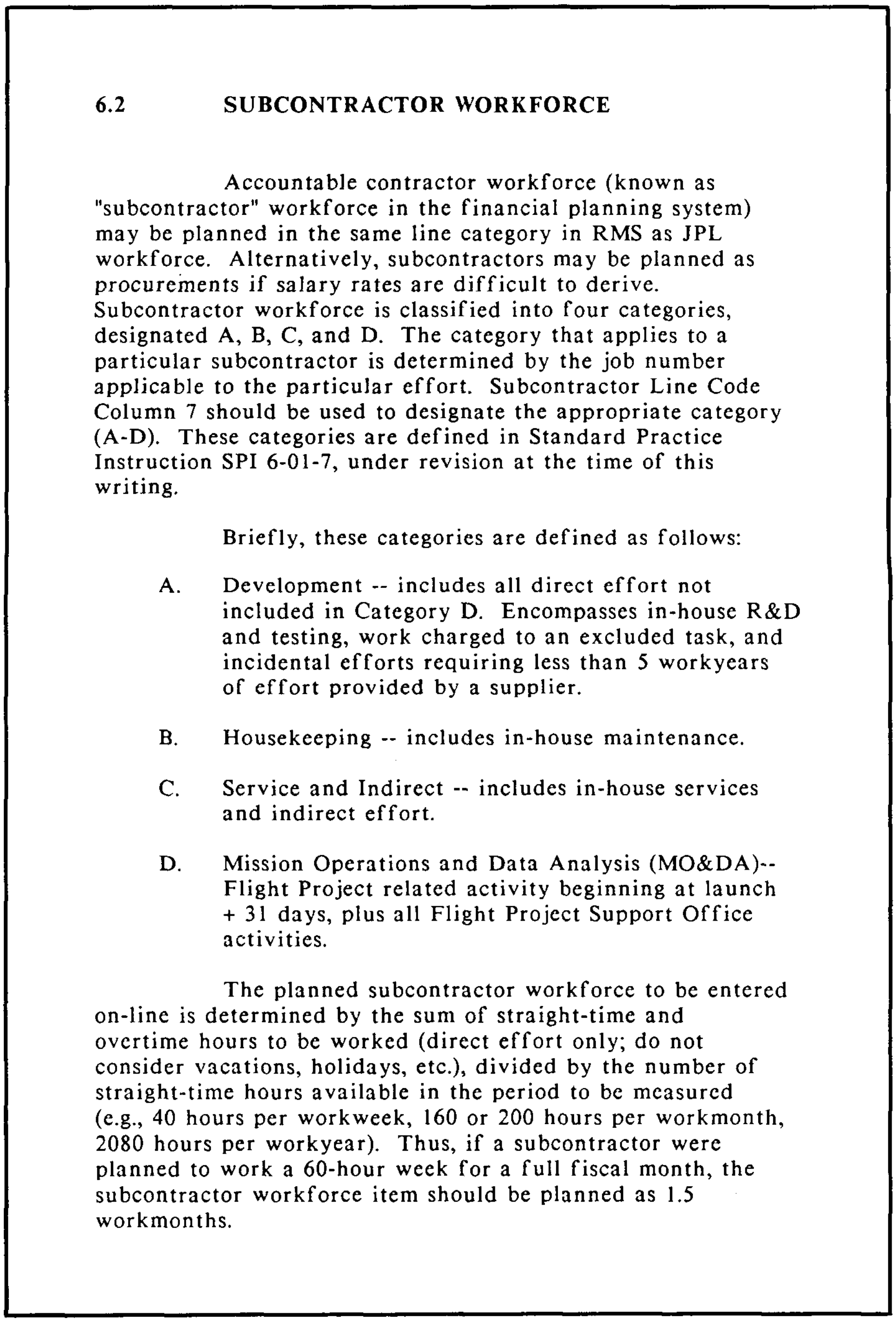
The Contract Management System at JPL contains sponsor (or customer) contract management information. It is linked to the Financial Management Chart of Accounts, which contains the valid charts of job numbers used to control all work being done internally at JPL. The Contracts Management System contains information on

- sponsors
- sponsor contracts
- sponsor account numbers
- funds
- funding agreements

while the Financial Management Chart of Account contains

- detailed sponsor accounts (JPL-determined)
- job numbers

Figure 7-1. Example of Visual Emphasis of a Series of Short Items



6.2 SUBCONTRACTOR WORKFORCE

Accountable contractor workforce (known as "subcontractor" workforce in the financial planning system) may be planned in the same line category in RMS as JPL workforce. Alternatively, subcontractors may be planned as procurements if salary rates are difficult to derive. Subcontractor workforce is classified into four categories, designated A, B, C, and D. The category that applies to a particular subcontractor is determined by the job number applicable to the particular effort. Subcontractor Line Code Column 7 should be used to designate the appropriate category (A-D). These categories are defined in Standard Practice Instruction SPI 6-01-7, under revision at the time of this writing.

Briefly, these categories are defined as follows:

A. Development -- includes all direct effort not included in Category D. Encompasses in-house R&D and testing, work charged to an excluded task, and incidental efforts requiring less than 5 workyears of effort provided by a supplier.

B. Housekeeping -- includes in-house maintenance.

C. Service and Indirect -- includes in-house services and indirect effort.

D. Mission Operations and Data Analysis (MO&DA)-- Flight Project related activity beginning at launch + 31 days, plus all Flight Project Support Office activities.

The planned subcontractor workforce to be entered on-line is determined by the sum of straight-time and overtime hours to be worked (direct effort only; do not consider vacations, holidays, etc.), divided by the number of straight-time hours available in the period to be measured (e.g., 40 hours per workweek, 160 or 200 hours per workmonth, 2080 hours per workyear). Thus, if a subcontractor were planned to work a 60-hour week for a full fiscal month, the subcontractor workforce item should be planned as 1.5 workmonths.

Figure 7-2. Example of Visual Emphasis of a Series of Longer Items

BEFORE SEQUENTIAL NUMBERING OF ITEMS:

The JPL Property System uses the following types of screens:

- Menu Screens
- Browse Screens
- Display Screens
- Data Maintenance Screens

AFTER SEQUENTIAL NUMBERING OF ITEMS:

The JPL Property System uses four types of screens:

1) Menu Screens
2) Browse Screens
3) Display Screens
4) Data Maintenance Screens

Figure 7-3. Example of Sequential Numbering of Items as a Learning and Memory Aid. (Order of items is usually not important.)

(6) **Give procedural information in sequentially numbered steps.** In order to fully describe how to perform a task using a computer system, you generally have to include some preliminary background information, give a few warnings or exceptions, and otherwise interrupt yourself to make sure the procedural steps are understood in the proper context. It is important to separate these narrative diversions from the actual steps the user is to follow in interacting with the computer. Figure 7-4 shows a before-and-after example of procedural information presented in sequentially numbered steps. In the "after" example, it is obvious where the procedure begins and ends.

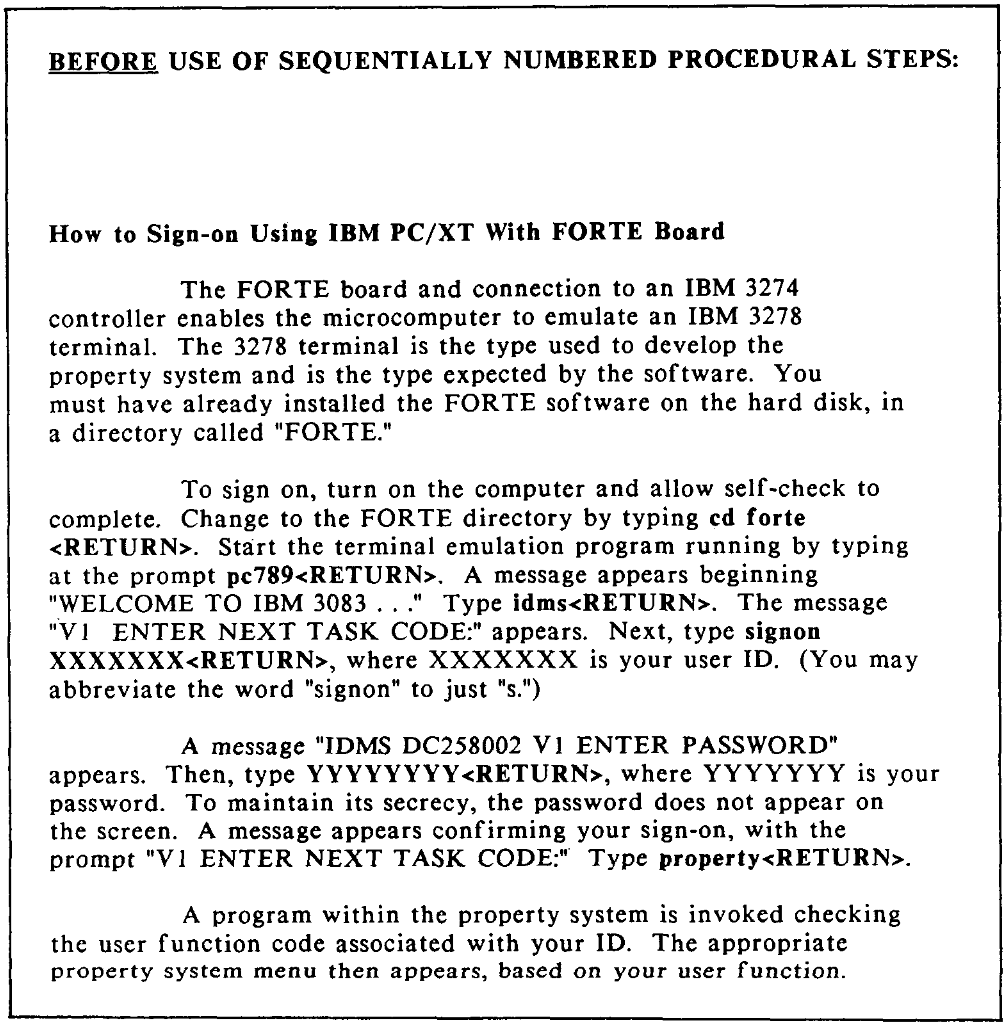
(7) **Write tutorials in a two-columned, play-script format.** Tutorials lead novice users through sample user tasks one small step at a time, instructing them on exactly what to do and showing them exactly what to expect from the computer at each step. Tutorials contain little in the way of theoretical information or explanation, but merely give users some hands-on experience with the system to get them started. The two-columned, play-script format, shown in Figure 7-5, clearly numbers the steps, and places the user input and computer response in visually meaningful proximity. Screen facsimiles should be used generously and can be laid out across the two columns.

(8) **Make headings highly visible.** Ways to do this include

(a) Surround headings with white space. Figure 7-6 shows a couple of different ways to do this. For example, you can use a two-columned format, where the left third of the page is reserved for headings and white space only. Or, you can use extra line spacing before and after the heading.

(b) Put headings in larger type, all capitals, or boldface type.

