1-1-1991

Designing effective computer documentation for end-line users

Tracy Jean Montenguise
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Recommended Citation
Montenguise, Tracy Jean, "Designing effective computer documentation for end-line users" (1991). Retrospective Theses and Dissertations. 17602.
https://lib.dr.iastate.edu/rtd/17602

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Designing effective computer documentation
for end-line users

by

Tracy Jean Montenguise

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF ARTS

Department: English
Major: English (Business and Technical Communication)

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa
1991
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER I. INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance of the Study</td>
<td>1</td>
</tr>
<tr>
<td>Questions of the Study</td>
<td>4</td>
</tr>
<tr>
<td>Assumptions of Study</td>
<td>4</td>
</tr>
<tr>
<td>Limitations of the Study</td>
<td>5</td>
</tr>
<tr>
<td>Definitions of Terms</td>
<td>5</td>
</tr>
<tr>
<td>Plans of the Study</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER II. REVIEW OF LITERATURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation Organization</td>
<td>9</td>
</tr>
<tr>
<td>Documentation Format and Style</td>
<td>10</td>
</tr>
<tr>
<td>Documentation Layout</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER III. SOFTWARE DOCUMENTATION PROCESS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style Sheet Development</td>
<td>29</td>
</tr>
<tr>
<td>Software Analysis</td>
<td>29</td>
</tr>
<tr>
<td>Documentation Design</td>
<td>33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER IV. CONCLUSIONS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFERENCES</td>
<td>38</td>
</tr>
<tr>
<td>WORKS CONSULTED</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPENDIX A. GENERAL STYLE SHEET FOR LOW-BUDGET COMPUTER DOCUMENTATION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX B. INTERVIEW QUESTIONS</td>
<td>52</td>
</tr>
<tr>
<td>APPENDIX C. SPECIFIC STYLE SHEET FOR THE B-STAT MANUAL</td>
<td>58</td>
</tr>
<tr>
<td>APPENDIX D. B-STAT COMPUTER SOFTWARE MANUAL</td>
<td>60</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENT

I would like to give a special thanks to Gerald Burrell for letting me document his software package and Dr. Carol David for her support and editing advice. I would also like to thank my committee members Dr. Don Payne and Dr. Roger Berger.
CHAPTER I. INTRODUCTION

Computers are increasingly occupying more space in our professional and personal lives, and the computer software market is booming as hundreds of new programs are marketed each day to help perform people's everyday tasks. Even individuals and small companies are attempting to break into the market by designing software packages to advertise on computer shared networks and shopping services, such as shareware and CompuServe. Thus, as the market for computers and software packages expands and becomes more competitive, computer manufacturers and software developers have begun to take an in-depth look at the one item which links buyers to their products, influences customer satisfaction, and affects the sales of computer software packages: software documentation.

Today researchers are finding that well-written and well-designed documentation is the key to successfully marketing computers and computer software packages. In fact, researchers believe that effective software documentation may be second in importance (the first being quality) when people consider buying a software package (see Gervickas), and software companies are now realizing that the quality of their user manuals is "critical to the commercial success of their products" (Mirel et al. RT-170). Carolyn Mullins even claims that a "literate, well-
organized user manual" is essential to a software package, and consumers can tell how good a software package is by looking at its documentation: "If a manual holds up under scrutiny, chances are the program will be of equal--or higher--quality" (44). Thus, documentation can sell or bury a software package. This finding is especially important to the individual who is trying to break into the software market. Besides attracting buyers, effective documentation also attracts other software developers and companies who may wish to buy the rights of an individual's program and then expand on it. Yet even with all this emphasis on computer documentation, a recent poll by the magazine Which Computer? reveals that fifty percent of its readers still find the documentation accompanying computer software packages to be the least satisfactory item in the total computer system ("Manual" 49). G. Prentice Hastings and Kathryn King also state that a recent survey reveals over eighty-five percent of all computer documentation offends the user and fails to instruct or inform (1). Even system specialists and technical writers in the computer field seem to be struggling with the fact that computer software documentation is still, on the average, well below par.

Poorly written computer software documentation is most likely due to a lack of skilled writers who specialize in developing computer documentation or a lack of accessible,
easy-to-understand information on how to produce effective
documentation. Although an abundance of research exists on
computer documentation, most software developers do not have
the time to sift through all the existing resources to
compare and contrast findings on text layout, organization,
and format in order to compile a basic documentation style
guide to follow. Also, most of the articles and research on
computer documentation are designed for big companies, or
those who can afford to hire a writing team and employ
sophisticated graphics in the documentation’s production.
Therefore, the individual software developer entering the
market or a newly formed company on a limited budget is left
without the needed practical guides to develop effective
documentation--both manual and online--that would make their
products more competitive.

Thus, the purpose of this paper is to provide basic
style guides which writers can use to develop low-budget
computer documentation. To achieve this purpose, I
condensed recent research on documentation writing and
developed a basic style guide for low-budget computer
documentation. I then used a modified version of this style
guide (limitations of the software that I was documenting
required certain modifications, such as eliminating icons)
to design a low-budget manual for a basketball statistics
software package called B-Stat, written by Gerald Burrell, a
The documentation I designed will be included as part of the B-Stat software package, which was developed for use in high schools and junior colleges. The package will be marketed in the fall of 1991 through computer shareware networks and direct mailings.

In this study the term documentation will be used to denote both manuals and online help screens. When I have found a difference in style guidelines to exist, I will note the difference; otherwise the guidelines and research discussed are applicable to both.

Significance of the Study

Besides adding to the research on computer documentation, this study provides a model of the computer documentation process which readers of this study can adapt in developing their own computer documentation, instructions, or other types of manuals.

Questions of the Study

This study will attempt to answer the following questions:

1. What basic methods or processes does one use to develop computer documentation?
2. What are the basic guidelines for text organization, format, and layout that one should
follow to design effective computer documentation manuals?

3. What are the basic guidelines for text organization, format, and layout that one should follow to design computer online help screens?

Assumptions of the Study

Because this study provides information for writers of computer documentation, two basic assumptions were made. First, I assumed that the users of this study would have a working knowledge of basic word processing skills and access to the appropriate software and printer needed to produce basic formatting features, such as layouts, headers, and footers, along with basic textual variations, such as type sizes, bolding, italics, and a variety of other fonts. Secondly, I assumed that a reader of this study would be writing documentation to instruct people on how to use a software package, not necessarily to explain how the package was developed or to describe its technical format; furthermore, the package would have similar budget and software capabilities as the one designed in the study.

