Professional™ 300 series

Documentation Guidelines

Order No. AA-N621A-TK

Developer's Tool Kit

digital
software™

Documentation Guidelines

Order No. AA-N621A-TK

December 1982

These guidelines suggest to managers and others how to document applications written to run on the Professional 300 Series.

The following are trademarks of Digital Equipment Corporation:

CTI BUS	MASSBUS	RSTS
DEC	PDP	RSX
DECmate	P/OS	Tool Kit
DECsystem-10	PRO/BASIC	UNIBUS
DECSYSTEM-20	Professional	VAX
DECUS	PRO/FMS	VMS
DECwriter	PRO/RMS	VT
DIBOL	PROSE	Work Processor
digital	Rainbow	

CONTENTS

APPENDIX A   PROFESSIONAL 300 SERIES SPECIFICATION END USER DOCUMENTATION

TABLES

CHAPTER 1
WHY DOCUMENTATION?

Do you really need documentation? After all, your software is good—and that's what counts. In fact, many of today's successes didn't even consider documentation for their software when they were starting out. And anything more than a cursory attempt means that documentation becomes a real budget item. Your business is software, not publishing.

On the other hand, will people who buy your software know exactly what they're buying? What happens when someone using your software loses a month's records? Can you afford hand-holding field support and emergency phone calls?

The truth is, documentation takes away the guesswork. Combine documentation and your unique application, and you have a total product that answers all the questions. Plus, you've given your software a better chance in today's market of professionals.

But you need to go beyond holding down field support costs, protecting yourself against suits, and providing a sales tool. Safety considerations and an accurate description of the software are not sufficient. You need to know your prospective buyers and write/produce with them in mind. You need documentation that is well designed and does its job—like your software.

This book tells you how to obtain accurate well-designed documentation, but it doesn't tell you how to write. After reading the book, you will know what criteria you need to make decisions about planning, writing, and producing documentation. You will also learn what is involved in the documentation product cycle.

Use the answers in the following list to work out your own needs. You have to know the purpose of your documentation before you can decide what kind you want.

Some answers to "Why documentation?"

1. To use as a sales/marketing tool.

    a. Present your product at a time and place convenient for the client.

    b. Substitute hard copy for a sales visit, send it ahead, or leave it as a reminder.

    c. Provide the sales people with drafts so that they can be better prepared and get a head start.

    d. Give a prospective buyer confidence in the product with a clear presentation and time to think.

    e. Keep sales/marketing on target with accurate documentation.

    f. Extend the customer base.

2. To keep down costs.

    a. Substitute for field support, or cut down on the need for it.

    b. Provide end users with the information necessary to use the software without jeopardizing vital data.

3. To make a better product.

    a. Make yourself and the customer happy.

    b. Maintain good relations for future products.

CHAPTER 2
BEFORE DOCUMENTATION

At the same time that you make a plan for your software project, start a parallel plan for documentation—because good documentation doesn't come after the software, but is part and parcel of the product. Documentation developed along with software can hold programmers to original intentions, help them recognize inadequate specifications, and make them shape the user interface in a consistent and clear way.

2.1 AREAS OF RESPONSIBILITY

Planning for documentation involves more than someone to write and someone to develop the software. You need representation from all the areas that documentation touches—Sales/Marketing, Field Support/Training, Legal, Writing/Production, and Software. Make sure that these areas are represented in your plan, so that even if very few people are involved, you'll have a checklist to cover the various responsibilities. Once the responsibilities are clear, they can be assigned according to your group's organization.

Table 2-1
Areas of Responsibility

Area	Responsibility
Legal	Warranties
	Copyright
	Language of Cautions and Warnings in text
	Language of text for truthful description of the product
	Product name (patentability)
Field Support/Training	Warranties
	Language of Cautions and Warnings in text
	Accuracy and clarity of text
Sales/Marketing	Warranties
	Language of text for truthful, competitive description of the product
	Clarity and attractiveness of the documentation
	Appropriateness of style
	Product name (competitiveness)
Writing/Production	Appropriateness of style
	Accuracy, clarity, and attractiveness of the documentation
	Product name (as language)
	Budget
	Schedule
Software	Support for all items of responsibility under Writing/Production

Include the documentation in your project plan for the total product so that the various areas involved can be taken into account and so that you can get the best documentation to complement your software.

2.2 DOCUMENTATION PLAN

You have a software project, and you've completed preliminary discussions. With some idea of what everyone wants from documentation, you can have the documentation plan written up and then reviewed by the same people you brought together for the start-up phase.

Who writes the documentation plan? Someone who has analyzed the product and is aware of the needs of all those involved in the planning effort. Someone who is organized, literate, and logical.

2.2.1 Criteria for Decisions

Whatever decisions have been made for marketing and selling the product drive the decision-making for documentation. Specifically, a first basic question is Who is the documentation for? closely followed by When is it needed?

The first question not only determines the presentation and style of the manual but can also set off a string of related questions: How many manuals do we want? How much overlap is there between manuals? What is the purpose of each manual?

The second basic question—When is it needed?—brings up schedule and budget, which are interdependent and determine the quality of the finished manual on all levels.

2.2.2 Contents of the Documentation Plan

1. Goals, intended audience, summarized content
2. Competition—comparison
3. Staffing and responsibilities, including reviewers; budget
4. Schedule; includes times for drafts, reviews, and editing
5. Production method and schedule; includes page counts, number of copies to be printed, and art requirements; budget
6. Annotated outline of the manuals

2.2.3 Goals

If this section doesn't write itself, start again. By now, you should know what you think of your software and how you want buyers to think of it. The important thing to remember is that you cannot assume that you have the answers. Make your goals as explicit and precise as possible. For example, if you want a friendly style, don't assume that your idea of friendly is the same as someone else's—it may be insulting. Socioeconomic and national differences alone make for a wide range of interpretations of "friendly". Another difficult approach, for the same reasons, is a humorous style. Remember your intended audience and give them the information they want about the software.