(9) **Whenever feasible, put information into a table.** Wherever relationships between two or more variables are discussed, a table is almost always easier and quicker to understand than are narrative paragraphs. Figure 7-7 shows an example of general procedural information that was originally written in paragraph form (Figure 7-7(a)) and translated into a table (Figure 7-7(b)). The two "variables" in this example are (1) to accomplish X, (2) do Y. Some other types of information best presented in tables are



BEFORE USE OF SEQUENTIALLY NUMBERED PROCEDURAL STEPS:

How to Sign-on Using IBM PC/XT With FORTE Board

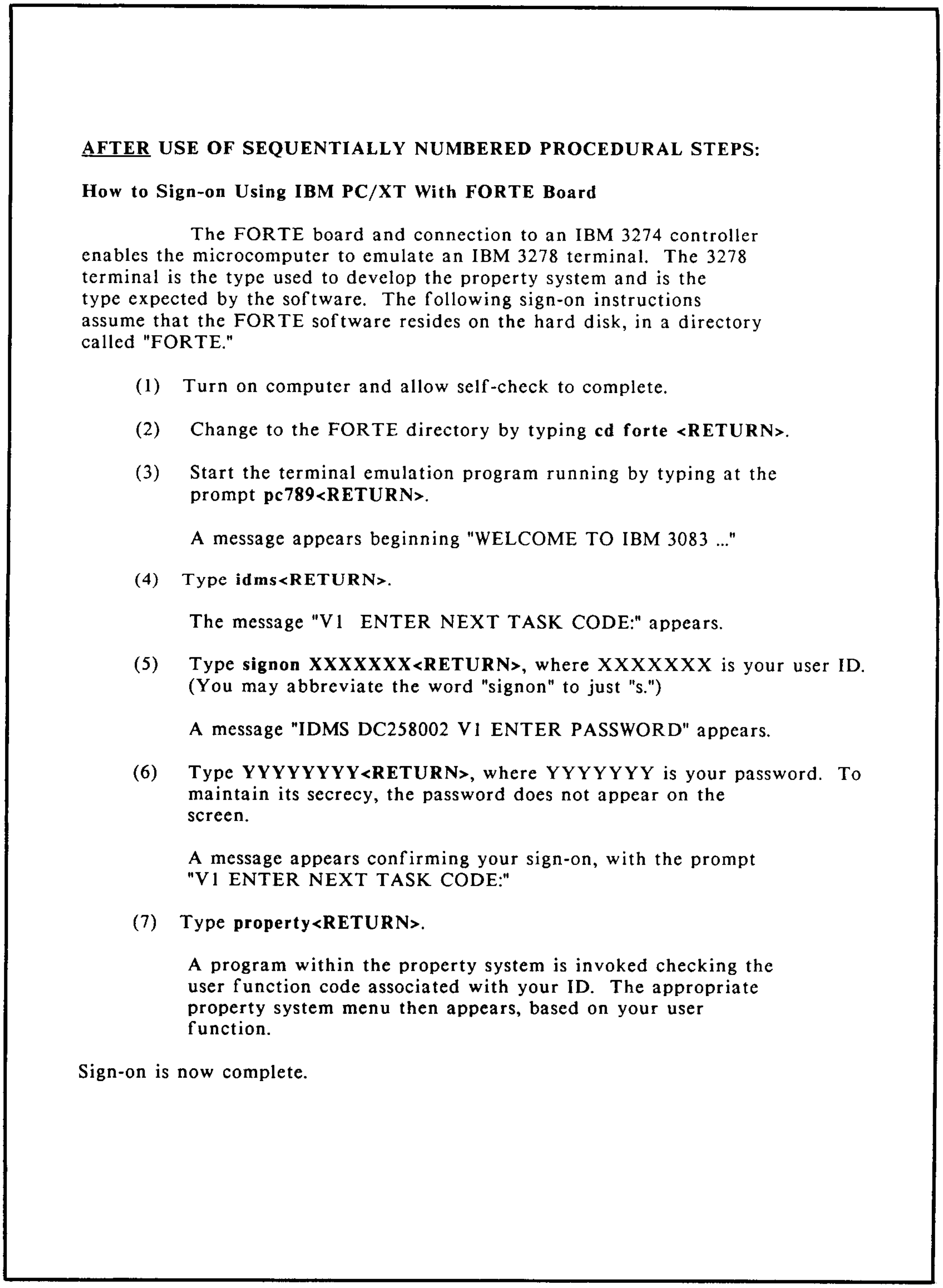
The FORTE board and connection to an IBM 3274 controller enables the microcomputer to emulate an IBM 3278 terminal. The 3278 terminal is the type used to develop the property system and is the type expected by the software. You must have already installed the FORTE software on the hard disk, in a directory called "FORTE."

To sign on, turn on the computer and allow self-check to complete. Change to the FORTE directory by typing **cd forte** **<RETURN>**. Start the terminal emulation program running by typing at the prompt **pc789<RETURN>**. A message appears beginning "WELCOME TO IBM 3083 . . ." Type **idms<RETURN>**. The message "V1 ENTER NEXT TASK CODE:" appears. Next, type **signon** **XXXXXXX<RETURN>**, where XXXXXXX is your user ID. (You may abbreviate the word "signon" to just "s.")

A message "IDMS DC258002 V1 ENTER PASSWORD" appears. Then, type **YYYYYYYY<RETURN>**, where YYYYYYY is your password. To maintain its secrecy, the password does not appear on the screen. A message appears confirming your sign-on, with the prompt "V1 ENTER NEXT TASK CODE:" Type **property<RETURN>**.

A program within the property system is invoked checking the user function code associated with your ID. The appropriate property system menu then appears, based on your user function.

Figure 7-4(a). Example of Procedural Information Buried in Narrative Paragraph Form



AFTER USE OF SEQUENTIALLY NUMBERED PROCEDURAL STEPS:

How to Sign-on Using IBM PC/XT With FORTE Board

The FORTE board and connection to an IBM 3274 controller enables the microcomputer to emulate an IBM 3278 terminal. The 3278 terminal is the type used to develop the property system and is the type expected by the software. The following sign-on instructions assume that the FORTE software resides on the hard disk, in a directory called "FORTE."

(1) Turn on computer and allow self-check to complete.

(2) Change to the FORTE directory by typing **cd forte <RETURN>**.

(3) Start the terminal emulation program running by typing at the prompt **pc789<RETURN>**.

A message appears beginning "WELCOME TO IBM 3083 ..."

(4) Type **idms<RETURN>**.

The message "V1 ENTER NEXT TASK CODE:" appears.

(5) Type **signon XXXXXXX<RETURN>**, where XXXXXXX is your user ID. (You may abbreviate the word "signon" to just "s.")

A message "IDMS DC258002 V1 ENTER PASSWORD" appears.

(6) Type **YYYYYYYY<RETURN>**, where YYYYYYY is your password. To maintain its secrecy, the password does not appear on the screen.

A message appears confirming your sign-on, with the prompt "V1 ENTER NEXT TASK CODE:"

(7) Type **property<RETURN>**.

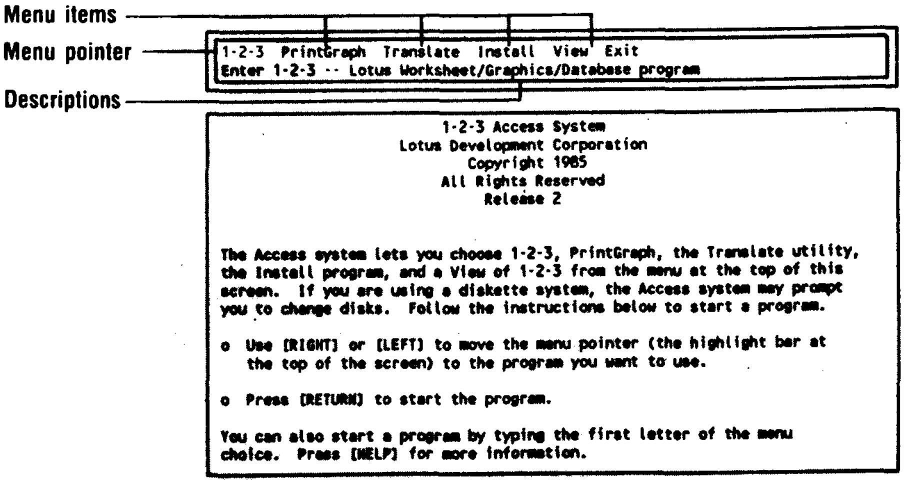
A program within the property system is invoked checking the user function code associated with your ID. The appropriate property system menu then appears, based on your user function.

Sign-on is now complete.

Figure 7-4(b). Example of Procedural Information Presented in Sequentially Numbered Steps

STARTING 1-2-3

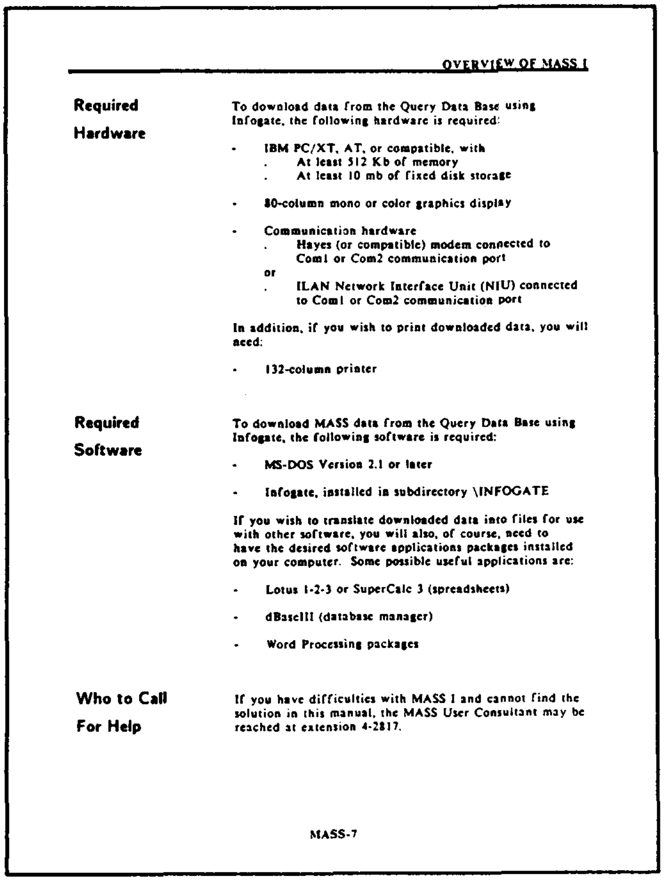
	Your Input	Computer Response
1.	At the operating system prompt, type **123<RETURN>**	The Access System provides a menu from which you can select the various programs. A menu pointer highlights **1-2-3**. Below this choice is the description.



```
1-2-3  PrintGraph  Translate  Install  View  Exit
Enter 1-2-3 -- Lotus Worksheet/Graphics/Database program

                      1-2-3 Access System
                 Lotus Development Corporation
                        Copyright 1985
                      All Rights Reserved
                           Release 2

The Access system lets you choose 1-2-3, PrintGraph, the Translate utility,
the Install program, and a View of 1-2-3 from the menu at the top of this
screen.  If you are using a diskette system, the Access system may prompt
you to change disks.  Follow the instructions below to start a program.

o  Use [RIGHT] or [LEFT] to move the menu pointer (the highlight bar at
   the top of the screen) to the program you want to use.

o  Press [RETURN] to start the program.

You can also start a program by typing the first letter of the menu
choice.  Press [HELP] for more information.
```

2.	Press **<RIGHT ARROW>**	The menu pointer moves to **PrintGraph**, and the description appears below it.
3.	Press **<RIGHT ARROW>** five more times.	Each menu item is highlighted in turn, wrapping back to **1-2-3.**
4.	Highlight **1-2-3** and press **<RETURN>**.	The disk drive light turns on, and soon the screen displays a copyright notice.
5.	Press **<RETURN>**	1-2-3 begins.

Figure 7-5. Example of a Two-Columned, Play-Script Format Used for Tutorials



OVERVIEW OF MASS I

Required Hardware

To download data from the Query Data Base using Infogate, the following hardware is required:

- IBM PC/XT, AT, or compatible, with
  - At least 512 Kb of memory
  - At least 10 mb of fixed disk storage
- 80-column mono or color graphics display
- Communication hardware
  - Hayes (or compatible) modem connected to Com1 or Com2 communication port

  or
  - ILAN Network Interface Unit (NIU) connected to Com1 or Com2 communication port

In addition, if you wish to print downloaded data, you will aeed:

- 132-column printer

Required Software

To download MASS data from the Query Data Base using Infogate, the following software is required:

- MS-DOS Version 2.1 or later
- Infogate, installed in subdirectory \INFOGATE

If you wish to translate downloaded data into files for use with other software, you will also, of course, need to have the desired software applications packages installed on your computer. Some possible useful applications are:

- Lotus 1-2-3 or SuperCalc 3 (spreadsheets)
- dBaseIII (database manager)
- Word Processing packages

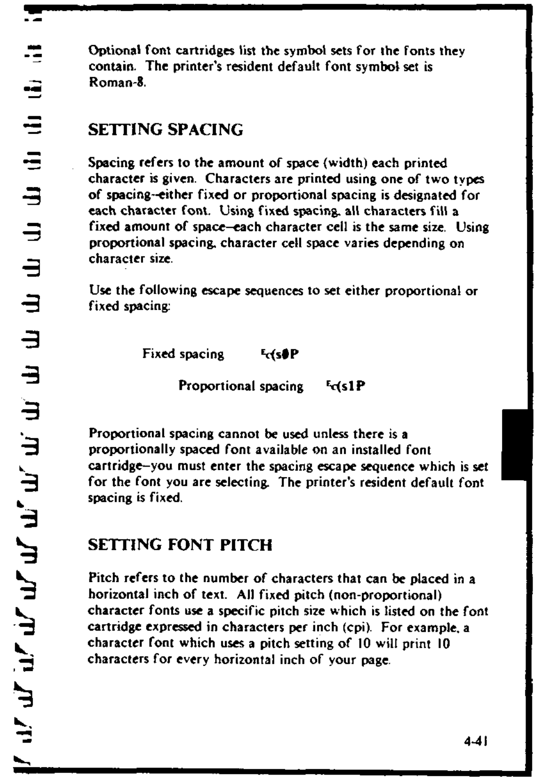
Who to Call For Help

If you have difficulties with MASS I and cannot find the solution in this manual, the MASS User Consultant may be reached at extension 4-2817.