Limitations of the Study

Although the documentation style sheet developed by this study can be applied as a basic writing foundation for any instructional or explanatory manual, it is especially
designed for developing computer software documentation with limited funds and software capabilities. Thus, the following limitations exist:

1. The budget of the software developer does not allow for multiple color or illustrations in the manual or online help screen.

2. The software package being documented does not allow for icons in the online help screen.

3. The software package being documented does not allow for printouts of pictures of the screen during the different stages of the software’s use.

4. The online help screen in this software package can be activated only when the user is not performing actual data entry or manipulation.

5. The online help screen in this software package is one complete unit; thus, the user does not have the capability of retrieving the help units by indexing specific procedures with cued commands or codes.

Definitions of Terms

End-line user - The people who use the software in the way it was intended to be used. For example, the end-line users of a software game package would be the people who play the game for entertainment, not the people who play the game to analyze the logic or write the documentation.
**Documentation** - "The written communication method for transferring and recording ideas, plans, details, methods, instructions, etc." of the software (Katzin 1).

**Minimalization** - A practice of using minimal explanations, especially minimal text and steps, to explain a procedure or instruction.

**Online documentation** - Any documentation presented on the actual computer screen.

**Product-oriented** - A phrase used to describe documentation that explains in detail what the software can do and how it is designed, instead of explaining how the software is used.

**Readability** - A term used to describe how easily the instructions can be followed, in other words how easy the text is to understand.

**Task-oriented** - A phrase used to describe documentation that explains itself to the end-line user by detailing the steps of the tasks which the user wants to perform.

**User-oriented** - A phrase used to describe documentation that is written from the perspective of the end-line user. In other words, the documentation is designed based on the end-line user’s language, experiences, and goals.

**Plans of the Study**

The following three chapters document the study.

Chapter II, Review of Literature, discusses the recent research conducted on computer documentation. This chapter
is divided into three subdivisions of the documentation process: methods, format, and layout. The discussion of methods includes the overall planning and task orientation of the documentation. The format discussion includes issues on text style and cueing devices. Finally, layout covers the overall page design by describing spacing and graphics.

Chapter III, Software Documentation Process, describes the methods which I employed to produce the computer documentation for the software package B-Stat. This chapter is divided into three steps: style sheet development, software analysis, and documentation design. In the first step I used research findings to develop a general style sheet with guidelines for designing low-budget manuals. The second step, the software analysis, discusses the actual steps I took to understand what B-Stat can do. Finally, the design analysis describes how I achieved the overall look and tone of the B-Stat software manual.

Chapter IV, Conclusions, summarizes the results and provides concluding remarks on this study. The results pertain particularly to the style sheet on documentation guidelines, while the concluding remarks discuss the manual and considerations with its production.
CHAPTER II. REVIEW OF LITERATURE

Much research has been conducted on effective computer documentation by academicians, practitioners, and engineers. The following review of literature categorizes this research into three main sections: documentation organization, format, and layout.

The first main section of the literature review, documentation organization, describes research pertaining to how one should organize the actual documentation writing process and what steps should be included in this process. This section is divided into two subdivisions: overall steps and task orientation. The overall steps division gives researchers' views of the steps one should include when producing computer documentation. The subdivision on task orientation discusses research on determining what perspective should be used when organizing the documentation.

The second main section of the literature review, documentation format, discusses research on how to design the text of the documentation. This section is divided into two subdivisions: textual considerations and cueing devices. Textual considerations discusses research on how to format the text on the page and the styles that should be used for the text and sentence development. Cueing devices
explains research on how to employ techniques to make the text easier to read.

The third main section of the literature review, documentation layout, describes research findings on the page design of computer documentation. This section is divided into spacing and graphics. The subdivision on spacing describes the research on page layout and design considerations. The graphics subdivision discusses research on the implementation of graphic devices into text.

Documentation Organization

Most documentation experts and researchers agree that every production of documentation begins with planning. Documentation writers must organize a plan of development for the document before writing any part of the documentation. They have to know the features of the software, the characteristics of its end-line users, and the intent of the developer--funds, equipment, expectations.

Overall planning

Although they may have developed different names for the production processes, most researchers agree that documentation production should include the following basic steps: (1) talking to the writer of the program--assuming the documentation writer is not the software writer, (2) analyzing the end-line users of the software package,
working with the program, designing the documentation, writing the documentation, testing the documentation, and editing the documentation. Susan Fowler and David Roeger; Patricia Williams and Pamela Beason; and Susan Grimm provide three different examples of the documentation process.

Fowler and Roeger categorize the documentation process with three steps: (1) "collecting information," (2) "developing a structure," and (3) "the review process" (23-24). According to Fowler and Roeger, collecting information, the first step, involves finding out the intent of the software and the characteristics of its end-line users. In this step, they explain that documentation writers need to ask the software author questions, use the program that they’re trying to document, and talk to the customers of the software package (23). Fowler and Roeger claim that talking to the author of the software is crucial in determining what the software is supposed to do and what limitations the software and developer have, such as funding, screen capabilities, etc. They state that the writer should "make it clear to the programmer that, until the manual is finished, the two of you [documentation writer and software developer] will be having an ongoing relationship" (23). Fowler and Roeger further advise that writers ask the software developer about his/her
expectations for the documentation. They also feel that the
writer needs to know the software in order to describe it
and that using the software is important: "Try every option
on every menu [of the software package]; print every report;
look at every screen; read every processing table (use the
debugger if the system has one). Do things wrong and see
what happens" (23). Finally, in this initial step, the
documentation writer needs to talk to potential customers of
the software to find out how to improve the software or any
existing documentation.

Before writing the documentation, Fowler and Roeger
believe that a writer should develop the structure of the
documentation--their second main step--by developing a table
of contents which can be used both as an outline and as an
aid in critiquing the documentation.

Finally, Fowler and Roeger's documentation process
involves the third main step: reviewing the document. In
this step the documentation writer should provide drafts of
the documentation to the programmers, users, and any other
people involved with the software. This step gives the
writer the opportunity to receive editing and design
recommendations for the document according to end-line
users' and software developers' suggestions.