The questions that need to be asked here are: How much of the software needs to be explained? How much software knowledge can be expected of the user? What is the user's level of education? Answers to these questions help structure the documentation and dictate the style. A software engineer, for example, may not want to read explanations of menu structures that guide a user through the system with encouraging words. On the other hand, an alphabetical list of commands with abstract parameters is not the best way for a nontechnical manager to approach a new system.

2.2.3.1 Types of Manuals —Traditionally, software manuals have been categorized according to usage; the list that follows shows these categories Although titles of this kind are not necessary, and may mean nothing to new users, it is probably a good idea to have them in mind—if only to prevent confusion.

☐ *Introductory* For users without software experience. The writing task here is to decide what information is helpful and necessary and present it so that it can be understood and retained. Repetition, examples, and graphics contribute most to the success of this kind of manual.

☐ *User's Guide* For users with or without software experience. The purpose of a user's guide is to describe how to best use the software to make the most of its capabilities.

☐ *Installation Guide* For those who have to install the software. Installation guides are best kept as simple as possible, with step-by-step procedures and consistent terminology. They may be included in introductory manuals or user's guides.

☐ *Programming Language Manual* For reference use, although it may have some tutorial material on how to best use the particular language. Users are generally assumed to know how to program with the software. The components of the language have to be presented in a way that is useful—that is, alphabetically within functional groupings.

☐ *Cards, Mini-References* For users who need a quick reminder or only certain basic commands. These can also be useful as sales tools.

2.2.3.2 System Documentation —The documentation of the system that your application will run on is as important as your own documentation. It should be read carefully. Make sure that the writing for your application is compatible by using the same terminology and the same approach for the same user.

2.2.4 Competition

There is documentation competition as well as software competition. Analyze how successful the competing documentation is by how well it measures up to the standards you are setting for your own documentation.

2.2.5 Staffing

The documentation has to be researched, written, reviewed, and produced. This means a minimum of two people, since it's hard to do a good job of reviewing your own work. In practice, the research and writing are done by one person; the reviewing by yet another person or persons; and the production by a third person or group.

Staffing must be considered with scheduling and budgeting, since you need to know how many people will be on the documentation staff, how much they cost, and how much of their time over a given period of time you will have. In particular, other projects may not be completed, or may be dependent on this project to finish by a certain date. An efficient way out of this kind of problem may be to hire contract help.

2.2.5.1 Contract Help —Small firms or groups having a specific need for a short time are ideal candidates for contract help. Someone recommended by a knowledgeable documentation person is also preferable to someone you know little about. So, the only difference in hiring contract documentation help, as opposed to contract software help, is that you probably don't know how to judge a good documentation person. If you know what you want a person to do, this will help you decide. The following sections give some background on the people who might be useful for your documentation. The information may also be helpful if you are thinking of developing a documentation group. (Chapter 3 provides information about writers.)

If you decide you need a contract documentation person, make clear what you want from that person. Even more than for in-house people, responsibilities must be clearly defined. It is up to you to know what you want, when, for how long—and for how much.

2.2.5.2 Editor —You will probably find an editor difficult to evaluate because "editor" is such a broad term. As a reviewer, an editor holds the writer to the best implementation of the goals stated in the documentation plan. This kind of editor is sometimes called a "developmental editor" or, misleadingly, a "literary editor", to distinguish from the more mechanical copy editor who edits just before production. Above all else, a developmental editor needs to be a good writer whose authority is accepted.

If you are looking for a copy editor, you want someone with the ability to concentrate on details with consistency and to correct grammar and awkward style. The copy editor's patience may or may not be accompanied by the developmental editor's flair and tendency toward analytical overview. A copy editor may also mark up a text for a typesetter and help with the final phases of the manual.

2.2.5.3 Proofreader —Although a copy editor may proofread, your project's budget and schedule could require the hiring of a proofreader. Your decision will depend on how you divide up the documentation functions of research, writing, reviewing, and producing, and especially on what kind of editor you have. Remember, too, that a lot of simple checking is best done with a computer program that compares two text files.

A proofreader must find and correct errors and be able to understand the copy editor's marks. To do the job properly, the proofreader checks the proof (final output) against the copy (the input to the proof). In fact, the production method you choose defines the proofreading task for your project. But don't believe that because there was no re-entering of data there can be no errors. Lines can disappear; strange symbols can print instead of capital letters; tabular material can become garbled. Someone should check for these machine errors as well as for the errors that arise from misunderstanding the format specification.

2.2.5.4 Designer —Like the editing function, design spans a wide range, a part of which you may want to contract. A book designer, at one end of the spectrum, needs a clear idea of the manual provided by the developmental editor, the writer, or a marketing person. All the issues considered in the Goals section of the documentation plan must be taken into account by the designer, together with schedule and budget constraints. A good book designer can give a book a professional quality that transcends a low-budget production method. Since many book designers are accustomed to working on a contract basis, your designer can probably come up with a set of ground rules to work by. Helpful hint: If your documentation is straightforward, think about looking for a designer in the textbook field.

2.2.5.5 Illustrator —A book designer's services may include illustration and pasteup either directly or by subcontract. If you contract separately for an illustrator, have an idea of the kind of illustration you want before you go looking. For example, do you want charts, graphs, and flowcharts or do you want people to be drawn? Some illustrators can work with ideas to produce a helpful graphic; others are more mechanical.

2.2.5.6 Pasteup —If you decide—for reasons of time, money, and availability that you will hire someone to paste up text into pages for the printer (and this stage may not occur in your particular production method), use an agency if you have no recommendation. Provided you give clear instructions, pasteup is a mechanical process. Someone with knowledge of the book's specifications should check the pasteup for bad page breaks.

2.2.5.7 Typesetter —You choose a typesetter the way you would a plumber or a roofer. Look at previous work, compare comparable costs, and ask for references. The ability to take your files via data line and compatability of equipment are special points to look out for. "Comparable" costs mean that one typesetter may offer a low bid per page but charge a lot for changes. Another may want to do a complete job from typesetting to pasteup, including proofreading, and therefore give a good price for the total job but a comparatively high price for any one part. This is where you can save—and spend—money. It's worth careful consideration. Remember what you want for the product, and go with the best for your money.