MASS-7

(a)

(b)



Optional font cartridges list the symbol sets for the fonts they contain. The printer's resident default font symbol set is Roman-8.

SETTING SPACING

Spacing refers to the amount of space (width) each printed character is given. Characters are printed using one of two types of spacing--either fixed or proportional spacing is designated for each character font. Using fixed spacing, all characters fill a fixed amount of space–each character cell is the same size. Using proportional spacing, character cell space varies depending on character size.

Use the following escape sequences to set either proportional or fixed spacing:

Fixed spacing $^{E}{}_{C}$(s0P

Proportional spacing $^{E}{}_{C}$(s1P

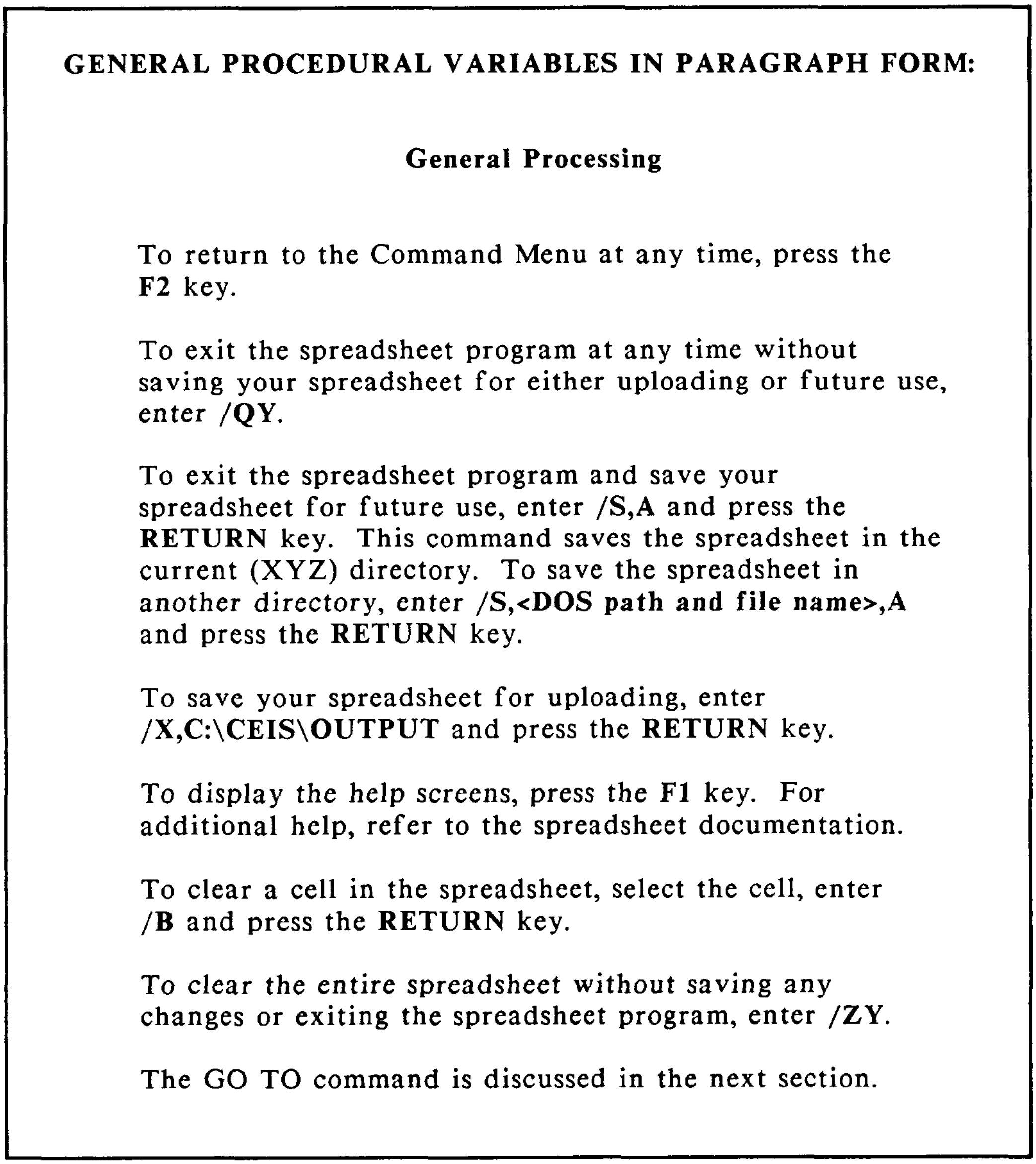
Proportional spacing cannot be used unless there is a proportionally spaced font available on an installed font cartridge–you must enter the spacing escape sequence which is set for the font you are selecting. The printer's resident default font spacing is fixed.

SETTING FONT PITCH

Pitch refers to the number of characters that can be placed in a horizontal inch of text. All fixed pitch (non-proportional) character fonts use a specific pitch size which is listed on the font cartridge expressed in characters per inch (cpi). For example, a character font which uses a pitch setting of 10 will print 10 characters for every horizontal inch of your page.

4-41

Figure 7-6. Examples of Ways To Make Headings Visible: (a) uses a two-columned format; (b) (reprinted by permission of Hewlett-Packard Company) has bold headings with ample line spacing.



GENERAL PROCEDURAL VARIABLES IN PARAGRAPH FORM:

General Processing

To return to the Command Menu at any time, press the **F2** key.

To exit the spreadsheet program at any time without saving your spreadsheet for either uploading or future use, enter /**QY**.

To exit the spreadsheet program and save your spreadsheet for future use, enter **/S,A** and press the **RETURN** key. This command saves the spreadsheet in the current (XYZ) directory. To save the spreadsheet in another directory, enter **/S,<DOS path and file name>,A** and press the **RETURN** key.

To save your spreadsheet for uploading, enter **/X,C:\CEIS\OUTPUT** and press the **RETURN** key.

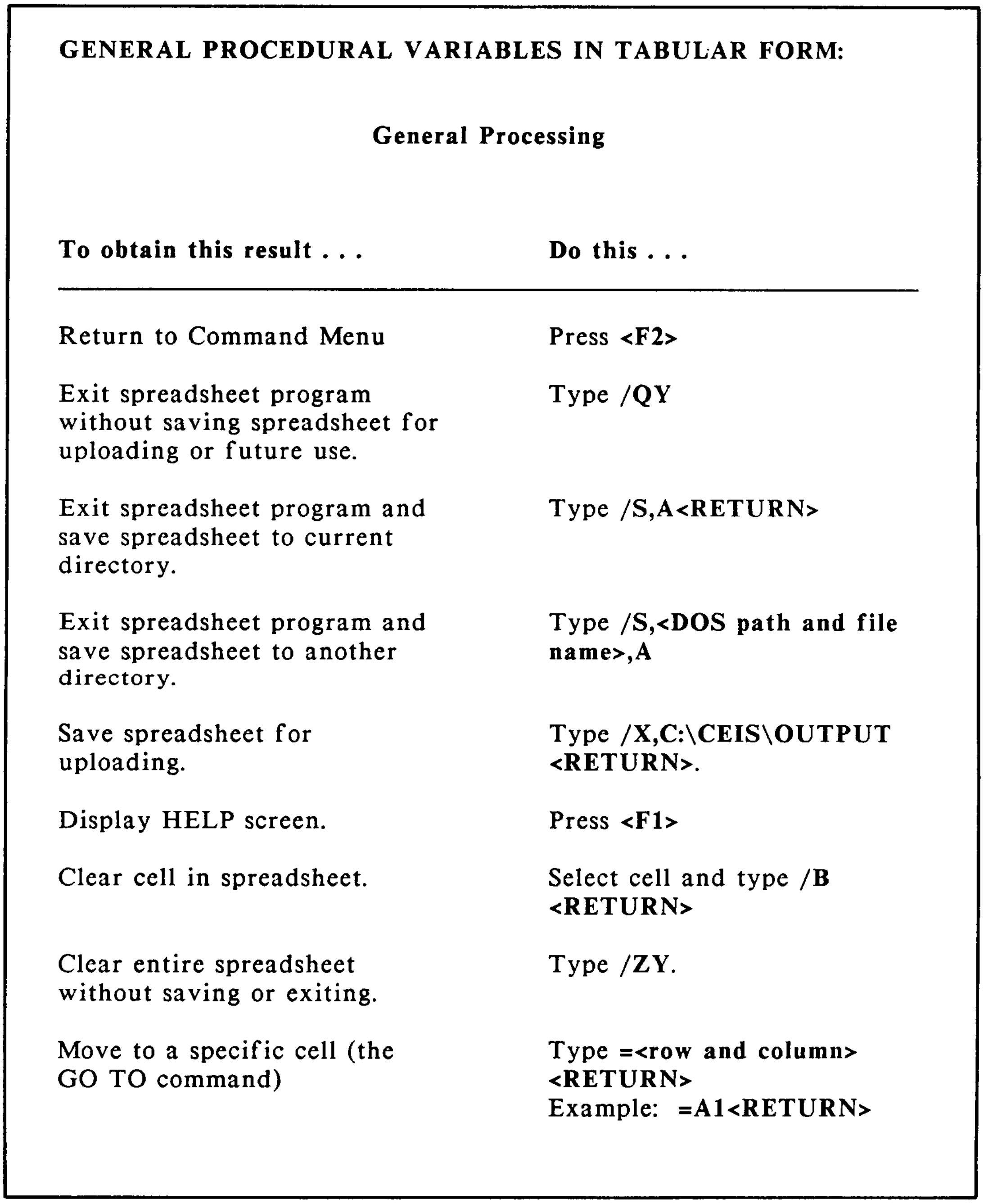
To display the help screens, press the **F1** key. For additional help, refer to the spreadsheet documentation.

To clear a cell in the spreadsheet, select the cell, enter /**B** and press the **RETURN** key.

To clear the entire spreadsheet without saving any changes or exiting the spreadsheet program, enter /**ZY**.

The GO TO command is discussed in the next section.

Figure 7-7(a). Example Showing Variables Buried in Narrative Paragraph Form



GENERAL PROCEDURAL VARIABLES IN TABULAR FORM:

General Processing

To obtain this result . . .	**Do this . . .**
Return to Command Menu	Press **<F2>**
Exit spreadsheet program without saving spreadsheet for uploading or future use.	Type **/QY**
Exit spreadsheet program and save spreadsheet to current directory.	Type **/S,A<RETURN>**
Exit spreadsheet program and save spreadsheet to another directory.	Type **/S,<DOS path and file name>,A**
Save spreadsheet for uploading.	Type **/X,C:\CEIS\OUTPUT <RETURN>**.
Display HELP screen.	Press **<F1>**
Clear cell in spreadsheet.	Select cell and type **/B <RETURN>**
Clear entire spreadsheet without saving or exiting.	Type **/ZY**.
Move to a specific cell (the GO TO command)	Type **=<row and column> <RETURN>** Example: **=A1<RETURN>**