Williams and Beason suggest three main organization
phases for documentation writers: (1) planning,
(2) outlining and writing, and (3) production. Williams and Beason’s planning stage is similar to Fowler and Roeger’s collecting information stage; thus it involves analyzing the software and its end-line users by conferring with programmers and experimenting with the software. Williams and Beason subdivide this step into "describing the project," "locating existing information," "defining documentation goals," and "writing an audience profile" (16-18). This information collected from these subdivisions includes the purpose of the software, features of the software, the hardware involved, and the objectives of the documentation (16-17). The authors suggest actually reading the programmer’s notes and functional specifications (16). Williams and Beason do explain the end-line user analysis in greater depth than Fowler and Roeger by dividing the software’s audience into primary and secondary users. They advise describing each of these groups by including "any information that affects the users' responses to your documentation, such as level of computer expertise, occupation . . . level of education, age group, and reasons for using the program" (19). The planning stage also involves actually writing down the desired physical characteristics and design of the document.

Williams and Beason’s second main production step, the outlining and writing phase, involves the development of
rough drafts, the review of those drafts, and the incorporation of accepted changes. Thus, this stage involves decisions on layout, textual format, and illustrations.

Williams and Beason’s final stage, production phase, involves producing the final text--along with indexes, title pages, and a table of contents--and printing the documentation in its final form.

Like Fowler and Roeger and Williams and Beason, Grimm divides the documentation production process into steps, but her process has four steps: (1) before writing, (2) during writing, (3) after writing, and (4) after the manual is finished. The first three steps of Grimm’s process are basically the same as the steps in the Fowler-Roeger and the Williams-Beason processes. Grimm’s first step, before writing, involves obtaining background information (understanding the program and its users) and determining the design (format and layout) of the manual. Grimm’s second step, during writing, occurs during the writing process and involves producing the text and all extra pages--appendixes, table of contents, indexes, etc--by using suggested style and grammar rules. Grimm’s third main step, after writing, occurs after the writing is done and involves editing, proofreading, and producing the final form of the documentation. Finally, unlike the Fowler-Roeger and the
Williams-Beason models, Grimm adds a fourth step, after the manual is finished. Her fourth step involves using the finished manual to conduct training centers and establishing up-to-date procedures for the documentation package which can be added later to customers’ manuals.

Task orientation

Historically, computer documentation has been written from a product-oriented perspective—the technical capabilities, such as memory use, that the software has, but researchers are now finding that documentation should be organized from a task-oriented perspective, instead. In fact, all preliminary software and audience analyses should be conducted from the view of the tasks that the user will perform with the software. Studies conducted by Bowman, Partridge, and Odescalchi support these findings.

Bowman conducted a study on task orientation by applying a ranking of task orientation to an existing user guide (described in Gervickas). After asking reviewers to comment on the revised guide, he concluded that information should be task oriented, but he also found that this method should not be carried so far as to eliminate necessary concepts or definitions that readers may benefit from knowing (Gervickas 216).

Partridge conducted a survey of technical writers, managers, and editors to see how effective task orientation
is in the documentation field. He asked these technical professionals how they organize documentation. He found that 38.5 percent of those who replied organize documentation according to tasks, and 61.9 percent believe that task orientation is the best strategy for document organization (28). Sixty-two percent of the respondents also felt that using task orientation by itself was a better strategy than combining it with some other documentation method. Partridge concluded that if a "reference manual is not organized by tasks, the user will find it much more difficult to learn how to use the software" (29).

The purpose of Odescalchi’s study was to "compare the productivity of people using both task-oriented information and product- or function-oriented information" (16). She gave twenty-six subjects three assembly projects to complete using instructions based on the two different methods of documentation. Odescalchi found that the subjects had much higher rates of dissatisfaction, difficulty, and failure using the product-oriented information. She states that the "type of documentation used by the individuals was the determining factor in the error rate. Clear step-by-step information dramatically reduced the error rate" (17). Her results further show that the number of errors made by those subjects who followed the product-oriented instructions were 480 percent higher. Odescalchi concludes by stating that
task-oriented information results in a forty-one percent gain in productivity of the readers.

Documentation Format and Style

Designing effective format and style for documentation is perhaps one of the most researched areas in writing. Researchers from a variety of academic fields, such as psychology, linguistics, technical writing, and business, have investigated how a document’s format and style influence—both negatively and positively—readers’ responses to the document. The researchers’ main goal is to find what format and style make the documentation the most readable, or what format and style work best to help readers understand and follow directions. Research on documentation format can be divided into two main areas: textual considerations and cueing devices.

Textual considerations

Textual considerations involve the physical characteristics—size, color, font, etc.—of the words and symbols used in the text of the documentation and how these are arranged together to form sentences, paragraphs, lists, etc.

Typography Researchers have found that the size and type of a document’s text influences the readability, and therefore the success of the document. All researchers
agree that the most readable type sizes are between 8 and 12 points (see "Simply Stated"; Gribbons; Benson; Debs). Benson also suggests to "use between 2 and 4 points of leading [vertical spacing] between lines of text" (37). Of course, many word processing software packages automatically set the leading.

The type of the text involves the typeface and additional characteristics, such as boldness, italics, capitalizations, underlining, shadowing, etc. Researchers have found that using the same typeface adds consistency to the document and increases readability. Gribbons states that "when organization is a primary objective, it is usually best to confer typeface selections within a single type family" (VC-36). Researchers have also found that serif typefaces are easier and faster for people to read (see Benson). Finally, research suggests that writers should limit the use of capitalization, bolding, underlining, italics, shadowing, and other style effects, since they make the text more difficult to read (see "Simply Stated"; Gribbons; Hartley "Eighty"). Gribbons claims that uppercase lettering "retards reading speed by as much as 12%, that it requires 35% more room, and that readers prefer the use of lowercase printing" (VC-37). Benson advises to avoid using all capital letters whenever possible, since "each lowercase word has a distinct outline that aids
recognition and recall of the word" (37). Gribbons and Benson both suggest that underlining is an outdated technique, and they agree that text sizes can be varied consistently to emphasize different levels of headings.

Research suggests that online documentation should follow these guidelines for text size and style whenever possible. For example, Hofmeister suggests to use screen fonts "similar to paper fonts to increase reading speed" (RT-56). However, because of screen size limitations, writers might have to vary the size guidelines to produce readable online text (Brockmann).

Lines Research shows that the best line length for text--including online--is 50-70 characters and that text should be left justified ("Simply Stated; Hofmeister; Hysell). Also, text--even for online--should be arranged vertically and in a single column: Hofmeister states that "two column formats for online text can retard reading speed by as much as 200 words per minute" (RT-56).