2.2.6 Schedule

```
                    ( Start )
                        |
              ┌──────────────────┐
              │ Documentation    │
              │ need             │
              │ identified       │
              └──────────────────┘
                        |
              ┌──────────────────┐
              │ Documentation    │
              │ plan             │
              │ written          │
              └──────────────────┘
                        |
                    < Plan >── No ──┐ ┌──────────────┐
                    < OK?  >        └─│ Plan         │
                        |            │ corrected    │
                      Yes            └──────────────┘
                        |
              ┌──────────────────┐
              │ Draft            │
              │ written          │
              └──────────────────┘
                        |
                    < Draft >── No ──┐ ┌──────────────┐
                    < OK?   >        └─│ Draft        │
                        |             │ corrected    │
                      Yes             └──────────────┘
                        |
              ┌──────────────────┐
              │ Illustrations    │
              └──────────────────┘
                        |
              ┌──────────────────┐
              │ Review           │
              │ draft            │
              └──────────────────┘
                        |
                    < Draft >── No ──┐ ┌──────────────┐
                    < OK?   >        └─│ Draft        │
                        |             │ corrected    │
                      Yes             └──────────────┘
                        |
              ┌──────────────────┐
              │ Final            │
              │ production       │
              └──────────────────┘
```
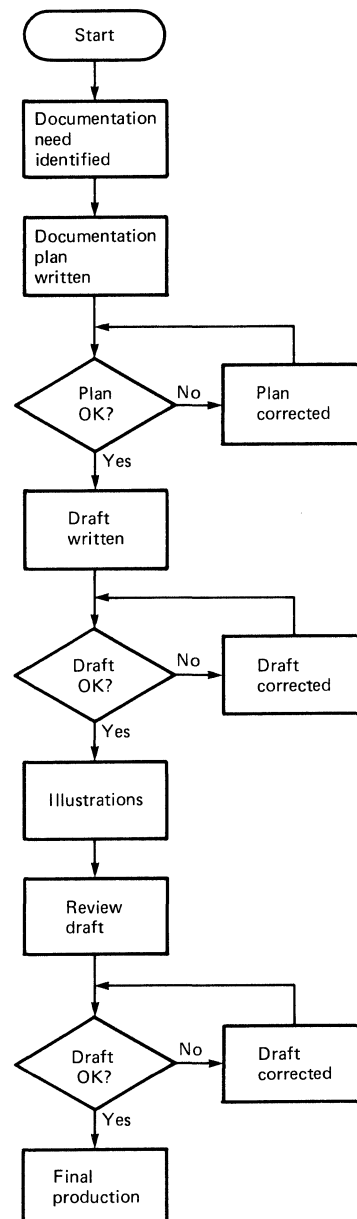
The documentation plan gives dates for the major points in the documentation product cycle. The dates are always dependent on the software product cycle, so you should have plans to deal with the realities of software slips. These Guidelines don't tell you how to schedule, but they do tell you what you have to think about and breaks the total project into manageable parts.

The first phase of the cycle is over when the documentation plan is written and approved. The second phase begins with the writing of a draft and ends when a draft is approved. The writing of a final draft and approval of that draft marks the third phase, which is the final phase for the writer, unless the writer also has responsibility for production. The production phase is the fourth and final phase of the documentation cycle. It ends when the documents are in final form, ready to be used.

If you are tempted to save time on the planning, you should know that you run the same risk as when you skimp on planning in any project. Writing can be done at various speeds, but not before the software. However, if a final shipping date is absolute, and software is lagging, you might consider making a list of software features that have not been pinned down and that need to be documented. Software development can use the list to decide what features will not be incorporated into the software for the coming release. Writers can use the list to strike a bargain with software development and write up features that are guaranteed to be part of the final software and guaranteed to be unchanged from the design.

Production is also an area where you might want to cut corners to save time. Because production is the last phase, it is particularly vulnerable to schedule pressure. Typesetters and printers can put on extra staff and work overtime; all the mechanical aspects of production can be finished faster by using more people and working longer hours. Find out ahead of time, in the planning stage, how many pages an hour a pasteup person can handle, and how many days a typesetter needs, so that you have a range from comfortable to impossible. Armed with the numbers, you can push back suggestions for "making up time in production".

2.2.7 Production

Documentation is written with the production method in mind. For example, the availability of boldface and italic type can make a difference to the way information is presented. In the documentation plan, you state what you have decided to spend preparing reproducible copy for the printer and how long it can take. To make a decision, you have to take into account your current equipment and personnel, and then what supplements are necessary. Find out what is available and what you get for your money. Take the advice of editors, writers, designers, and anyone interested in how the final documentation looks. In any event, make sure that the drafts are easily translatable into final copy by having a total writing plan that takes equipment into account.

To determine costs, you must have an estimated page count, since prices will be quoted in terms of pages. You also need an estimate of the number of copies to be printed, since the more you print the larger the total cost, but the lower the price per copy.

It is also a good idea to know at this point what page size you want. A small page size may seem friendly; the traditional 8 1/2" x 11" page may seem cumbersome. But what does your application need? Will you have to reduce illustration labels to a tiny type size? Will you have a lot of material that begins on a new page but only takes a third of the page? Does the manual have to fit with other manuals? Before you make a page size decision, talk to printers and find out what your choices are. There is very little standardization in book page size. However, if you want to put your manual in a binder, you do have to consider costs, because off-the-shelf binder sizes are limited.

The last special cost factor to consider in production is the use of special art. Are there half tones or computer graphic displays? Are there illustrations that appear on a fold-out page? Do you want color? These all take time and cost money. Weigh the time and cost against what they bring to the manual, and then decide.

A detailed schedule is not necessary at the documentation plan stage, but you will need one eventually. For the plan, estimate about eight weeks from the final approved draft to a finished book if the book is about 100 pages. This is a generous amount of time that allows bargaining time, depending on the method of production and your resources.

2.2.8 Annotated Outline

An outline given in the documentation plan means that the writer has had to think about the subject and organize it. It may change significantly during the course of the project, because the software or the audience changes, or just because the writer has a better idea. As long as everyone knows about its changes, the outline may be improved with these corrections.