Figure 7-7(b). Example Showing Variables Presented in Tabular Form

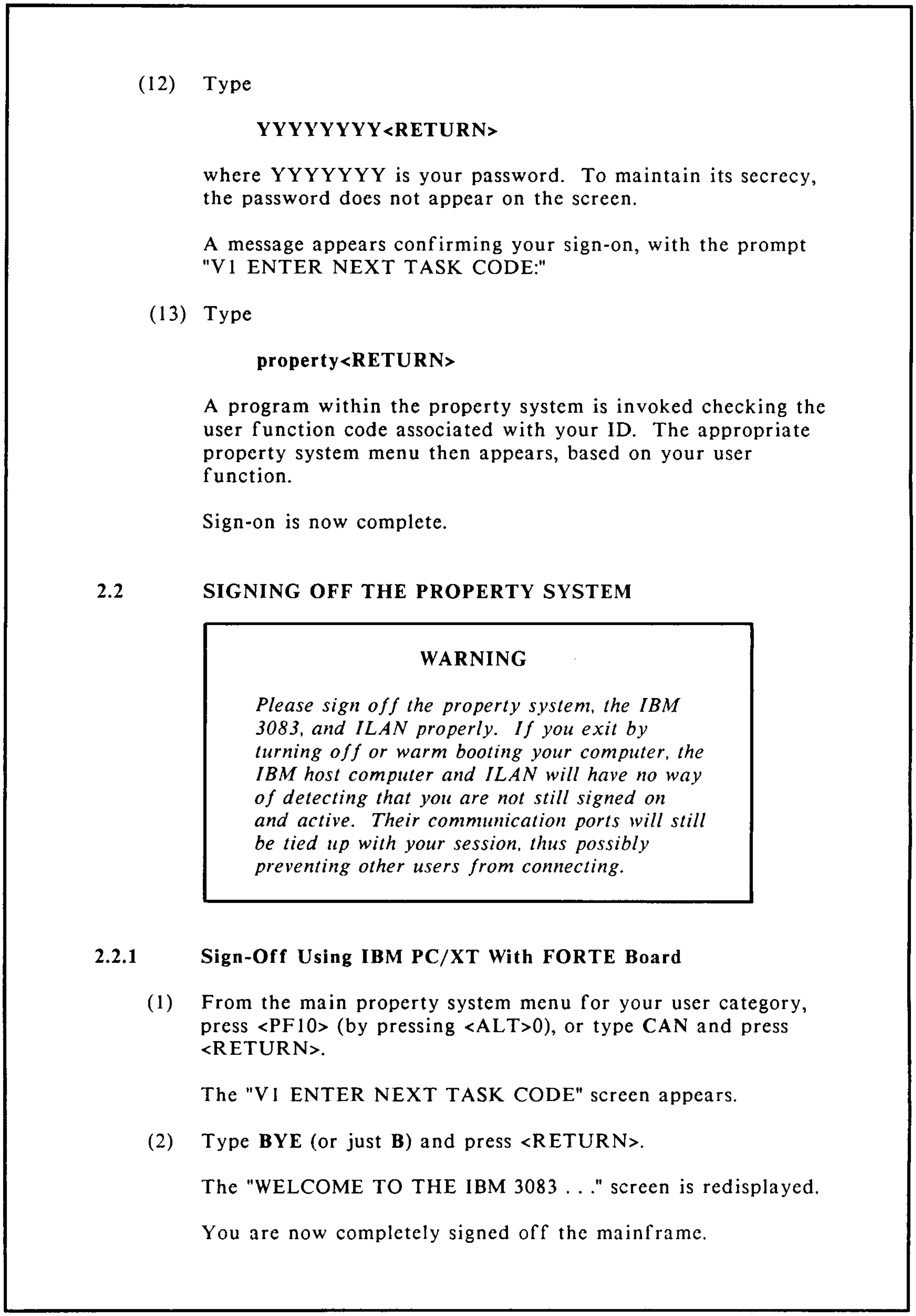
(a) Definitions of data fields on the screen. You may have columns for each of several parameters, such as name of field, maximum field length, whether required or optional data, default value, and definition.

(b) Descriptions of fields on a form. You may have columns for field number, field name, and definition.

(c) Function key names with associated commands and a description of the operation performed.

(10) **Warnings or information on special exceptions should be set in italic or boldface type and indented from both margins.** These devices are standard practice in technical instruction and specification manuals. You may also graphically box this information for further emphasis. Figure 7-8 is an example of a page containing special warning information.

Incidentally, if many warning or special exception notices are required, the software (or hardware) needs work. As a user advocate on the development team, you might gently prod the appropriate software engineer.



(12) Type

YYYYYYYY<RETURN>

where YYYYYYY is your password. To maintain its secrecy, the password does not appear on the screen.

A message appears confirming your sign-on, with the prompt "V1 ENTER NEXT TASK CODE:"

(13) Type

property<RETURN>

A program within the property system is invoked checking the user function code associated with your ID. The appropriate property system menu then appears, based on your user function.

Sign-on is now complete.

2.2 SIGNING OFF THE PROPERTY SYSTEM

WARNING

Please sign off the property system, the IBM 3083, and ILAN properly. If you exit by turning off or warm booting your computer, the IBM host computer and ILAN will have no way of detecting that you are not still signed on and active. Their communication ports will still be tied up with your session, thus possibly preventing other users from connecting.

2.2.1 Sign-Off Using IBM PC/XT With FORTE Board

(1) From the main property system menu for your user category, press <PF10> (by pressing <ALT>0), or type CAN and press <RETURN>.

The "V1 ENTER NEXT TASK CODE" screen appears.

(2) Type **BYE** (or just **B**) and press <RETURN>.

The "WELCOME TO THE IBM 3083 . . ." screen is redisplayed.

You are now completely signed off the mainframe.

Figure 7-8. Example Showing Indentation of Warnings or Special Exceptions

SECTION 8

USE VISUALS AND EXAMPLES LIBERALLY

8.1 USE SCREEN FACSIMILES

Every procedure and task you are documenting in a software system should be illustrated by an appropriate screen facsimile in the user's guide. Your screen facsimiles should show the entire process from the user's view. (You do not need to show every help screen, however.) The following guidelines summarize the preferences of most users:

(1) For database systems, screens should show sample user data (rather than just a blank screen) to assist the user in understanding all the data fields on the screen.

(2) Screens should be inserted into the document at the point in the text where they are applicable. They should show as nearly as possible how each screen will appear at precisely the step being discussed.

(3) Screen facsimiles may be numbered as figures, or not. Where many screens are presented, the screens become almost merged with the text in the flow of information. To number them as figures actually becomes a distraction.

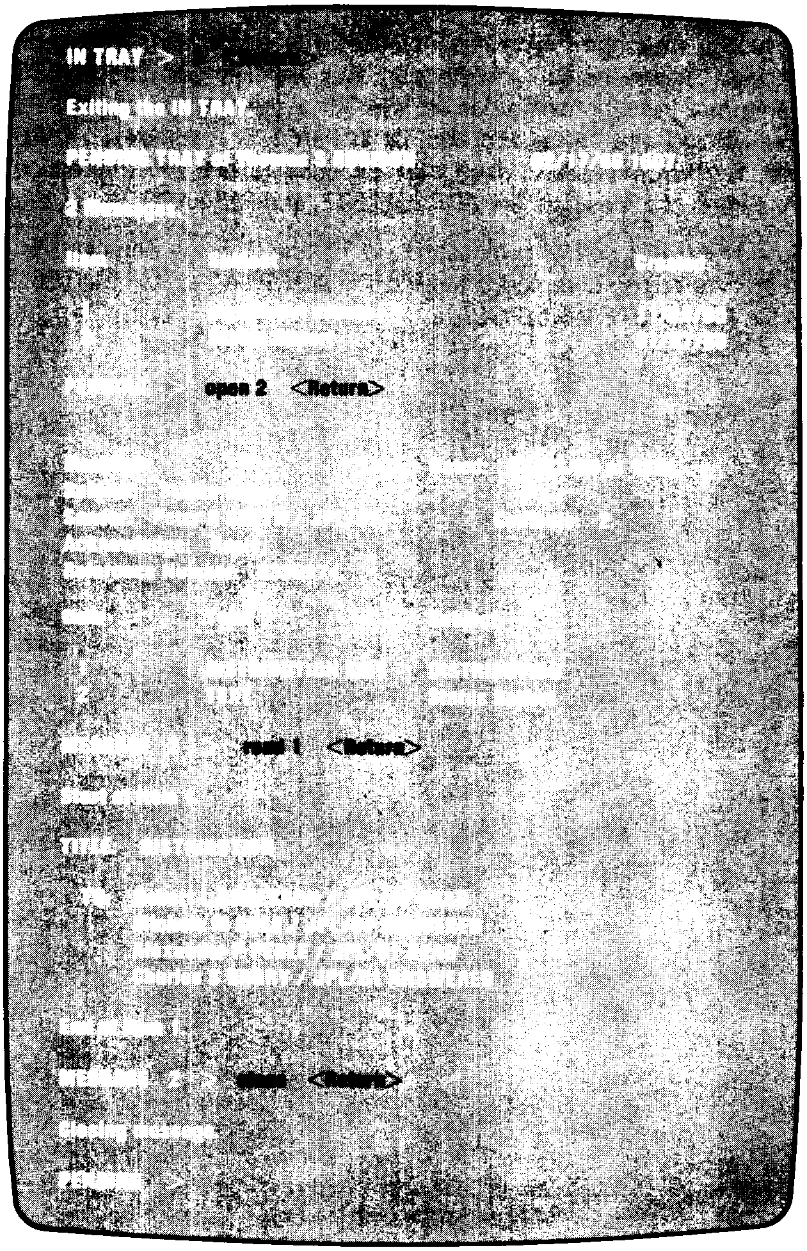
(4) Screens should always be fully captioned underneath. Readers often look at only the pictures and read the captions. Much information can be conveyed by highly descriptive, cogent, terse captions. For users who do read the text, the captions reinforce important points.

(5) For screens that use reverse video to highlight certain portions (such as spreadsheet row and column labels, options for selection, or status lines), use either reversed type or a screen-patterned overlay to show appropriately highlighted parts.

(6) Differentiate in some way between what the user enters on the screen and the information (such as field names, prompts, messages, and data) displayed by the computer. Some devices for setting the user input apart are boldface or italic type, highlighting with a screen pattern, or using indicating arrows or surrounding boxes. Color is another way; however, color printing is expensive and your budget may dictate a more economical solution.

Figures 8-1 through 8-3 are examples of how these devices can be used to reproduce screen displays.

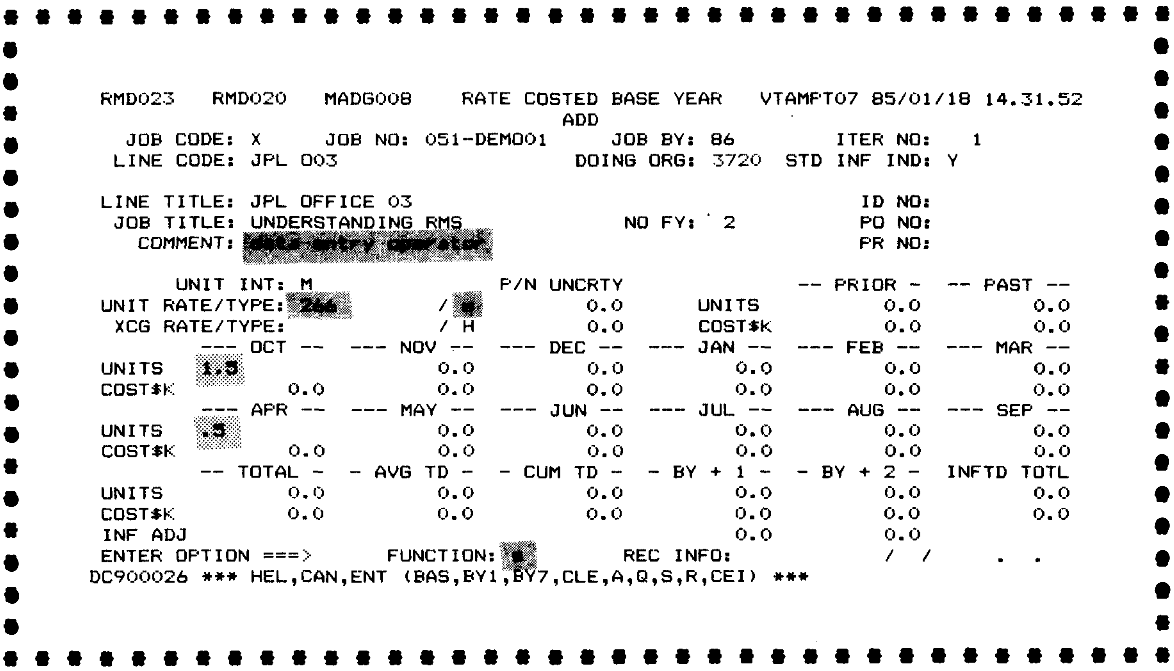
HOW TO READ AN ACKNOWLEDGMENT LEVEL



```
IN TRAY > [illegible]

Exiting the IN TRAY.

[illegible] TRAY of [illegible]                    [illegible]

2 Messages.

Item        [illegible]                                  [illegible]

  1         [illegible]                                  [illegible]
  2         [illegible]                                  [illegible]

[illegible]      open 2  <Return>

[illegible]

TITLE:  [illegible]

[illegible]

[illegible]  2 >  [illegible]  <Return>

Closing message

[illegible] >
```