Language and tone All researchers agree that the language and the tone of the documentation must be user-friendly and that writers can achieve this tone by avoiding jargon and using simple, familiar, and conversational language. Rubens determined which style of writing people prefer by using three different computer tutorials with three writing styles: formal, moderately friendly
After surveying 123 undergraduate students with varying computer experience, Rubens concluded that all students, even those with extensive computer skills, prefer documentation with a conversational tone.

Sentences Most researchers agree that sentences in documentation should adhere to standards of minimalization; they should be short and to the point. Unnecessary adjectives, adverbs, and phrases should be omitted (Katzin). Hartley suggests that good sentences are less than twenty words long ("Eighty" 18). Researchers also agree that sentences in documentation should be written in the active voice and instructions should be in the command form to speak directly to users (see Grimm; Katzin; Debs; Hartley Designing and "Eighty"). Grimm also claims that active voice is more effective than passive, because it is simpler, shorter, more interesting to the reader, and defines responsibilities for the reader. Finally, researchers have found that sentences using positive terms instead of negative qualifications are easier to understand and make the reader feel more comfortable (see Hartley "Eighty").

Paragraphs Research shows that paragraphs in documentation should be short, six lines or less, and separated by at least one blank line (Hysell; Hartley "Eighty"; Katzin). Not all researchers agree on whether
paragraphs should be indented; for example, Hartley suggests that unindented paragraphs, when short, are easier to read, while Hysell suggests that all paragraphs in documentation should be indented.

**Punctuation**  
Research shows that normal punctuation techniques should be used in documentation writing but that writers should avoid excessive punctuation, since it slows the reader and impedes comprehension. For example, Grimm suggests using commas and semicolons sparingly and avoiding quotation marks, since they clutter the page and distract the readers.

**Lists or steps**  
Since documentation is designed to be task-oriented, researchers suggest developing procedures or instructions through lists or steps of tasks. Lists also keep the text short and simple (Hysell). Kujawski advises three ways to keep lists or steps of instructions simple: "(a) number instruction in sequence; (b) let each instruction require only one action; [and] (c) use simple words" (108). Kujawski further states that instructions should be "one-way directions," avoiding items in reverse order of their execution (108).

**Color**  
Research has shown that using color for the text can enhance the overall appearance of the document, but its use requires some precautions. Brockmann discusses a study done by Tinker which found that "black print on a
white background is most preferred, followed in decreasing
order by green on white, blue on yellow, black on yellow,
and red on yellow" (77-79). For online color
considerations, Brockmann warns against using sounds and
flashing colors since they distract the readers: "Blinking,
flashing, and color especially draw a reader's eye,
therefore, they should be used sparingly. . . . Overuse of any
device destroys its usefulness" (225). Also, he points out
that online screens have different color resolutions than
paper. Both Hofmeister and Hysell suggest that color should
be limited to showing emphasis and that no more than four
colors should be used in the documentation.

Cueing devices

Researchers advise using cueing devices--headings,
titles, tables of contents, and indexes--in documentation to
help the reader through the text and make referencing
sections easier.

Headings and titles Research suggests that
"readers understand and remember text preceded by titles
and/or headings better than text with no headings" (Benson
38). Benson also claims that "poor readers are
significantly aided by headings that are full statements or
questions" (38). A study conducted by Mirel, Feinberg,
Allmendinger, Mead, and Theodos support Benson's view.
Mirel and her colleagues' findings suggest that writers
should use a question/answer format in headings to help readers process the explanations (RT-172).

Carliner expands on the above findings by explaining that a good heading motivates readers and "creates an impression about the text" (72). He believes that headings should be used to spark readers' curiosity and make them want to read further. To create effective headings Carliner gives four suggestions: (1) use the gerund form of verbs to induce a friendly attitude; (2) use personal pronouns to create a rapport between the documentation and the user; (3) keep headings consistent in grammatical forms to make reading easier; and (4) use questions that users may have as the headings to spark readers' interests.

**Tables of contents** Researchers agree that a table of contents provides an introduction to the documentation and helps readers organize ideas and retrieve information. Southard states that tables of contents "act as advance organizers providing outlines of documents so that readers know the content and the order in which the contents are presented" (ET-87). Southard advises that the table of contents should contain the same main headings and main subheadings as those in the documentation. Lang also states that the table of contents should use all headings and subheadings which appear in the documentation and should be organized around the user’s needs. Finally, Mullins
suggests that the table of contents is a reflection of the software package and can be used as a means of judging the quality of the software package. She advises that chapter titles, topics, and subtopics should reflect the user’s needs and progress from basic to complex tasks.

Indexes Research suggests that indexing the contents of the documentation helps readers locate and retrieve information. Brockmann claims that indexes are "a reader’s most important key for finding information in a manual that is read using random access style of reading" (193). Brockmann suggests that indexes can be arranged by function, topic, command, or procedure. However, he advises to alphabetize the index and cross-reference the entries. Mullins advises that the index should break topics into subtopics and that it should be five percent--one page for every twenty written pages of a manual--of the documentation’s total length.

Finally, Scanlon and Raven conducted a study on how indexes should be organized and if indexes improve document retrieval. They gave subjects items to find using four types of indexes: (1) the user selects terms from previously defined terms, (2) the user indexes any word he/she feels is appropriate, (3) the user can index only the exact phrases that appear in the documentation, and (4) the user has a combination of the first three types of indexes.
The results of their study suggest that none of the indexing methods was faster. Scanlon and Raven also found that subjects who did not use an index did not find information significantly more slowly, so they suggested that a good table of contents can, perhaps, do the job of an index on a smaller manual. The manual they used in the study was fifty-one pages.

Documentation Layout

Researchers agree that the layout of the documentation can enhance readers’ understanding of the information, so it affects a document’s readability. Layout includes spacing and graphics.

Spacing

Researchers have found that the spacing of text on the page and the overall size of the page determine the effectiveness of the text. Spacing is divided into white space and page size.

White space

Generally, the space on the page which is not covered by text is referred to as white space, and researchers agree that documentation needs ample white space to increase readability. Theodos states "the amount and placement of white space on a page is as important (if not more important) as any other factor in determining the impact of the written word" (VC-28). He explains that white
space lets the writer group information into "chunks" and the amount of white space used around these chunks cues the reader to the relative importance and interconnections of the chunks (VC-29). The reader can thus see and understand the information more easily. Theodos also reports that recent studies have shown that white space has a tremendous impact on the reader's ability to remember text.