It is good practice to encourage queries on the outline at this stage. Structure changes are more expensive the later they occur in the documentation cycle.

CHAPTER 3
DOCUMENTING THE PRODUCT

Once you have identified what you need and written it up in a documentation plan, which also addresses how you intend to fill the needs, you are ready to start the writing process—provided that everyone involved agrees with the plans. Agreement is really vital at the three decision poi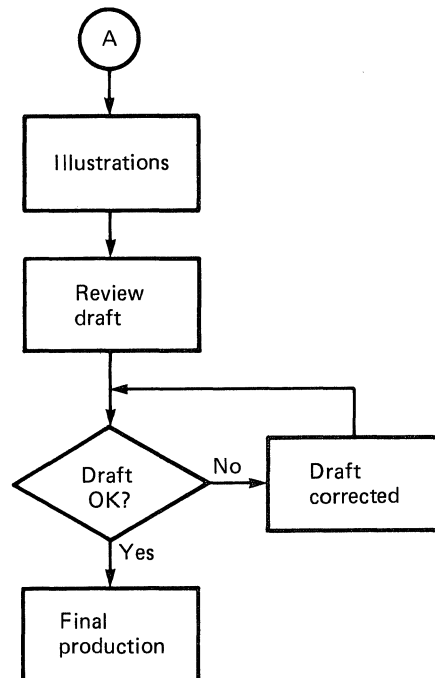nts in documentation development: (1) after the documentation plan; (2) after the first complete draft; (3) after the final draft.

```
              ( Start )
                 │
                 ▼
        ┌──────────────────┐
        │ Documentation    │
        │ need             │
        │ identified       │
        └──────────────────┘
                 │
                 ▼
        ┌──────────────────┐
        │ Documentation    │
        │ plan             │
        │ written          │◄───────────────┐
        └──────────────────┘                │
                 │                           │
                 ▼                           │
              ╱ Plan ╲        No    ┌─────────────┐
             ╱  OK?   ╲────────────►│ Plan        │
             ╲        ╱             │ corrected   │
              ╲      ╱              └─────────────┘
                │ Yes
                ▼
        ┌──────────────────┐
        │ Draft            │
        │ written          │
        └──────────────────┘
                 │                           │
                 ▼                           │
              ╱ Draft ╲       No    ┌─────────────┐
             ╱  OK?    ╲───────────►│ Draft       │
             ╲         ╱            │ corrected   │
              ╲       ╱             └─────────────┘
                │ Yes
                ▼
               ( A )
```

```
        ( A )
          │
          ▼
  ┌───────────────┐
  │ Illustrations │
  └───────────────┘
          │
          ▼
  ┌───────────────┐
  │ Review        │
  │ draft         │
  └───────────────┘
          │      ◄──────────────┐
          ▼                     │
       ╱╲                 ┌───────────────┐
      ╱   ╲      No       │ Draft         │
     ╱ Draft╲─────────────▶│ corrected     │
     ╲  OK? ╱              └───────────────┘
      ╲   ╱
       ╲╱
        │ Yes
        ▼
  ┌───────────────┐
  │ Final         │
  │ production    │
  └───────────────┘
```
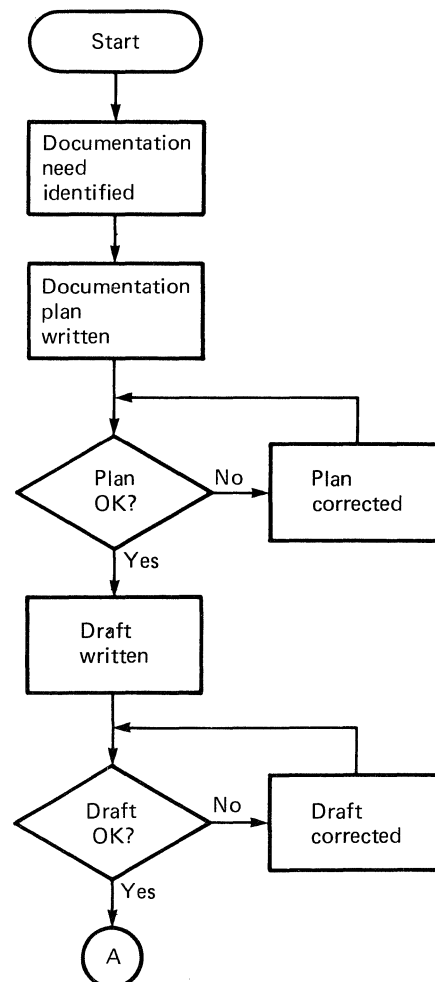
3.1 WHO DOES THE WRITING?

A writing task requires a writer—in this case, a communicator who can analyze market needs and address those needs by presenting the software in a usable way. For example, writing up a software functional specification in a fuller, more descriptive style is not enough if the reader has never opened a computer manual before, and doesn't want to know everything in the functional specification anyhow. So, writers not only translate the software into English, they also restructure information for the person using the application. For example, information about diskette size can be given in terms of printed pages, as well as in more traditional terms. The person who writes the manual for your application has to be a communicator, then. Understanding the product itself is fundamental, but not enough.

3.2 WRITING TASKS

3.2.1 Gathering Information

The software development schedule has to allow time for developers and writers to talk to each other throughout the development cycle. The work environment, the management structure or nonstructure, and even temperament, must all be considered in setting up communication lines between software and documentation.

Software still has to be developed, and there's never enough time. On the other hand, there is no point in producing documentation that is inaccurate because software developers didn't see it as their job to answer writers' questions. Getting answers to questions as the product develops will track software changes and even shape the software. If developers are faced on a regular basis by writers who represent users, they will pay more attention to users' needs.

3.2.2 Writing Information

So you have an ideal situation where a competent person develops software in parallel with a competent person gathering information about the software. And everyone is clear about the project as a result of good planning. However, nobody has done any writing yet.

In fact, that isn't quite true. The documentation plan has an outline of the manual. Keeping as close to this outline as possible, the writer can start writing blocks of information that are possibly disconnected, even within sections. As more software becomes firm, and more information available, these blocks can be enlarged and connected. If the direction changes, because the software changes, or because the writer has a better idea for the structure of the manual, make sure everyone who reviewed the documentation plan reviews the changes. Letting people know about change doesn't eliminate trouble, but it certainly helps avoid it. Showing the various incomplete drafts to developers and other interested people can also help the writer be sure, even before the first "official" draft, that the documentation is on the right path.