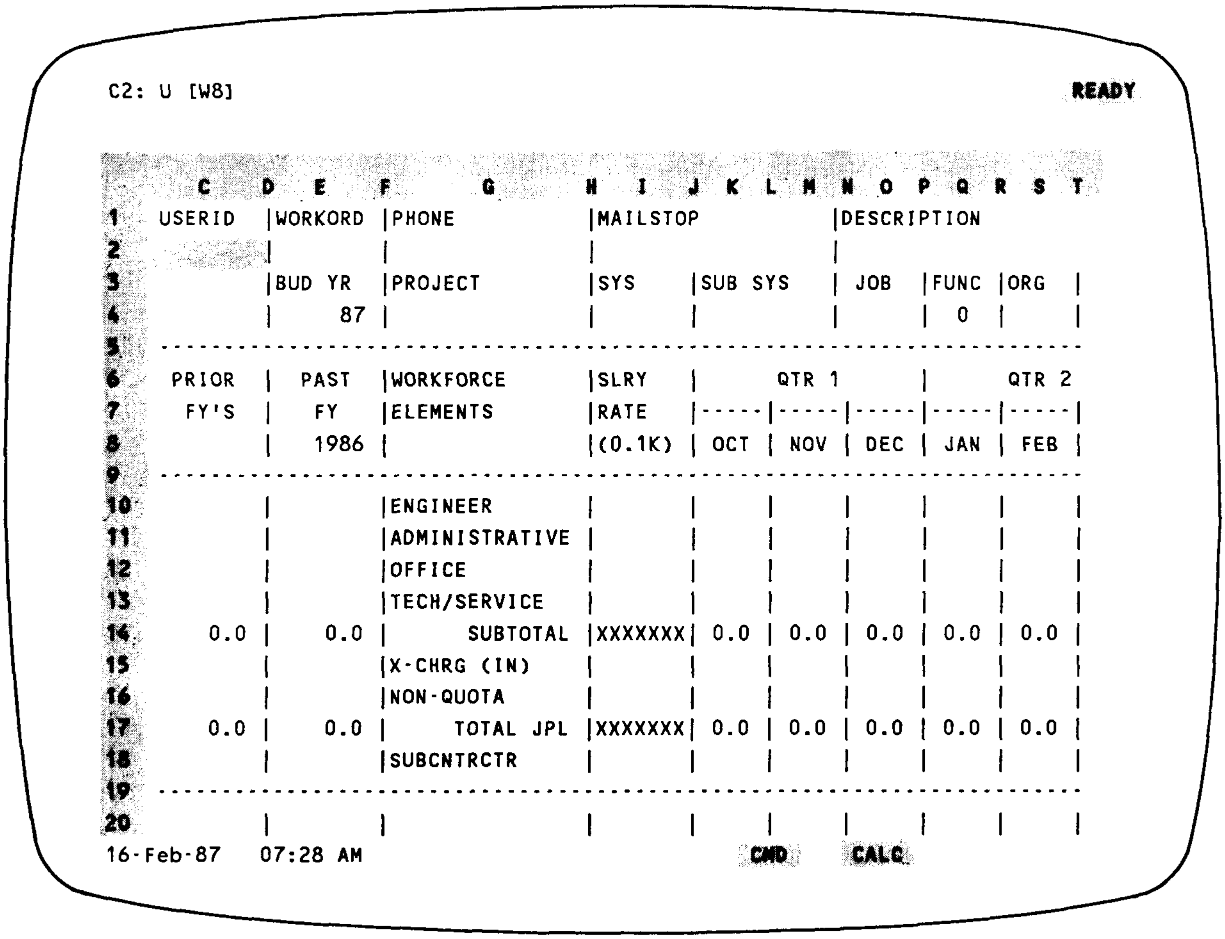
Figure 8-1. *Example of a Screen Facsimile Using a Screened Background With White Letters for Computer Output and Black Letters for User Input. (Notice that an entire procedure is presented in one screen illustration, even though half the information scrolls off the top of the screen before the procedure is complete.)*

ORIGINAL PAGE IS
OF POOR QUALITY



```
RMD023   RMD020   MADG008    RATE COSTED BASE YEAR   VTAMPT07 85/01/18 14.31.52
                                        ADD
  JOB CODE: X      JOB NO: 051-DEMO01      JOB BY: 86        ITER NO:   1
 LINE CODE: JPL 003                       DOING ORG: 3720  STD INF IND: Y

LINE TITLE: JPL OFFICE 03                                        ID NO:
 JOB TITLE: UNDERSTANDING RMS                  NO FY: ' 2        PO NO:
   COMMENT: data entry operator                                  PR NO:

      UNIT INT: M                 P/N UNCRTY                -- PRIOR -   -- PAST --
UNIT RATE/TYPE: 266         / [illegible]     0.0        UNITS          0.0          0.0
 XCG RATE/TYPE:             / H        0.0        COST$K         0.0          0.0
          --- OCT --   --- NOV --   --- DEC --   --- JAN --   --- FEB --   --- MAR --
UNITS     1.5               0.0          0.0          0.0          0.0          0.0
COST$K          0.0         0.0          0.0          0.0          0.0          0.0
          --- APR --   --- MAY --   --- JUN --   --- JUL --   --- AUG --   --- SEP --
UNITS     .5                0.0          0.0          0.0          0.0          0.0
COST$K          0.0         0.0          0.0          0.0          0.0          0.0
          -- TOTAL -   - AVG TD -   - CUM TD -   - BY + 1 -   - BY + 2 -   INFTD TOTL
UNITS           0.0         0.0          0.0          0.0          0.0          0.0
COST$K          0.0         0.0          0.0          0.0          0.0          0.0
INF ADJ                                               0.0          0.0
ENTER OPTION ===>      FUNCTION: [illegible]         REC INFO:                / /        . .
DC900026 *** HEL,CAN,ENT (BAS,BY1,BY7,CLE,A,Q,S,R,CEI) ***
```

Figure 8-2. ***Example of a Screen Facsimile Using a Screen-Patterned Overlay To Differentiate User Input From Computer Output***



```
C2: U [W8]                                                          READY

      C      D   E    F        G        H  I  J  K  L  M  N  O  P  Q  R  S  T
 1  USERID  |WORKORD |PHONE              |MAILSTOP            |DESCRIPTION
 2          |        |                   |                    |
 3          |BUD YR  |PROJECT            |SYS     |SUB SYS    | JOB  |FUNC |ORG  |
 4          |     87 |                   |        |           |      |  0  |     |
 5  ------------------------------------------------------------------------------
 6   PRIOR  |  PAST  |WORKFORCE          |SLRY    |       QTR 1      |      QTR 2
 7    FY'S  |   FY   |ELEMENTS           |RATE    |-----|-----|-----|-----|-----|
 8          |   1986 |                   |(0.1K)  | OCT | NOV | DEC | JAN | FEB |
 9  ------------------------------------------------------------------------------
10          |        |ENGINEER           |        |     |     |     |     |     |
11          |        |ADMINISTRATIVE     |        |     |     |     |     |     |
12          |        |OFFICE             |        |     |     |     |     |     |
13          |        |TECH/SERVICE       |        |     |     |     |     |     |
14      0.0 |    0.0 |        SUBTOTAL   |XXXXXXX | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
15          |        |X-CHRG (IN)        |        |     |     |     |     |     |
16          |        |NON-QUOTA          |        |     |     |     |     |     |
17      0.0 |    0.0 |       TOTAL JPL   |XXXXXXX | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
18          |        |SUBCNTRCTR         |        |     |     |     |     |     |
19  ------------------------------------------------------------------------------
20          |        |                   |        |     |     |     |     |     |
16-Feb-87   07:28 AM                                  CMD      CALC
```

Figure 8-3. *Example of a Screen Facsimile Using a Screen-Patterned Overlay To Show Reversed Video Portions of Display*

8.2 GENEROUSLY USE DIAGRAMS, FLOW CHARTS, AND OTHER MODELS

Keep in mind the truism "a picture is worth a thousand words." Anytime you

(1) give an overview
(2) describe a process
(3) describe a system architecture
(4) explain the structure of a database
(5) describe a hierarchy of nested menus
(6) trace the flow of data or lightning-fast electronic impulses

try to think visually. Many users look at nothing but the pictures; most users who do read the text depend on visuals to help them fully and quickly grasp the information.

Every user's manual should have a brief overview of the entire system or application in the first or second section. A good idea is to make a simplified diagram of the entire process or architecture before attempting to describe it in text. Then, let the text describe the diagram, adding details of terminology. Later, you may wish to repeat parts of the overview diagram, adding specific details, as you discuss the functions, features, and operation of the system in detail.

Figure 8-4 is a sample illustration of a system overview, with specific steps of the process further elaborated in Figures 8-5 and 8-6. In this instance, Figure 8-4 appeared in the module that initially overviewed the system. Figure 8-5 appeared in one of the modules describing the procedure to download data from the mainframe to the PC, while Figure 8-6 appeared in the module describing how to translate the downloaded data and import it into a spreadsheet file.

Figure 8-7 is a diagram showing a good way to present a hierarchy of nested command options. A separate diagram such as this would be required for each command on the main command menu. A similar type of diagram could be used to show a hierarchy of nested screens in a complex system. Every screen the user would encounter (except help screens) would be shown, showing its relationship to the other screens.

Figures 8-8, 8-9, and 8-10 are examples of diagrams that model the structure of a database. The database structure is modeled from the user's point of view, rather than the programmer's. It is a simplified way for the user to understand how the data elements are interrelated, and what elements are key. This type of conceptualization is invaluable to users, helping them to quickly learn and remember their way around a system.