Researchers have found that white space should generally be added amply between divisions of text and in the margins (see Benson; Hartley "Eighty"; Brockmann). Brockmann suggests that "well-designed pages have a 'print density' of 40 percent--only 40 percent of the page has print on it" (75). He adds that white space should be "active white space"; in other words, white space helps the reader to discern between information on the page rather than serving as a frame surrounding the edges of the text (75). Williams and Beason also suggest having more white space in the left margin--or the side of the document which will be used for binding--to allow space for the gutter, or binding area.

Page size

Few researchers recommend a specific page size to use for documentation, but they agree that the document's size should have an appealing look and be practical--easy to handle and carry, or in the case of online, be easy to see
on the screen. Hartley's system for manual page layouts, although British, has been adapted by some American manual writers. Hartley recommends using a rectangular shape with sides having the ratio of 1:1.412 (Designing 10). He claims that this proportion will generate geometrically similar rectangles at each point along a vertical axis drawn between two opposite corners. This design allows for a standard page size in which work, text, or diagrams can easily be enlarged or reduced (12).

Graphics

Research has shown that graphics--illustrations, headers, footers, diagrams, pictures, tables, charts, etc.--can enhance the documentation when used effectively. Graphics need to adhere to the same basic standards as text; they need to be simple, easy to read, and spaced aesthetically on the page. Bernhardt uses examples of combinations of text and graphics to decide how the page of the document should show equilibrium. In other words, he claims the horizontal, vertical, and diagonal axes running through the center of the page should have a balance of print and white space on each of their respective sides. Hartley provides the following three basic graphic guidelines: (1) use regular spacing to help retrieval, (2) place figures in columns instead of rows for easier readability, and (3) use illustrations to help the reader by
motivating, explaining information, and promoting recall ("Eighty"). Benson adds to this by suggesting that each graphic device be explained in the text (38).

Online documentation has limited space, so elaborate pictures and illustrations would probably not be appropriate. However, numbering systems can be an effective online, as well as a manual, graphic device. Katzin suggests using a simple decimal form for steps, such as 1., 1.1, 1.2, 1.2.1., 1.2.2, 1.3, 2., 2.1, 2.2, etc. (37). Martin calls this type of numbering "tag text," and she claims it adds to the visual hierarchy of the documentation, which helps readers interpret the information (10).
I formulated my software development process based on the suggestions made in the research. Thus I used a progression including the following steps: style sheet development, software analysis, and documentation design. In the first step, I developed a basic style sheet on guidelines for developing a low-budget manual. In the second step, software analysis, I became familiar with the software, B-Stat, which I would be documenting. In the third step, documentation design, I developed the manual for the software package.

Style Sheet Development

Before beginning the software package documentation project, I developed, formatted, and organized a style sheet of basic documentation guidelines based on methods and specifications found in the research (see Appendix A). Since the style sheet is aimed particularly at low-budget documents, I selected researchers’ suggestions applicable to a low-funded documentation project: for example, sophisticated graphics and colors would most likely not be in low-budget manual so information on these is excluded from the style sheet.

Software Analysis

The second step in my methodology was analyzing the
software package that I would be documenting. In this analysis I performed two main steps: I interviewed the software developer and I experimented with the software.

Interview of the software developer

Since the author of the software B-Stat, Gerald Burrell, was my only connection to the program, I interviewed him to find out the main purpose of the software, his intentions, the intended end-line users, his expectations for the documentation, and any limitations for the documentation or the software package. (See Appendix B for the questions I used in the interview.) After speaking with Burrell I had the following information:

1. Purpose - B-Stat is a software package written in Pascal to provide basketball statistics, such as baskets attempted, baskets made, steals, fouls, etc. B-Stat keeps individual and team records of all the stats per game and for the season for the home team, the end-line users. The program keeps only team--not individual--stats of opponents when they play the home team. B-Stat is designed to be an interactive program, so the user inputs the statistics and B-Stat can perform some small analyses of the data and graph the results for each stat for the whole team or per player for each game or the entire season.

2. Programmer’s intent - Burrell intends to polish the program and market it on shareware and through direct mail
order after the documentation is complete. This is his first attempt at marketing a program.

3. End-line users - Burrell wrote the program for high school and junior college basketball teams. He assumed the coaches, their assistants, or team managers would enter the data, so the audience would have at least a tenth-grade education or above. He didn’t know how much computer experience the end-line users would have. Therefore, I assumed my manual would be used by both experienced and inexperienced computer users, so I had to make sure the documentation was written thoroughly and simply enough to effectively communicate to the inexperienced users. Burrell also stated that people other than end-line users, such as programmers and companies interested in purchasing the software package to resell, would be looking at the manual, so it should look professional and well-organized.

4. Expectations - Burrell expected the document to be simple and short. Because of his budget and equipment limitations, he didn’t want a lot of graphics or elaborate designs in the manual, so he wanted it to look conservative, yet well-designed. He also didn’t want indexes or lengthy title pages that might give make the manual look complex. Finally, he showed me a size he wanted the pages to be.

5. Limitations - Burrell mentioned several limitations. Limited funds would be available to produce the manual, so
he wanted to be able to print the manual with his own equipment. Therefore, I used his IBM compatible computer system and WordPerfect 5.1 software to produce the manual. Also, he needed the manual to be small enough to mail cheaply and designed so that changes and additions could be added later to the manual.

Software experimentation

To understand the software and devise the organization of the documentation, I experimented with the software for several days. During my initial experience, Burrell was present in case I became stuck in the program. During my later experiments I was alone. For my initial experiment of the software package I devised a fictitious basketball team and games and statistics for its players. I then attempted every possible manipulation of the statistics with the software. During this first experiment, Burrell realized several programming errors in the software.

The purpose of my second experiment with the software was to find all possible error messages, system reactions, or mistakes, so I intentionally entered incorrect data or pushed the wrong keys. After this step I had some suggestions which Burrell used on the program. For example, I found the use of the same color for main menus and submenus to be confusing, so Burrell varied the colors of the menus. I suggested that an error message appear across
the bottom of the screen. This message gave the users options or told them which keys to push. I advised having a way to readily exit the program, and I suggested the use of a simple online help screen. While I was writing the documentation, I also had suggestions that were not implemented because of time; for example, I advised a key-coded help screen, but Burrell did not have the available time to program it into the software package.