Some of the questions that might apply for each feature or for the application as a whole:

- ☐ What is it supposed to do?
- ☐ What happens?
- ☐ What are the defaults?
- ☐ What are the advantages?
- ☐ What are the trade-offs?
- ☐ What should the user be careful of?
- ☐ What other parts of the software are affected?

3.2.3 Illustrations

Whoever is responsible for production will want to know as early as possible how many illustrations the manual will have and if special art or typographic treatment has to be provided.

It is good technique, in any case, to consider what information can be presented graphically rather than in words. An illustration can be clearer, more immediate, and give the reader a break from the written page. An illustration that requires a lot of text, tries to make several points, and needs a lot of explanation has none of these qualities. Make sure that the writer has a clear idea of the intent of a particular illustration, that the style is consonant with the writing style, and that the illustrator understands what is needed. An illustrator capable in the kind of artwork you want can help the writer by simplifying or by suggesting other presentations. The most common problem is for a writer to assume that the illustrator knows what size to make a figure. If figures are thought of in terms of fractions of the page, their size can be judged in a standard way throughout the book. This avoids similar figures being different sizes in the book, and it also prevents a whole page from being wasted by a simple box-like drawing that could more appropriately have occupied a quarter of the page.

When the writer submits draft illustrations for final preparation, the drafts should be in good enough shape to have appeared in the first complete draft of the manual. "Good shape" here means the drafts should be neat enough to read, be appropriate in size, and use upper and lower case consistently.

3.2.4 Examples

Sometimes listing the various ways that a HELP text for example, can be reached is more effective than writing a full, tedious, and perhaps unclear, explanation. If the HELP list is actually a list of usage examples, it serves double duty—once as a mechanism to promote clarity, and once as a reference tool. Good examples fill this dual function. When they don't serve as a reference tool, they are really too trivial to be of help to the reader. And trivial examples serve no purpose except to annoy the reader, who expects better treatment.

The most common, and useful, kind of example in software manuals is the program fragment. Whether the fragment is written by the writer or the developer, it must still be appropriate in the context of the manual's style. So, in a conservatively written book, an example about gas mileage is preferable to an example featuring Miss Piggy. Examples should always be reviewed carefully—not only for accuracy, but also for unwanted whimsy, personal comments, and general eccentricity.

Another reason for checking examples is to see if they can be translated into a foreign language. An example about gas mileage has to specify miles and gallons, which have to be translated into kilometers and liters; but a program example that depends on a play on words is not such an easy matter.

3.2.5 Reviews

Reviewers who are not solely responsible for software may be editors, other writers, or anyone from the various areas that documentation touches.

- ☐ *Technical Reviewers*   Make sure that the documentation describes the features and limitations of the software, and that it is accurate.

- ☐ *Editorial Reviewers*   Make sure that the documentation is understandable, well organized, and appropriate for the audience. Check for consistency on all levels—format, terminology, and style.

- ☐ *User Reviewers*   Make sure that the documentation has all the necessary information presented accurately and helpfully.

In a small group or firm, it is easy to shuttle drafts and sections of drafts back and forth for informal review. However informal the atmosphere, the project has a formal completion date, and documentation has to go along with the schedule that was published in the documentation plan. The first formal review according to that schedule would occur when there is a fairly complete first draft of the manual. Remind everyone involved when a draft review date approaches. Arrange ahead of time for the right number of copies to be made from a master copy that includes illustrations in some form as well. The copies can be regular listings from a printer, which allow space for comments, or more manageable printer listings on letter-size paper. Hard copy can also be obtained by using a photocopier. On-line copies, which don't reflect where changes are made, are not so useful for the writer, who has to compile all the changes.

The second draft review occurs as late as possible in the process, so that the writer has time to incorporate reviewer comments and yet catch the software at its latest developmental point.

Some time before the second draft, software and documentation may be tested in the field. Field test will also require copies of the latest draft. It may be possible to let the field sites make their own hard copy; but this can mean that nobody at the site bothers, so that nobody reads the documentation. If you do in fact field test your product, send a questionnaire with the product asking for feedback. Base the questions on the goals stated in the documentation plan.

When a draft is sent to reviewers, a memo accompanies it explaining what is wanted of a reviewer, when the review is due, and anything special that needs a note. If it seems possible that reviewers will not respond—at all, on time, or appropriately—a review meeting could work, provided that the focus of the meeting remains on documentation and doesn't blur into software design.

3.3 COMMUNICATING

In the description of the writing task at the beginning of this chapter, the emphasis was on structure. Until a writer is ready to put out a complete draft for the first review, structure must remain a major concern. However, what words are used and how they are strung together has to receive more attention as writing progresses.

3.3.1 Vocabulary

"Keeping it simple" seems self-explanatory until examined. There is nothing simple about the word "system", for example, except that it has only two syllables. But this word has to be used in software manuals as a technical term. Technical terms represent technical concepts in a shorthand form, so that a concept need not be expressed in an intolerably long descriptive phrase every time that concept occurs. Maintaining a balance between technical shorthand and simple, everyday language is not an easy task. One way to simplify the task is to compile a terminology list, as soon as there is sufficient text, with definitions for those terms that need them. The list will show what terms have more than one meaning and what terms are so specialized that they need explanation in text. It is also helpful for maintaining consistency of hyphenation and capitalization.

Ambiguous terms and highly specialized terms flourish in a tight-knit group where everyone understands the subject matter and each other. A terminology list prevents proliferation of unacceptable terms and pinpoints problems as they occur. Whoever keeps the list should publish it frequently and make sure that everyone subscribes to the choices made.

So, the difficult task of "keeping it simple" needs an added dimension—elegance. Jargon and forced solutions to language problems create ugly, heavy-footed writing that puts up barriers between the reader and the message. If you have international markets and intend to translate, the clearer the original text, the more accurate (and possible) the translation.