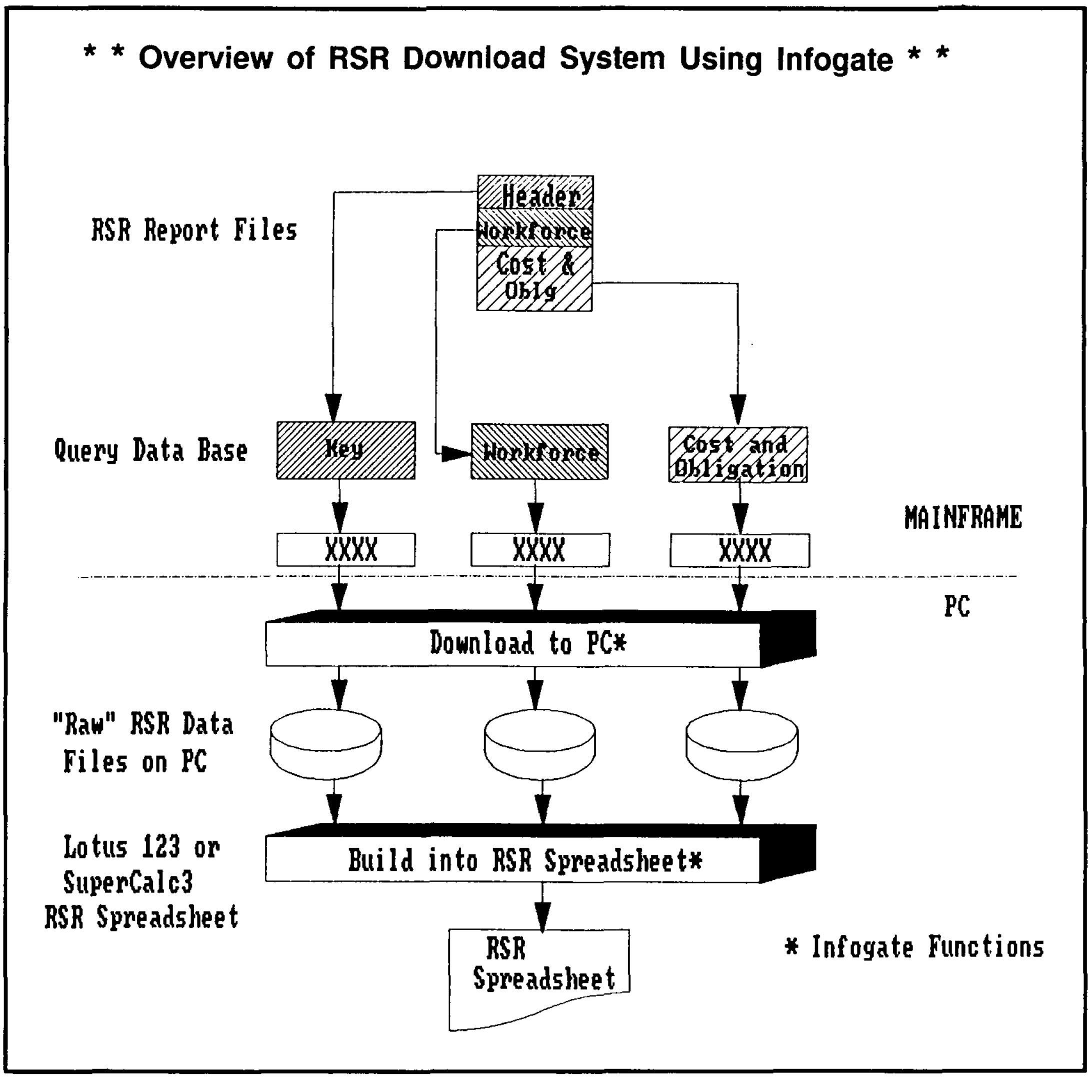


Figure 8-4. *Example of a Simple Diagram Overviewing a Complete Process*

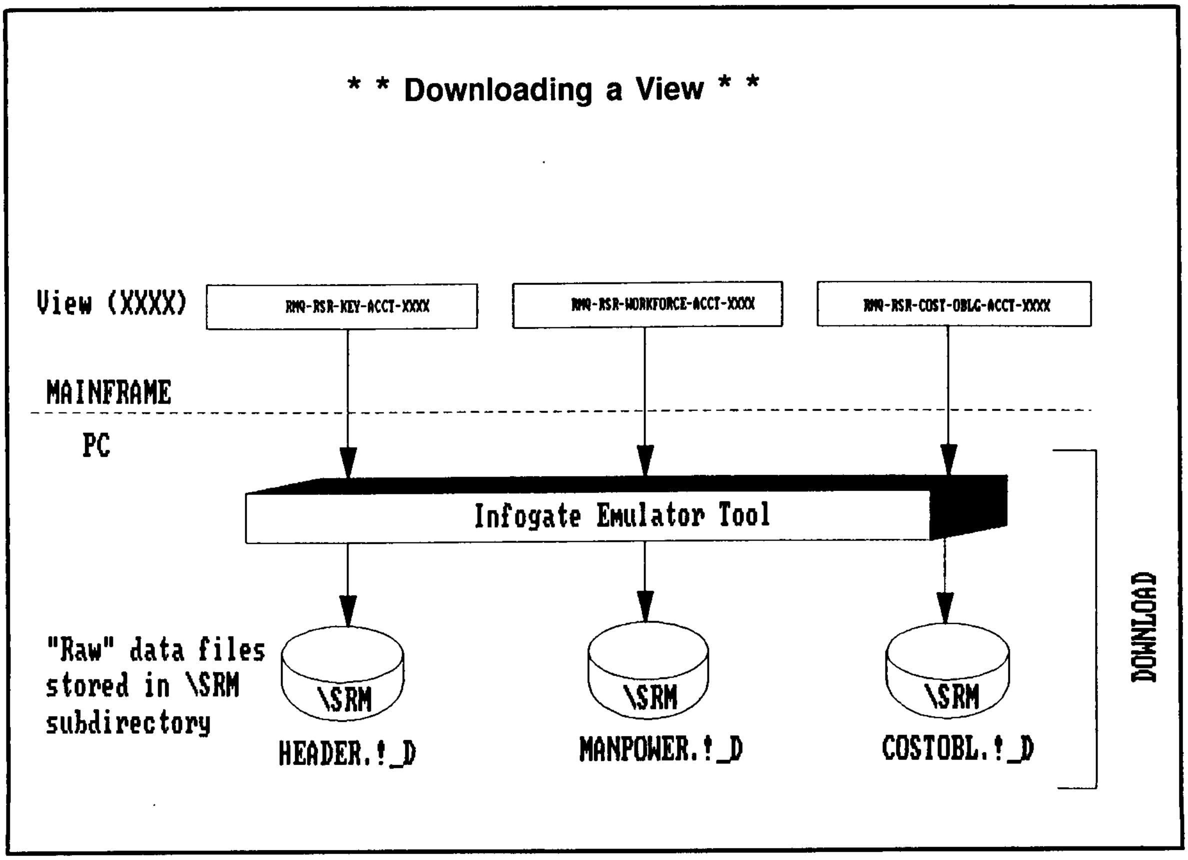

* * Downloading a View * *
View (XXXX)
RMQ-RSR-KEY-ACCT-XXXX
RMQ-RSR-WORKFORCE-ACCT-XXXX
RMQ-RSR-COST-OBLG-ACCT-XXXX
MAINFRAME
PC
Infogate Emulator Tool
DOWNLOAD
"Raw" data files stored in \SRM subdirectory
\SRM
\SRM
\SRM
HEADER.!_D
MANPOWER.!_D
COSTOBL.!_D

Figure 8-5. *Example of a Diagram Illustrating a Process in Some Detail. (This figure is an elaboration of the middle portion of Figure 8-4.)*

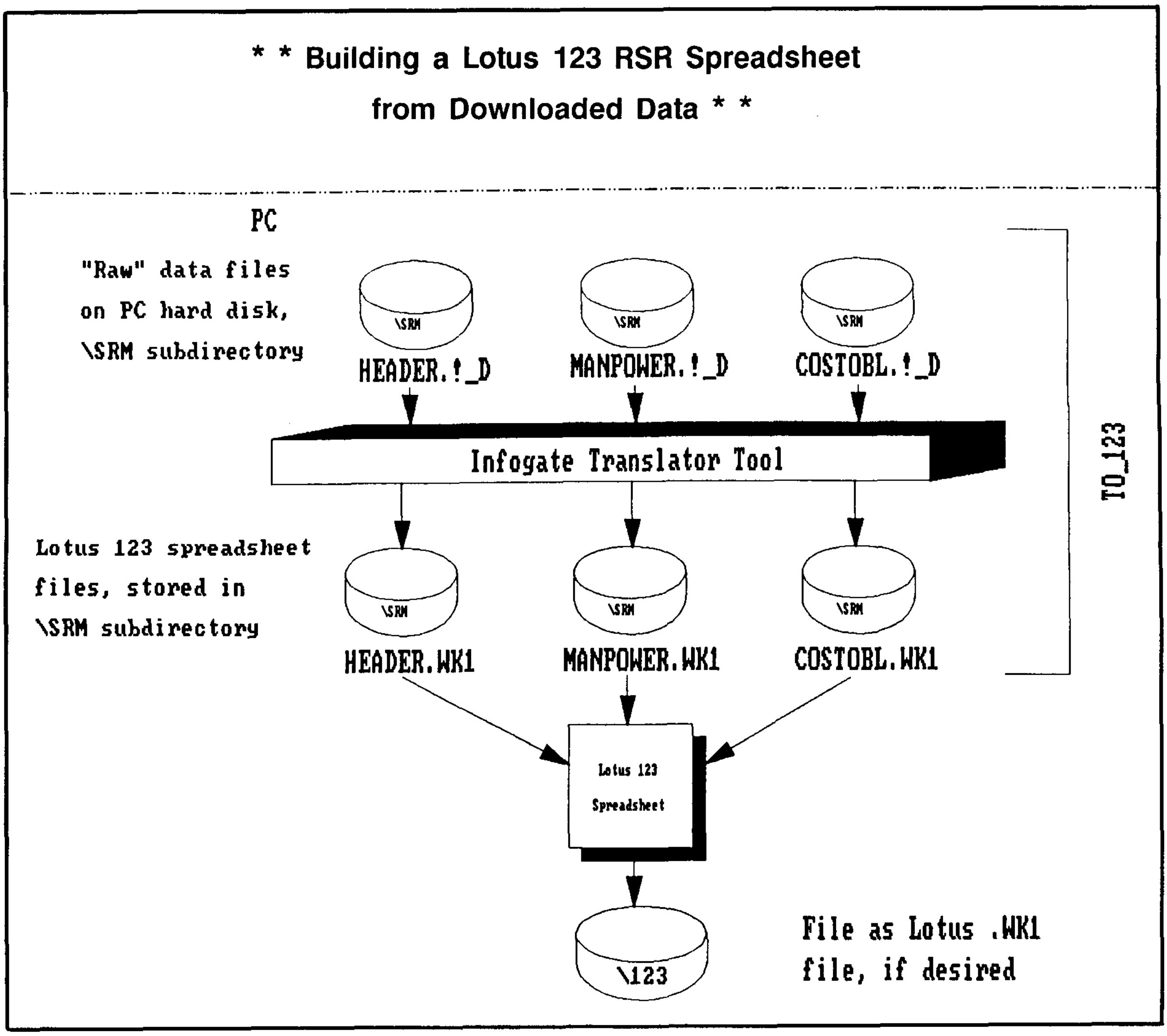


Figure 8-6. *Example of a Diagram Illustrating a Process in Some Detail. (This figure is an elaboration of the lower portion of Figure 8-4.)*

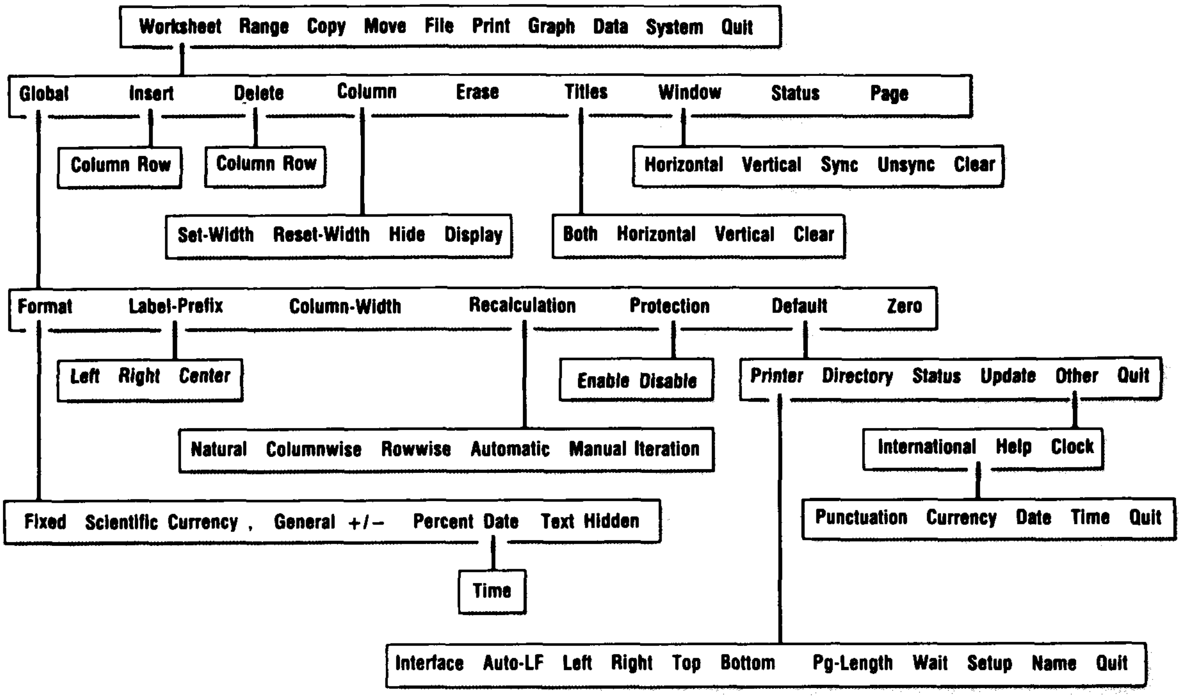
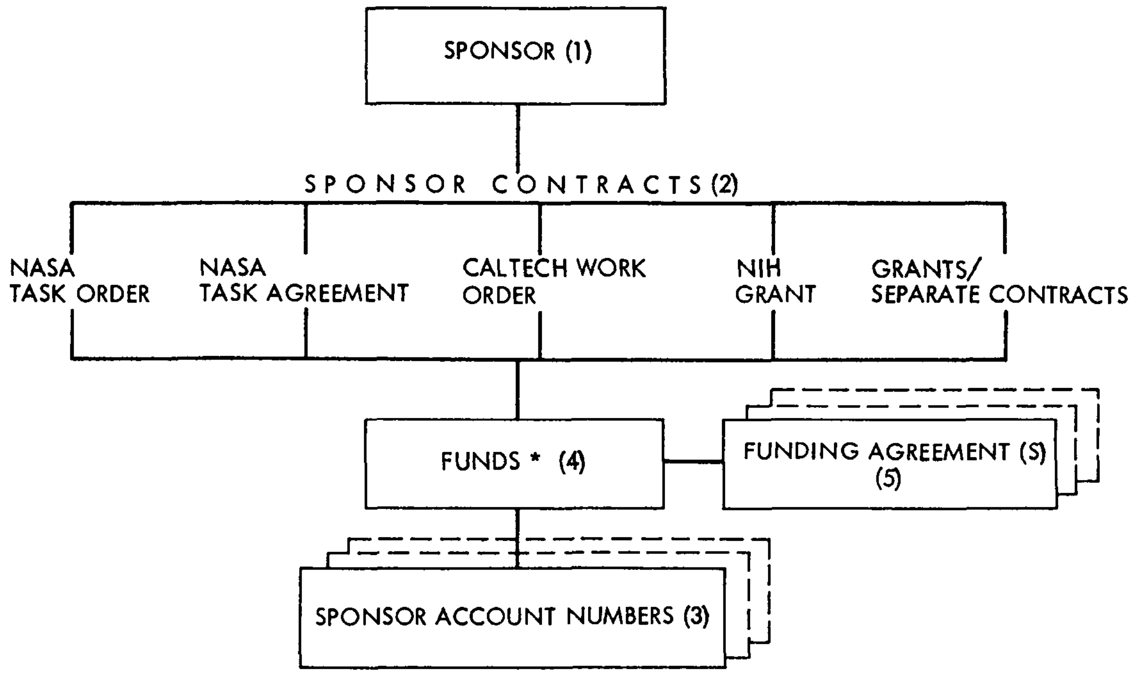

Worksheet Range Copy Move File Print Graph Data System Quit
Global Insert Delete Column Erase Titles Window Status Page
Column Row
Column Row
Horizontal Vertical Sync Unsync Clear
Set-Width Reset-Width Hide Display
Both Horizontal Vertical Clear
Format Label-Prefix Column-Width Recalculation Protection Default Zero
Left Right Center
Enable Disable
Printer Directory Status Update Other Quit
Natural Columnwise Rowwise Automatic Manual Iteration
International Help Clock
Fixed Scientific Currency , General +/− Percent Date Text Hidden
Punctuation Currency Date Time Quit
Time
Interface Auto-LF Left Right Top Bottom Pg-Length Wait Setup Name Quit

Figure 8-7. *Example of a Diagram Showing a Hierarchy of Command Options in a Software Application. (This sample is from the Lotus 1-2-3 documentation, reprinted with permission from Lotus Development Corporation.)*



(x) = ORDER OF DATA INPUT: "FUNDS" TIES SPONSOR CONTRACTS TO SPONSOR ACCOUNT NUMBERS.