During my first two experiments with the software I kept written records of every procedure that I performed, including its consequences and screen messages. From there I was able to produce a rough map of the program and its use from a task-oriented perspective. I experimented several more times, checking and altering the map of processes that I had devised.

Finally, I had Burrell devise fictitious team members and statistics, and I entered these data as though I would be starting the whole process from scratch--earlier I saved my data and retrieved it after each manipulation. During this experiment Burrell watched my methods to make sure that I was using his software in the way he intended and that I had the procedures and processes noted and mapped correctly.

Documentation Design

Using the general documentation style sheet and the software analysis as a background, I designed the overall
appearance and content of the software documentation (see Appendix D). To do this I followed three basic design steps: layout design, textual design, and style sheet design.

**Layout design**

In my first step, layout design, I devised the overall layout of the page and sections of the manual by experimenting with different layouts and printouts to find the one most appealing. I decided to use a page size of 7" x 9", because several manuals that I looked at for ideas used this size and it was attractive. Also, Burrell wanted the manual to be in this size range. I used both a header and a footer--each having two black lines .02" thick and separated by a blank line--to add continuity to the pages and set off the text. This idea I also got from examples, not research. The header and footer are 0.5" from the top and bottom margins. In the headers I placed the section names of the manual in bold above the top black line to help make the pages easier to reference. I placed page numbers in the middle of the page below the footer so they would be easy to spot. Finally, I decided that the text would be double-spaced where possible from the header and footer to add more white space; however, each step in an instruction and introductory paragraphs would be single-spaced to help clump information together and break up the
page. I avoided confusing line breaks at the end of pages by adding more or less spacing, and I tried to fit sets of instruction on one page where possible.

I planned the overall sections of the manual according to tasks. Thus, the sections are organized according to how the end-line users will use the B-Stat program; for example, the users will have to use the first section of the manual, "Getting Started," before they use any of the other sections, since this section installs the program and the basic data. Also, I set up the data modification sections before the viewing, graphing, and printing sections, since the user needs correct data to get correct statistics. I placed the next four sections in the order of viewing, graphing, and printing, since this is logically what users would do; they would enter data, view the data to confirm that it’s correct, graph the statistics to see if they want a printout, and, finally, print out the graphs and statistics. I also added a section entitled "Helpful Hints" at the end of the manual, so the software users would have some general information about the software or problems that might occur while using it. This section was placed last in the manual, since not every user will need it.

To help the reader understand the procedures and to see the information, I used two classes of headings along with a
numbering system for the steps of the instructions. The main headings correspond to the main headings on the title page and denote sections of the manual, so they each begin on a new page. The subheadings are divisions of the main heading, and they also correspond to the subheads on the table of contents. The headings and paragraphs are left-justified.

The numbering system in the manual is the basic 1., 1.1, 2, 2.1, 2.1.1, 2.1.2, 2.2, etc. system. To differentiate between these different levels, I indented the different levels by five spaces.

I added a cover sheet, a table of contents corresponding to the headings, and an introduction to help the user understand the manual and to make the manual more inviting and give a positive image to the reader. I wanted to produce an index; however, the manual seems to be easy to reference, it doesn’t contain complicated, technical procedures, and Burrell did not want an index. However, I did recommend adding it if the manual is lengthened.

Textual design

I used my research findings and the general style sheet to plan and format the text of the manual. I used left-justified Courier 10 cpi (characters per inch) for all the text and Courier 12 cpi for all the headings. I used a smaller font for the text so that it would look more
condensed on the page and take less space. Since serif type is the easiest to read, I chose the only serif available on the computer system that I was using. I enlarged the main headings to make them stand out, and I bolded the subheadings for easier reference--these choices were made by experimenting with the text.

Besides being visually readable, the text needed to be semantically and syntactically easy to read, so I used short command sentences and familiar words. For example, instead of using the term cursor to describe one of the keyboard buttons, I used the term arrow keys. Also, since I wanted to cover all possible users, I wrote the instructions step-by-step in very simplified form. Finally, I placed all text that was describing what the user would see on the screen into boxes to give the effect of the screen on the page and to help break up the text on the page.

*Style sheet design*

After finishing the manual I developed a specific style sheet for it (see Appendix C). This specific style sheet can be used in the future by anyone to expand on the manual or revise it.
CHAPTER IV. CONCLUSIONS

By doing this study, I realized that little research exists for many areas of computer documentation writing and that much of the existing research, which is accepted as fact, is based on opinions and observations. Also, the current research does not deal with specifics. For example, one can find several articles on graphics, but none explain how writers decide what graphics to use and how designers can actually place them on the page. The research also doesn’t provide a general consensus or specifications for the best page size or format. Thus, the documentation writer is still left using his/her personal tastes and creativity to design the manual.

Finally, I determined from this study that the interaction between the documentation writer and the software writer or developer is crucial to the success of the software package. Without the programmer, Gerald Burrell, working with me, I could not have understood the software and its users, so the documentation would have reflected this lack of insight. Also, my experiences with the software helped Burrell find bugs and problems in the software that he was unaware of, and my suggestions as a first-time user gave him ideas on how to improve the program and make it more user friendly. In fact, after I had the manual completed, Burrell changed several menus on the
screen and developed online help cues for the user. For this reason, developing a final style sheet for the documentation is important; if Burrell makes further changes, they can easily be incorporated into the manual by following the guidelines in the style sheet.
REFERENCES


Gervickas, Vicki. "Current Recommendations in Document


Kujawski, A. "Follow These Rules to Write Good Instructions." *Chemical Engineering* 15 April 1985: 107-110.


Partridge, Stephen Kurt. "So What Is Task Orientation,


WORKS CONSULTED


Gettys, Deborah. "If You Write Documentation, Then Try a Decision Table." IEEE Transactions on Professional Communications 29.4 (1986): 61-64.


APPENDIX A

GENERAL STYLE SHEET FOR LOW-BUDGET COMPUTER DOCUMENTATION

Text

1. Use 8-12 point typeface. If this isn’t possible for online screens, make sure the type used is easy to see and read.
2. Use serif typeface, since it is easier to read.
3. Avoid hard-to-read fonts, such as script or cursive.
4. Use black text on white paper.
5. Be consistent with fonts and type sizes throughout the documentation.

Page

1. Use an attractive, easy-to-read page design.
2. Use a familiar page size, not bigger than 8.5" x 11" and not smaller than 6" x 8".
3. Number the pages in an easily seen location.
4. Provide page headings for easy referencing.
5. Use white or off-white paper.
6. Have margins at least 1" from top and bottom.
7. Have at least 1.5" left margins and 1" right margins (assuming the document is bounded on the left).
8. Arrange text and graphics vertically on the page.