3.3.2 Syntax

A great deal has been written about how to write technical material. However, the rules are no substitute for someone who can communicate. Your documentation could be written according to rule—with only an occasional long sentence, few passives, no adjective pileups before nouns, no unconnected sentences, no unbalanced clauses—and still need help. But it won't need as much help as documentation written hurriedly, randomly, and incompetently.

CHAPTER 4
PRODUCTION

4.1 METHOD

Production is not something to consider after the manual has been written, despite the position of this chapter in this book. Planning for production begins when you start to plan for documentation, because the production method you choose affects the whole documentation schedule and budget. When you make your decision for the documentation plan, you need to choose basically between typesetting or not.

4.1.1 Typesetting

Typesetting gives a more professional appearance to your manual by offering more flexibility in design and better copy for printing. The flexibility is reflected in the many fonts available and in the finer increments possible in vertical spacing. (A font is the total set of characters that make up a particular typographic family in a specific size and in roman, bold, or italic face. For example, Helvetica—used for this text—is a type that has capital letters, lowercase letters, numbers, foreign accents, and various symbols, all of which may occur in different sizes and faces, according to specification). The typesetting process you use may require paste-up of pages from galleys, or it may have automatic paging.

4.1.2 Other

Files can be output to a letter-quality printer, in which case the text must be formatted online and go through a paging pass. You are more constrained with this production method than with typesetting, and the type that is produced is monospaced—that is, each letter and space occupies the same amount of space, unlike typeset material where the line is proportionally spaced.

4.2 DESIGN

The first task in design is to analyze the text for length and for text that may be treated differently from the main text. Lists, notes, glossaries, indexes, and examples are the most likely candidates for special treatment in software manuals. Illustrations also have to be estimated for space and type (line or half tone). The design analysis should occur at first draft time so that you have a basis for a per page cost.

4.2.1 Illustrations

The design problem with illustrations lies in their placement. It has to be made clear during the design process where the illustrations may go in the manual: Must they accompany the relevant text, and if so, how closely? The answer to these questions involve time and money, as well as readability.

4.2.2 Paper

The weight and texture of the paper you use to print the manual makes a difference in ease of reading and in the way that illustrations reproduce—particularly half tones. Consult with printers about the paper suitable for your manual, and know how much to budget for it well before it's time to actually print.

4.2.3 Typographic Specification

Once the designer (in an ideal documentation team) has analyzed the book, the type spec can be written. Whatever the production method you choose, you will find this exercise helpful. For typesetting, the spec is more complex. The specification for the Professional 300 Series end user documentation appears in Appendix A. If you need to, you can look up the technical terms used in the spec in any book on bookmaking. By noting what has been covered in the spec, you can see what needs to be decided to make a book design. This is not intended to show you how to design a book, just what is involved.

4.2.4 Sample Pages

The type spec should really have a dry run before you decide on the final design. This applies to tapes and floppies, or formatting routines too. Try them out on the vendor's equipment so that problems can be ironed out early, and schedules and budgets made clear.

Certain pages of the manual have to be chosen as representative. They make up a simulated chapter, with various head levels, a table, a figure, and anything else that occurs systematically throughout the book. The pages are then typeset, or produced in some final form. Two or three pages are usually enough for a simple book; six to eight may be necessary for one with complicated design problems. This is the time to change the design, not later. Sample pages can also be used to encourage the group working on the project and to further the group's image within the firm.

4.3 DRAFTS

The writing task continues while the design is being established. But after the final review draft has been approved, changes must be halted. One person should be responsible for maintaining a master copy of the final review draft. Any changes—the typographical errors discovered by the proofreader, or the rewrite of Chapter 4 to reflect a last-minute software fix—must be controlled. Otherwise, the copy that goes to print is not the same as the copy that you had before final production, which means revisions and updates become time-consuming processes.

4.4 MARKUP

Even when you are dealing with computer files, it is still a good idea to provide a hard copy listing of the files, so that whoever is producing final output copy can compare it with input. The accompanying type spec indicates heads and special parts of the text with numbers and letters. The final draft is keyed to the type spec with the same numbers and letters. This keying of the hard copy is called "markup". The word "markup" is also used to refer to the process by which a data entry person marks up copy with code before the code is implemented online. Keying eliminates questions and saves time in the long run. The symbols used must, however, be mutually understandable.

4.5 REPRODUCIBLE COPY

The typesetting process results in galleys. Whatever output you have at this point is the final copy that will be reproduced, that is, printed. If you have galleys, and they are not paged, then indications for illustration placement can be made now. Otherwise, this must be done earlier. Galleys must be checked against the marked up copy that accompanied the draft to the typesetter. Allow time for corrections to be made and rechecked. The corrected galleys are then pasted up into pages which also have to be checked. When galleys are sent for paging, make sure that running heads or feet are specified. (Running heads or feet are the small heads that run outside the type area at the very top or the very bottom of the page. They tell you the part, the chapter, or the book title, and occur with the page number.)

4.6 PARTS OF THE MANUAL

While galleys are being set, or while reproducible copy is being produced, the contents and index can be compiled. Check the list that follows for what should be in the final book. The printer will need an assembly sheet of what pages are in the book and where the blank pages occur. Arrange with the printer to give you a sample assembly sheet ahead of time.

Title page	(right)
Copyright page	(left)
Contents	(right)
Preface	(right)
Text	Divided into numbered chapters, each beginning on a right-hand page; numbered parts are also possible.
Appendix	(right)
Glossary	(right)
Index	(right)

4.7 COVER

The cover is where documentors like to begin. It advertises the book to the world and sets expectations. But for these very reasons, covers are best left to experts, and only begun after there is a complete draft of the manual. Even if you decide to go with a cover that only has type, and no graphics, it should be handled by a graphics expert. It will cost very little and reap great benefits. Anything besides type on the cover should be approached with care. Make sure that it's appropriate and attractive.

Before you go to the printer, make sure that you have a checklist that has been made after talking to the printer about what should be in the package. All the pages need to be accounted for, all the art, and anything special, such as a color markup package. Production is a matter of checklists. Make them in good time, use them, and keep things organized. Your handsome cover can then go on a well-produced book that is not only accurate but elegant.