* THERE CAN BE MORE THAN ONE FUNDING RECORD (SOURCE OF FUNDS) PER CONTRACT AND MORE THAN ONE FUNDING RECORD PER SPONSOR ACCOUNT NUMBER. EACH NEW FUNDING RECORD FOR EACH SPONSOR ACCOUNT NUMBER MUST HAVE A FUNDING AGREEMENT (506) AND MAY HAVE MORE THAN ONE.

Figure 8-8. Example of a Conceptual Model of the Overall Structure of a Database (Modeled From the User's Point of View)

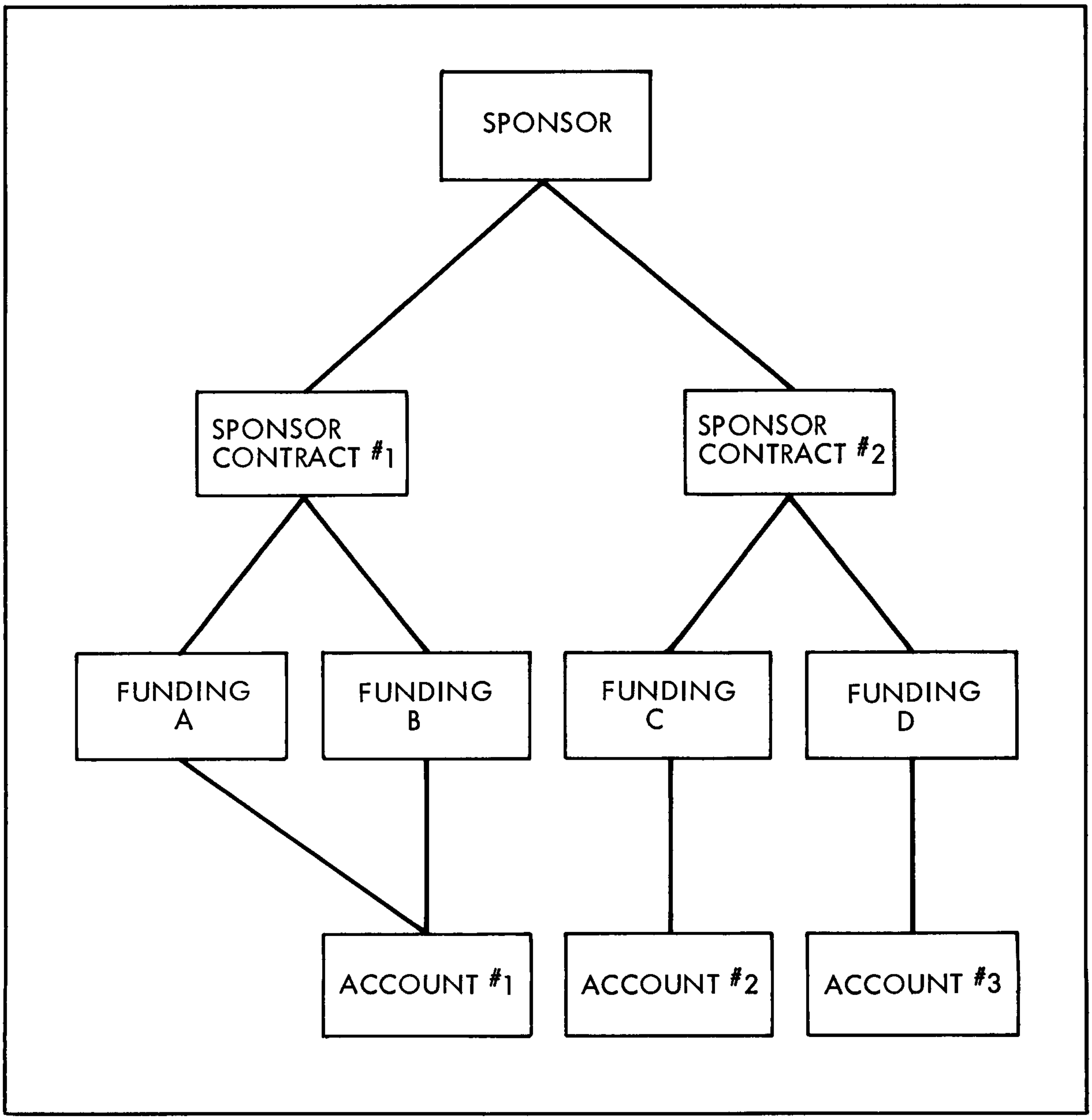


Figure 8-9. *Example of a Conceptual Model of the Structure of a Hypothetical Group of Data in a Database (Modeled From the User's Point of View)*

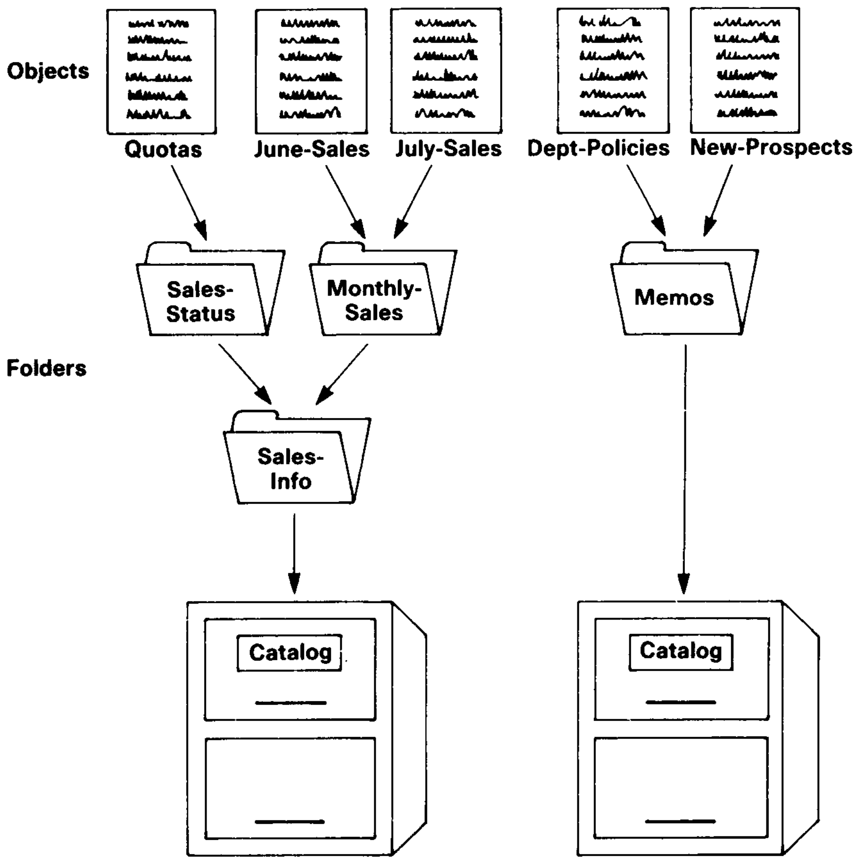


Figure 3-15. Catalogs, Folders, and Objects in INFOGATE

Figure 8-10. Example of a Graphic Interpretation of the Hierarchy of "Files" Provided by Infogate's Information Manager Feature, a Desk Manager Type of Application (Modeled From the User's Point of View), (Reprinted by permission from Cullinet PC Software, Inc.)

SECTION 9

USE STANDARD CONVENTIONS AND TERMINOLOGY

By "conventions," we mean commonly understood ways of expressing certain common procedures (such as pressing the carriage return key) or entities (such as keys on the keyboard).

Whether there is a standard for user document conventions and computer terminology is arguable. However, certain conventions seem to be emerging. Even if some of them seem arbitrary, it is better to adopt a set of conventions, state them up front, and then use them consistently throughout a manual, than to burden the reader with the added task of interpreting instructions given in several different formats. This section describes a suggested set of conventions. (To avoid confusion, examples are shown here in italics.)

(1) Where the user workstation is a microcomputer, the keyboard of the IBM PC/XT may be considered the standard in naming keys. Keys that are identified only graphically on this keyboard would be named as follows:

Symbol on Key	Name To Use in User's Guide
↵	RETURN
←	BACKSPACE
⇆	TAB and BACK TAB
/	SLASH
\	BACKSLASH
↑	UP ARROW
↓	DOWN ARROW
←	LEFT ARROW

	RIGHT ARROW
(none)	SP (or simply leave a space)

(2) Characters that are the shifted case of a key are understood to be the topmost of the two characters printed on the key cap. Therefore, do not add the word "SHIFT" in the instructions to type the character. For example, write

press **!<RETURN>**

Do <u>not</u> write

press **<SHIFT>!<RETURN>**

(3) Names of key caps should be typed in all capital letters and enclosed in less-than and greater-than brackets < >. This practice will prevent confusion as to what characters are to be typed literally and what characters represent names of keys. For example:

Type

cd \ceis<RETURN>

(4) End-user software is often written to accept user input in either upper- or lower-case letters, or a combination. If this is the case, the documentation should make this fact clear, but adopt the convention of showing the user input in lower-case letters. This practice will further help in detailed procedural instructions to distinguish the strings of character keystrokes the user is to type from names of operational or function keys the user is to press, such as <RETURN>, <SHIFT>, and <F2>.

(5) Where the user must supply a specific value to a variable (for example, user ID or password), show the generic name of the variable in lower-case letters and enclose it in less-than and greater-than brackets < >. The brackets will distinguish such variables from literal character strings to be typed and the lower-case will distinguish the convention for generic names of variables from that for operational or function key names. For example,

Type

signon <user id><RETURN>

(6) Show all user input in **boldface** type. User input includes only the keystrokes the user actually makes, including variables that can be indicated only generically.

Do not write, for example,

To change to the XYZ directory, type

cd \xyz and press the <RETURN> key

Instead, write

To change to the XYZ directory, type

cd \xyz<RETURN>

(7) Show spaces in the user input in one of two ways, depending on the likelihood for confusion (which, in turn, depends on the types of input possible):

(a) Simply leave a space where a space is to be inserted, as in

copy a:*.* c:

(b) Refer to the spacebar as <SP> and insert this symbol wherever a space is required in the input. Thus, the example in (a) would be

copy<SP>a:*.*<SP>c:

(8) In documentation for novice users, be sure to state conspicuously (but just once) that ALTERNATE, SHIFT, and CONTROL keys are always pressed simultaneously with another key (or keys). As an example from WordPerfect, the user input would be shown simply as

To center words between the margins, press **<SHIFT><F6>**.

(9) Messages from the computer given in text (as opposed to in screen facsimiles) are to be enclosed in quotation marks. For example,

If you see the message "Warning: Communications link still active," re-enter Infogate and log off the mainframe.

SECTION 10

USE A CLEAR AND CONCISE STYLE

10.1 WHAT DOES "USER FRIENDLY" MEAN?

"User friendly" is a buzz phrase which, unfortunately, is seldom described clearly enough to provide a basis for evaluation. When applied to user documentation written for all levels of experience, this term actually means that the manual

(1) Is written in plain, ordinary English and avoids acronyms, unnecessary data processing jargon, and undefined technical terminology.

(2) Is written in short, uncomplicated sentences.

(3) Is written from the user's point of view, taking into account both beginners and experienced users.

(4) Provides a generous number of examples, showing representative screens wherever appropriate.

(5) For the uninitiated user, gives information in the order needed for use and does not assume knowledge of information not yet given.

(6) For the experienced user, has an excellent cross-referencing index composed of broadly used terms which makes it easy to solve specific problems.

In short, it is a style that intrinsically creates a persuasive document--one that persuades the user that the software product is understandable and easy to use. These characteristics are desirable for all software user's manuals, regardless of the type or complexity of the application or the technical or managerial level of the user. The point is not to create a manual that makes it possible for the user to understand and use the product. Rather, the goal is to create a manual that makes it as easy as possible for the user. Such a manual will assure maximum acceptance, use, and efficiency of the product being documented.