9. Avoid running over less than one sentence of a paragraph or an instruction onto the following page.

10. Leave extra space on the left margin for binding the document.

**Language/Tone**

1. Use a positive tone (avoid negations and negative words).

2. Use conversational tone and the language of the user.

3. Include the personal pronoun "you" to address the readers.

4. Avoid jargon and scientific terms (i.e., byte, database, textbase, RAM).

5. Avoid acronyms. If you do use an acronym, place the full name in parentheses beside the acronym when you first use it.

6. Avoid ambiguous language and ambiguous pronouns (i.e., Put it in your disk drive.)

7. Avoid wasteful words and use simple terms, for example:

   **Avoid**                     **Use**
   in the course of             in
in conjunction with and
proceed go
in the event of in/if
following after
at this point in time now
negative no
actualize do/begin

8. Avoid redundant words; for example:
link together
last of all
eliminate altogether
repeat again
refer back

Sentences
1. Use short, simple sentences.
2. Make sentences 20 words or less.
3. Use active voice.
4. Use command form, especially for instructions.
5. Avoid compound sentences.
6. Avoid multiple noun, verb, adjective, or adverb phrases in sentences.
7. Write task-oriented sentences and instructions.
8. Use positive language; for example, instead of
saying "Don’t turn off the computer." say "Keep the computer on."

Paragraphs
1. Use paragraphs that are 6-10 sentences.
2. Do not indent paragraphs.
3. Write procedures and instructions as steps not paragraphs.

Headings
1. Use headings to make text more readable.
2. Use the chapters, sections, or parts from the table of contents as headings in the documentation.
3. Regulate heading levels by indentation or bolding.
4. Use gerund phrases and questions to make headings.

Tables of Contents
1. Develop tables of contents from the divisions of the text in the document.
2. Organize alphabetically, or some way which is easy to read, like task orientation.
Headers and Footers

1. Use headers and footers to unify pages and help readers reference material.
2. Place headers and footers at least 1/2" above and below text.
3. Have at least 1/2" of blank space above and below headers and footers.

Capitalization

1. Avoid using all caps for titles and headings.
2. Follow general rules, such as beginning of sentences, computer languages, proper nouns, etc.

Punctuation

1. Avoid or minimize the use of semicolons, exclamation points, colons, quotation marks, and hyphens--especially in instructions.
2. Use colons before the steps of instructions when preceded by the phrases "for example," "do the following," or "as follows."
3. Use commas sparingly, but do use commas before coordinating conjunctions in compound sentences and to separate the items in a list of two or more entities.
Italics and Bolding

1. Avoid the overuse of italics and bolding, since it makes text harder to read.
2. Use bolding in shorter headings to show level, but avoid bolding in headings longer than one line.
3. Use italics instead of underlining for emphasis.

Underlining

1. Avoid underlining.
2. Substitute italics for underlining.

Graphics

1. Use graphics to enhance appearance and explain text (this may not be possible in online documentation).
2. Place graphics next to text they describe or represent.
3. Discuss all pictures, graphs, etc. in the text.
APPENDIX B
INTERVIEW QUESTIONS

1. What is your software package called?
2. What is the purpose of the software?
3. Who is the software written for?
4. Who do you see as the main audience or buyer for the software? Their education? Their ages? Their occupations? Their goals and uses for the software?
5. Have you contacted any of these potential customers to see what their backgrounds are?
6. Do you have firsthand experience with the software’s use?
7. What does the software do?
8. What does the software actually output?
9. How does the software basically work? (i.e., does it require extensive data input, etc.)
10. What are your intentions for the software package? Marketing endeavors?
11. What are your expectations for the manual?
12. Do you have any limitations for the software or the manual?
14. Is there a certain system you want me to use to produce the manual?
15. Do you have equipment I can use to produce the manual?
16. Will I need special equipment to run the software?
17. What is your time schedule to have the manual completed?
APPENDIX C

STYLE SHEET FOR THE B-STAT MANUAL

Text

1. Use 10 cpi (characters per inch) Courier typeface for all text, except headings.
2. Use 12 cpi Courier typeface for all headings.
3. Use black text on white paper.
4. Use large (Word Perfect 5.1 term) text for the main headings in the title page.
5. Use large (Word Perfect 5.1 term) for main headings within the document.
6. Use bolded text for subheadings on title page and within document.
7. Use very large (Word Perfect 5.1 term) for the title "contents" on the table of contents page.
8. Use large (Word Perfect 5.1 term) for the text on the cover page.
9. Double-space all text, except introductory paragraphs and individual instruction steps.

Page

1. Use an 7" x 9" page size.
2. Have a top margin of 1" and a bottom margin of 1".
3. Use white or off-white paper.
4. Have a 1.5" left margin and 1" right margin.
5. Arrange text and graphics vertically on the page.

6. Avoid running over less than one sentence of a paragraph or an instruction onto the following page.

7. Place the steps of each instruction on the same page whenever possible.

**Language/Tone**

1. Be positive, avoid negations and negative terms.

2. Use conversational tone and the language of the users.

3. Include the personal pronoun "you" to address the readers.

4. Avoid jargon and scientific terms (i.e., byte, database, textbase, RAM).

5. Avoid acronyms. If you do use an acronym, place the full name in parentheses beside the acronym when you first use it.

6. Avoid ambiguous language and ambiguous pronouns (i.e., Put it in your disk drive.).

7. Avoid wasteful words and use simple terms, for example:

<table>
<thead>
<tr>
<th>Avoid</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>in the course of</td>
<td>in</td>
</tr>
<tr>
<td>in conjunction with</td>
<td>and</td>
</tr>
</tbody>
</table>
proceed          go
in the event of  in/if
following       after

8. Specifically, use "arrow keys" instead of
   "cursors," and "you" instead of "user."

9. Use the word "type" before commands involving
   printable keys (keys which print a letter or
   symbol) and use the word "press" for nonprintable
   keys, for example:

   Type <<Q>> and press <<RETURN>>.