Appendix A
Professional 300 Series Specification
End User Documentation

trim	7 × 9 inches
type area	30 × 47 picas (including running head + foot folio)
text area	30 × 42 picas (42 lines per page)
margins	gutter: 1-1/8" head: 1/2"
ink colors	black, PMS 199, PMS 877
basal type	10/12 Century Expanded set on track 2 × 30 picas, justified; no paragraph indent, 24 pts bb between paragraphs. Italic and bold as emphasis with text per ms.

CHAPTER OPENINGS

New right-hand page; background bleed PMS 199

Run three 14 pt drop-out rules; bleeding across the page from gutter to fore, with 6 pt space between rules, with the first rule hanging 6 pi, 3 pts from the top of the page (paper).

CN   *Chapter number*   60 pts high Avant Garde Book Arabic number indented 8 picas + 3 points from fl left of text area.

CT   *Chapter title*   24/27 Avant Garde Book C/lc fl left with CN; set × 22 picas maximum width.

The chapter number sits 25 picas from the top trim, hold 9 pts visual space to a 1 point rule fl left with CT, bleeding fore; 27 points base to base (bb) to chapter title. Print all type black. No page number.

Right page, following chapter openings:

The first line of the text for the chapter sits 22 picas from the top of the text area. Use drop folio with no running head.

Chapter number   16 pts Avant Garde Book C/lc, fl left × 30; with word "Chapter" spelled out with an arabic number. Hold 9 pts visual to 1 pt rule × 30, 9 pts visual after rule to chapter title.

Chapter title   16/18 Avant Garde Book C/lc, fl left. 36 pts b/b to first line of text. If title is long, wrap fl left, no hyphenation.

TEXT ELEMENTS

	folio	9 pt Helvetica Medium fl outside × 30, sits 24 pts bb below normal text depth.
	running head	7 pt Helvetica Regular caps fl outside × 30; <u>left</u>: word "CHAPTER" and its number, 3 pt#, vertical solidus, 3 pt#, chapter title. <u>right</u>: most recent "1" head.
1	subhead	9/12 Helvetica Medium caps, fl left × 30, color, 32 pts bb above, 20 pts bb below.
2	2nd subhead	9/12 Helvetica Medium C/lc, fl left × 30, 26 pts bb above, 18 pts bb below.
3	3rd subhead	9/12 Helvetica Medium Italic; C/lc, fl left × 30, 24 pts bb below, 18 pt bb below.
4	4th subhead	9/12 Helvetica Italic, C/lc, fl left × 30, 24 pts bb above, 18 pts bb below.

LISTS

LH	list head	10/12 Helvetica Regular C/lc, fl at 3 pica indent, 24 pts bb above, 28 pts bb below to first entry.
EH	entry head	10/12 Helvetica Regular italic init cap + lower case, fl at 3 pica indent, run-in + en dash to entry text.

If lists are multi-columned, hold 2 em# between columns.

UNL	unnumbered	10/12 Century Expanded, fl at 3-pica indent, ragged right with open square bullets fl left at 2 pica indent at size shown on layouts; 24 pts bb above and below, 18 pts bb between entries and between paragraphs within entries.
NL	numbered	Same as UNL with numbers aligning on the period which is fl left at 2 pica indent; same spacing. RR.
NL/DIS	list	Set as DIS or CDIS.

LISTS (Cont.)

L/CDIS	displays	
DIS	displays	9/12 Helvetica Regular set track 1, fl at 3 pica indent; 15 pts bb above and below, 15 pts between groups of entries or as marked on manuscript.
CDIS	computer displays	9/12 Helvetica (track 1) at 3 pica indent, color as marked on script; (indicates what user types); 18 pts bb above, below and between groups of entries.
		Menus may appear as CDIS when they are not treated as figures; Set as CDIS with menu title 9/12 Helvetica. Bold caps, fl at 7 1/2 pica indent. Use two levels of hanging indent for text, per ms: fl left of menu in manuscript = 3 pica text indent; indent of menu in manuscript = 7 1/2 pica text indent. Use 9/12 Helvetica. Caps for text; 15 pts bb above and 18 pts below, 18 pts bb as spacing between elements within the menu.
FTN	footnotes	7/9 Century Expanded, fl left × 30, ragged right, 12 pts bb above to 1/2 pt rule × 30; allow 12 pts visual space above rule from basal text. 12 pts bb between consecutive footnotes.
N	note, "caution"	Set the word "note" or "caution" or "warning" 8 pt Century Expanded Bold caps followed by a colon and an en space. Run into text. Set text 8/10 Century Expanded Bold fl left × 24 picas, ragged right; position fl at 3 pica indent, 24 pts minimum above and below, 20 m text. Print warnings in color (red).
SH	side head	*head*: 9/12 Helvetica C/lc, fl left of text area × 9 picas, ragged right, no hyphenation + 1-pica gutter to hanging 10-pica indent.
		text: 10/12 Century Expanded × 20 ragged right position fl at 10-pica indent, 24 pts bb above first entry (head); 18 pts bb between entries (heads) and between paragraphs with entries, 24 pts bb below last entry (head).
SH/DIS	side head	Set as DIS or CDIS but position fl with side head text, hanging at 10-pica indent.

LISTS (Cont.)

	keys	Set 8 pt Helvetica caps, base aligned with surrounding text. Enclose per ms in 1/2 pt rounded corner box: print box PMS 199. Box clears sides of enclosed word(s) by 3 pts and is 11 pts high. Normal word spacing and leading surrounds box to other text.
A	section head no tab	14 pt Helvetica Medium, C/lc, fl outside × 30 picas left and right page. New page each section head, or as marked on manuscript 24 pts bb below head to text.
B	chapter section head no tab	*head*: 14/15 Helvetica Medium C/lc, fl left × 30 picas; preceded by a 1/8 inch rule fl left with text area and bleeding fore; minimum 24 pts visual space above the rule; 18 pts bb below rule to head; 24 pts bb below head to text.

tab: 7/9 Helvetica Regular caps, fl left × 3 picas, ragged right, no hyphenation. Position so last line sits 6 pts above 1/8" rule, 1 pica outside the 30 pica text area (clears page trim × 18 pts)

Print rule, tab, and head PMS 199.