10.2 WRITE FROM THE USER'S POINT OF VIEW

A major aspect of this subject was covered in the section on organization, Section 5. In addition to organizing the manual around the user's tasks, rather than the structure of the software itself, writing from the user's point of view implies a subtle identification of the writer with the user. The writer should hypothetically put himself or herself at the user's workstation and observe the following guidelines:

(1) **Use the second person you** (rather than the third person or passive voice) in referring to the user (reader). Say, for example, "*If you specified dBase as a tool in the installation procedure, the procedure invoked by the . . . .*" **Do not say** "*If dBase was specified as a tool in the installation procedure . . .*" or "*If the user specified dBase as a tool in the installation procedure . . . .*"

(2) Use **imperatives** in giving instructions. Some examples are

> *Insert the XYZ diskette into Drive A:*
>
> *Warm boot the computer by pressing* **<CTRL><ALT><DEL>.**
>
> *Wait while the downloading procedure executes.*

Do not cast such sentences as, for example,

> *The XYZ diskette should be inserted into Drive A:*
>
> *You should warm boot the computer by pressing* **<CTRL><ALT><DEL>.**
>
> *The user should wait while the downloading procedure executes.*

(3) **Give information in the order the user will need it** in performing the task. For example, if any preparatory steps are required, mention them first--not as an afterthought. As another example, if you have just told the user how to access a software configuration profile screen for modifications, proceed to describe how to make the modifications before telling how to exit the screen. (If the procedure is presented in sequentially numbered steps and visually set off from narrative text, the occasional user who changes his or her mind can always skip to the last step.

(4) Actually an issue of organization, **do not assume knowledge of information you have yet to give.** This principle sounds like common sense, but if a user's guide is produced piecemeal, only careful editing can avoid such problems. For example, do not in Section 2 tell a user to "select the desired option" if you do not explain "how to select options" until Section 3.

(5) **Emphasize only information relevant to the user.** In most cases, users are interested only in the parts of a

computer system with which they interact. They are interested in what they see on the screen, what they have to enter from the keyboard, and what they will get back in the form of reports or other results. They do not have to know, for example, details of how the programs work or what kind of mainframe hardware is used. It may be useful to give the user a simplified overview or diagram of such information, but anchor the voice of the document at the user's workstation.

(6) **Be generous in helping the user avoid problems and get out of jams.** For example,

(a) Use highly visible headings in giving information on how to abort a computer operation already under way (such as sorting, compiling, generating, calculating, downloading, etc.). Chances are that, at the times when the users most need this information, they will not have been following your step-by-step procedure as written in the document, but rather will be madly thumbing through the document looking for **HOW TO ABORT THE XYZ PROCESS** to jump out at them. Give them a hand by making it highly visible.

(b) Warn users about operations that, once begun, are irrevocable, or situations that will hang up the system and require turning the computer off or rebooting.

10.3 AVOID ACRONYMS AND JARGON

If only one improvement could be made to most JPL documents to make them more readable, it would be to remove about 98 percent of the acronyms. Until a person has heard and used a particular acronym about 20 times, a mental translation step is required in order to understand it, even if it was defined on the previous page. Acronyms in documents are there for the author's convenience, not for the reader's. Users have enough to do to learn a new application and use it to get their work done, without having to pause to translate acronyms sprinkled throughout the instruction manual.

A few acronyms are so widely used in your particular environment that you may think no translation step is needed by most people. Even most of these acronyms, however, should be spelled out at least the first time used in the document in order to tune in newcomers. Other acronyms, not so widely known but which the writer still feels are necessary, should be redefined on each page. Remember, few users read a manual through from front to back.

10.4 USE A CLEAN, DIRECT STYLE

Although we advocate use of the second person and a basically conversational tone in addressing the user, user friendly does not mean chatty, overly familiar, patronizing, or even entertaining. The most effective style for a user manual is concise, direct, and ever-considerate of the reader. Formal, standard English is used. Stuffiness and dryness are avoided through use of simple, declarative sentences, common terminology, and a conversational tone (e.g., "If you wish to do A, follow these steps:"). The following guidelines elaborate with examples:

(1) **Use formal, 100-percent standard English.** The American language is alive and rapidly evolving. At any point, there will always be some words, spellings, and usages that are in between the time they were considered marks of ignorance or ill breeding and the time they are commonly accepted. In most cases, it is better to use the language conservatively and thus avoid the risk of being catalogued by some readers as ignorant or ill-bred. Have a usage reference book handy and use it often. Reference 3 on page 2 is highly recommended. The book is extremely easy to use. A few examples of these "borderline" cases in the language follow:

   (a) Avoid ending a sentence or clause with a preposition, even though you hear and see this done all the time.

   (b) Commas and periods, when in juxtaposition to closing quotation marks, always go inside the closing quotation marks, whether the quote applies to the whole sentence or not. An exception may be made if the quotation marks are to be included in a literal string of characters to be typed into the computer.

   (c) Be conservative in combining two words into one if you cannot find them combined in your standard dictionary. For example, dataset, datafile, logon, and signon are often seen combined as one word in computer software manuals and even in specialized computer dictionaries. Although these combination words are commonly accepted in the data processing industry, do not forget that many of your readers may not be working in the data processing industry.

   (d) Do not use the possessive form for inanimate objects. Say "the name of the file" or "the file name," not "the file's name."

(2) **Avoid humor.** Even what you might think is your wittiest prose may leave the reader flat. After all, the user is not generally reading your manual to look for laughs. At

worst, you may annoy or offend the reader. Using humor in a manual is just too risky. Save it for a more creative genre.

(3) **Avoid gender exclusivity**. It is an increasingly common and accepted practice to use gender inclusive terminology. Workforce, workhours, staff, or labor (instead of manpower), chair or chairperson (instead of chairman), drafter (instead of draftsman), etc., are in common usage. The use of allegedly generic masculine pronouns has also been largely abandoned in favor of one of the following solutions (listed in order of preference):

(a) Recast the sentence to avoid the necessity of a generic pronoun.

(b) Make the antecedent plural (thus using the truly generic they, them, or their).

(c) Use both masculine and feminine pronouns (he or she, him or her, his or hers).

(4) **Avoid sounding patronizing**. Even if your intentions are otherwise, certain types of comments can sound overly familiar and patronizing. Examples of such phrases are "don't worry," "be patient," "now, that wasn't so difficult, was it?"

(5) **Be concise**. This point cannot be emphasized enough. Conciseness, by nature, makes a document more user friendly because it makes it shorter. It makes important information easier to find. It makes information easier to understand because only the necessary words have to be read and understood. This does **not** mean you should leave out articles, prepositions, and other small words that give clues about the logical structure of the sentence. It simply means to avoid redundancy, overly complex sentence structure, weak descriptive adjectives, long words when shorter words will do as well or better, whole phrases when one word says it all, and generally pointless discussions. All these dilute the information the user really needs and make it harder to find.

(6) **Be decisive**. The user wants to regard the user's guide as the always-handy authority on the product. Because you have thoroughly researched all the information you are communicating, and because you have used the product extensively as a user yourself, you can make statements with authority. There is always a slim chance you could be wrong about something, but be wrong with confidence.

SECTION 11

EDIT RUTHLESSLY

A user's guide draft should not be distributed, even to reviewers, without first being carefully edited. Every writer, no matter how skilled or experienced, needs the services of an editor. It is strongly recommended that this editor be a professional technical editor; only if none is available should another person with proven writing and editing skills be called upon to edit the draft.

An editor is not just another reviewer. An editor is one who shares responsibility for the document with the author. The editor reviews the document as soon as it is reasonably complete and technically accurate, but before anyone outside the immediate development team sees it. Whatever is done with comments from managers, users, and other reviewers, the editor's comments must be seriously considered. As the document draft is revised, the editor is given additional opportunities to review it.

The editing process is intended to improve the draft in six areas:

(1) **Organization and structure.** Make sure organization and structure are clean, logical, and user oriented. Check that modules are brief and contain digestible chunks of information.

(2) **Mechanics.** Correct errors in spelling, grammar, and punctuation.

(3) **Appropriateness of language.** Make sure terminology is common to readers and not unnecessarily difficult. Check for appropriate tone.

(4) **Clarity.** Look for ambiguity, imprecise use of words, or misleading statements. Also, eliminate awkward, wordy constructions, backward sentences, and other common problems.

(5) **Accessibility.** Make sure the index is present and perform several spot checks on page references.

(6) **Urgency.** Revise to make writing more interesting and engaging by use of careful diction, varying sentence length and style, and improving visuals.

There is no substitute for professionalism, regardless of the technical level of the manual. The editing process need not take long if done by a skilled person.

GLOSSARY

Beta users--persons or groups who are selected or who volunteer to test a new software product by using it in their everyday work before the product is released for general distribution or sales.

Chunking--general term for use of various technical writing devices for organizing information into small groupings in order to make a document easier to understand and use as a reference.

Cognitive psychology--the branch of psychology concerned with how human beings process information and learn.

Comb binding--binding technique requiring a special machine to punch holes and insert a plastic comb device to secure the pages together.

Conventions--commonly understood ways of expressing certain common procedures or entities.

Defaults--a value automatically selected by a computer when it is not provided overriding information.

DOS--Disk Operating System; program(s) that control communication and transfer of data between a computer and its disk drive(s).

Index generator--software that operates on a text file to assist in developing an index to a document.

Index tabs--special pages or markers inserted into a document to separate major sections. Often extend beyond the edge of the document.

Jargon--any terminology specific to a particular group and not readily understood by the general population.

Module--a small section (ideally one or two pages) in a document dealing with one discrete topic of interest to a reader.

Output--any information the user receives from a computer system, whether it is a screen display, message, or paper printout.

Perfect binding--binding technique whereby the cover is wrapped around the document and the pages are glued to the inside of the spine.

Play script--a technique of writing a procedure wherein the user's input is clearly differentiated from the computer's response, as in a dramatic script.

Process-oriented outline--a user's guide outline structured around the processes performed by the system, rather than the tasks facing the user of the system.

GLOSSARY (Contd)

Quick reference guide--a small card, booklet, or key template containing only essential, condensed information for performing basic user operations on a computer system.

Saddle stitching--binding technique whereby a document is printed "two-up" on a press, and sheets are folded and stapled or stitched in the center.

Sans serif--having no serifs.

Serif--a fine line finishing off the main strokes of an alphabetic letter, such as the top and bottom of M or ending the cross stroke of T.

Society for Technical Communication--the major international professional organization for technical communicators (technical writers and editors, visual designers, their managers and educators, etc.).

Tutorial--a document (or online application) designed to introduce a user to a software application by giving step-by-step instructions for performing sample user tasks.

User-oriented outline--a user's guide outline structured around the tasks and problems facing the user, rather than the internal processes of the computer system.

INDEX

INDEX (Contd)

INDEX (Contd)

INDEX (Contd)

INDEX (Contd)

READER REPORT FORM

GUIDELINES FOR PREPARING SOFTWARE USER DOCUMENTATION

Please use this form to give comments and suggestions for improving this document. Return it to

Diane F. Miller
Jet Propulsion Laboratory
Mail Stop 171-209
4800 Oak Grove Drive
Pasadena, California 91109

Name ______________________________ **Date** __________________

Address ____

____

Comments or Suggestions:

TECHNICAL REPORT STANDARD TITLE PAG

1. Report No. JPL Pub. 87-22

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Guidelines for Preparing Software User Documentation

5. Report Date
October 1, 1987

6. Performing Organization Code

7. Author(s)
Diane F. Miller

8. Performing Organization Report No

9. Performing Organization Name and Address

JET PROPULSION LABORATORY
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109

10. Work Unit No.

11. Contract or Grant No.
NAS7-918

13. Type of Report and Period Coverec

JPL Publication

12. Sponsoring Agency Name and Address

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546

14. Sponsoring Agency Code
B-055-00-00-00-94

15. Supplementary Notes

16. Abstract

Clear, easy-to-use software user's manuals make strong demands on special technical communication techniques. Principles and guidelines are given for analyzing the audience and dealing with wide-ranging backgrounds of potential users. Types of information to be included in a complete manual are suggested, with a technique for creating a user-oriented rather than process-oriented organization. Accuracy verification is emphasized. Simple tips are given for formatting for quick comprehension and reference, for deciding on packaging, for creating helpful illustrations and examples, and for setting up clear and consistent conventions. Simple guidelines are offered for writing clearly and concisely and for editing.

17. Key Words (Selected by Author(s))

Computer Programming and Software
Technical Writing
Software User Documentation

18. Distribution Statement

Unclassified--Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
85

22. Price