TABLES   All set × 30 pica

TN	table number	8/10 Helvetica Medium C/lc, fl left × 30
TT	table title	8/10 Helvetica Medium caps, fl left × 30, ragged right on line below TN; no added space above.
TI	Table column head	8/10 Helvetica Medium Italic, C/lc, fl left over columns.
Tb		9/10 Helvetica Regular init cap + lower case or per manuscript; ragged right with hyphenation;
table text		16 pts bb between entries or groups of entries per manuscript.

Allow equal space between columns, minimum 1 pica.

TABLES (Cont.)

Table rules (print PMS 199): top rule (below TT and above T1): 2 point rule × 30 picas, all other rules (below T1 and below last entry of Tb) are 3/4 point × 30 picas; hold 6 pts space above rules, 15 pts bb below rules.

ART

Text art may be drawn up to 30 picas wide, art for text accompanying side heads should be drawn up to 20 picas wide; art for lists, displays and other material at a 3-pica indent should be drawn up to 27 picas wide. All art hangs fl left with the corresponding material. Art for side art should be drawn up to 14 picas wide.

FN	figure number	8/10 Helvetica Medium C/lc, fl left with figure.
FL	figure legend	8/10 Helvetica Medium init cap + lower case, fl left × width of figure on line below FN. No added space above.
	labels	6 pt/8 Helvetica caps.
	Position FN art to text	15 pts bb below art; allow maximum 36 pts space above or below FL to text.
	Position FN art to text	15 pts bb below art; allow minimum 36 pts space above or below FL to text.
SA	side art	Text accompanying side art sets 10/12 Century Expanded × 14 fl left, ragged right. Position art × 14 picas wide fl left text area + 2 pica gutter to accompanying text. Hang art to align with top of lower case of first line of accompanying text; 24 pts bb above first piece of art between entries and after last piece of art in the section.
	keyboards	*American:* (full page, turn) Enclose text area 30 × 42 picas in a 1/2 pt ruled box. All type and art is turned facing fore.
		head: 10/11 Helvetica Medium caps fl with keyboard, 18 pts bb below top rule of box. Keyboard (× 40 picas long) hangs 24 pts below base of head.
		text: 9/11 Helvetica Regular × 40 ragged right, first line sits 18 pts bb below keyboard; 18 pts bb between paragraphs.
		Foreign language: (2 per page) Run 3 1/2 point rules × 30 picas; at top of text, 21 picas from top of text, at bottom of text.

ART (Cont.)

head: same as head, American; 18 pts bb below top rule. Keyboard (×30 picas long) hangs 24 pts below base of lead.

text: 9/11 Helvetica Regular × 30 ragged right,first line sits 18 pts bb below keyboard, 18 pts bb between paragraphs.

PAGING INSTRUCTIONS

Pages may run normal depth, or one line short in spreads.

If possible, when they do not follow text directly, position art and tables at the top of the page.

When a "2" head follows a "1" head directly, allow only 24 pts bb above the "2" head. When a "3" head follows a "2" head directly, allow only 18 pts bb above the "3" head.

When a "1" or "2" head fall at the page bottom, at least 3 lines of text must follow; when a "3" head falls at the page bottom, at least 2 lines of text must follow.

When a "1", "2", or "3" head falls at the top of the page, allow no In adjusting space around elements to fill a page, adjust as equally as possible.

FRONT/END MATTER

Cover	background bleed	PMS 877 silver
	product identity rules:	PMS 199 red
	type:	black

Copyright Information

Set a 1 pt. rule, 9 pts visual to first line of text. 8/10 Century Expanded × 30 ragged right. Build up from bottom of text area; use 16 pts bb between elements or paragraphs per ms.

Contents

Word "Contents": Set 24 pt Avant Garde Book C/lc, indented 8 picas + 3 pts from fl left of text area, sitting 18 picas + 9 pts below top of text. Hold 9 pts visual to a point rule, fl left with word "Contents", justified with the text area; 27 pts bb below to first chapter title.

Chapter numbers and titles: 10/10 Helvetica Medium caps, with word "CHAP-TER" and its Arabic number fl left × 30 and the chapter title indented 8 picas + 3 pts from fl left; no page numbers. Turns fl with word above; 6 pts visual below chapter title to a 1/2 pt rule × 30 picas.

1 head: 10/12 Century Expanded, init cap and lower case, fl left × 30, clearing 2 ems to fl right page numbers. Turns fl with word above.

2 heads: 10/12 Century Expanded, init cap and lower case, indented 18 pts from fl left × 30 + en# to run-in page numbers. Run ragged right to width which clears the 1 heads' page numbers × 2 ems. Turns fl with word above.

Allow 15 pts bb below chapter rule to first 1 head, 15 pts bb between 1 head, no added space around 2 heads, 36 pts bb above new chapter title.

Index

Opening and head follow style of Contents.

Text: Set 8/9 Century Expanded × 14 + 2 + 14, ragged right, with subentries indented 1 em and all turns indented 2 ems. Leave 18 pts as divisions of the alphabet.

FOR USER'S GUIDE – HARD DISK SYSTEM

OP   Additional Options

       Run a 3/4 pt rule × 24 picas.
       Indent 3 picas left and right.

       Hold 6 pts visual from rule to head.

       Set head as 4th level head, 9/12 Helvetica Italic, C/lc.
       Indent 3 picas.

       Set text 10/12 Century Expanded × 24.
       Indent 3 picas left and right. Ragged right.
       Hold 6 pts visual to 3/4 pt horizontal rule.

       18 pts bb above and 24 pts bb below.

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized?
Please make suggestions for improvement.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Did you find errors in this manual? If so, specify the error and the page number.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Please indicate the type of reader that you most nearly represent.
☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Other (please specify) _____

Name _____ Date _____
Organization _____
Street _____
City _____ State _____ Zip Code _____
or
Country

d i g i t a l

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Professional 300 Series Publications
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET
MAYNARD, MASSACHUSETTS 01754