**T**erences

1. Use short, simple sentences.
2. Make sentences 20 words or less.
3. Use active voice.
4. Use command form, especially for instructions.
5. Avoid compound sentences.
6. Avoid multiple noun, verb, adjective, or adverb
   phrases in sentences.
7. Write task-oriented sentences and instructions.
8. Use positive language; for example, instead of
   saying "Don’t turn off the computer." say "Keep
   the computer on."
Paragraphs

1. Use paragraphs that are 6-10 sentences.
2. Do not indent paragraphs.
3. Write procedures and instructions as steps not paragraphs.
4. Use only after headings if they need an introduction (most likely only main headings will need an introduction and the section Helpful Hints).

Headings

1. Place the main heading in large text at the top of a new page with only the first letter in the main words capitalized. Double-space after the main headings.
2. Place the subheadings in bold with the first letter of the first word of the heading and the first letter of the rest of the words---except articles, coordinating conjunctions, and prepositions-capitalized. Double-space after the subheadings.
3. Use the chapters, sections, or parts from the table of contents for the wording for all headings.
4. Left justify all headings.
5. Indent 5 spaces only after the subheadings, not the main headings.

6. Use gerund phrases to make new headings if needed.

Table of Contents

1. Use the exact heading names for the items in the table of contents.

2. Organize by task.

3. Use the roman numeral numbering system (i.e., main headings would be I., II., III., and subheads would be A., B., etc.).

4. Begin the table of contents at the top margin on the first page following the cover sheet.

5. Triple-space between main headings and double space between subheadings and subheadings and main headings.

Headers and Footers

1. Use two solid black lines 0.2" in thickness with one space between them as the headers and footers.

2. Place these headers and footers 1/2" from the top and bottom of the page.

3. Have at least 1/2" of white above and below headers and footers.
4. Begin headers and footers on the first page of the table of contents.

5. Place section names in bold flush right on top black line of the header.

6. Place page numbers centered below the footer set off by hyphens (i.e., - 1 -).

**Capitalization**

1. Avoid using all caps for titles and headings.
2. Capitalize the main words in all headings throughout the documentation.
3. Use all caps for any letters or commands placed in corner brackets, for example: <<RETURN>>, <<BSTAT>>, <<Q>>, <<ESC>>.
4. Do not capitalize prepositions, conjunctions, and articles in headings.
5. Follow general rules, such as beginning of sentences, computer languages, proper nouns, etc.

**Punctuation**

1. Avoid overusing semicolons, exclamation points, colons, quotation marks, and hyphens (except for the hyphen in B-Stat), especially in the instruction steps.
2. Use commas sparingly (use only for lists, independent and dependent clauses).
3. Use commas before coordinating conjunction in compound sentences.

**Italics and Bolding**

1. Avoid the overuse of italics and bolding, since they make the text harder to read.
2. Use bolding in shorter headings to show level, but avoid bolding in headings longer than one line.
3. Use italics instead of underlining for emphasis.
4. Use bolding for all headings when referring to them in the text of the manual, not when they are acting as headings (i.e., In *Getting Started* you will learn how to set up B-Stat on your computer).

**Underlining**

1. Avoid underlining.
2. Substitute italics.

**Graphics**

1. Place all text messages or screen descriptions in
a text box below the text it follows and indented five spaces.

2. Double-space before and after the text box were able.

Lists/Instructions

1. Place all instructions in list format using the numbering system 1., 1.1, 1.2, 2., 2.2.1, 2.2.2, 2.3, 3., etc.

2. Indent each number 5 spaces from the preceding text, for example:

To check the information do the following:

1. Press <<N>> if information is wrong, and do the following:
   1.1 Press the number beside the wrong entry.
   1.2 Press <<RETURN>>.

2. Press <<Y>> if information is correct.

3. Use command form sentences.

4. Use only 1-2 instructions per step.
B-Stat

Basketball Statistic Software Package
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Section I: Getting Started</strong></td>
<td>6</td>
</tr>
<tr>
<td>A. Installing the Program</td>
<td>6</td>
</tr>
<tr>
<td>B. Running the Program</td>
<td>8</td>
</tr>
<tr>
<td>C. Entering Team Information</td>
<td>9</td>
</tr>
<tr>
<td>D. Entering Player Information</td>
<td>10</td>
</tr>
<tr>
<td>E. Entering Season Games</td>
<td>12</td>
</tr>
<tr>
<td><strong>Section II: Entering Stats</strong></td>
<td>14</td>
</tr>
<tr>
<td><strong>Section III: Modifying Games</strong></td>
<td>17</td>
</tr>
<tr>
<td>A. Changing Game Information</td>
<td>17</td>
</tr>
<tr>
<td>B. Adding Games</td>
<td>19</td>
</tr>
<tr>
<td>C. Deleting Games</td>
<td>21</td>
</tr>
<tr>
<td><strong>Section IV: Modifying Players</strong></td>
<td>22</td>
</tr>
<tr>
<td>A. Adding Players</td>
<td>22</td>
</tr>
<tr>
<td>B. Deleting Players</td>
<td>24</td>
</tr>
<tr>
<td>C. Changing Player Information</td>
<td>24</td>
</tr>
<tr>
<td><strong>Section V: Modifying Stats</strong></td>
<td>26</td>
</tr>
<tr>
<td>A. Modifying Player Stats</td>
<td>26</td>
</tr>
<tr>
<td>B. Modifying Team Stats</td>
<td>28</td>
</tr>
</tbody>
</table>
Introduction

Welcome to B-Stat software package! With B-Stat you have a winning edge. B-Stat keeps track of player and game stats, so you know your team's strengths and weaknesses. You can use B-Stat's statistics to design team practices and game strategies to best use your player's abilities and capitalize on your opponent's weaknesses. With B-Stat your team is on its way to Number 1!

What can B-Stat do?

B-Stat is an easy-to-use program that keeps track of stats for twenty players during a thirty-six game season. B-Stat can compare, summarize, and graph player and game stats. B-Stat can also track your opponent's stats and produce printouts comparing your team's performance against your opponent's throughout the season.

B-Stat is simple and easy to use. B-Stat has a built-in automatic installation program, an online help screen, and online command cues. B-Stat also uses multi-colored screens and menus to help you enter and retrieve data quickly.

What does B-Stat require?

B-Stat requires an IBM compatible computer with at least one megabyte of free hard drive space. Installation requires only a basic understanding of the DOS prompt codes. B-Stat is designed for a color monitor, but it will work on a monochrome monitor.
Who designed B-Stat?

B-Stat was designed by avid basketball lover, Gerald Burrell of Burrell Software Concepts. Mr. Burrell has developed software for a variety of projects, including a voice activated robot. Burrell Software Concepts is currently developing new statistical packages and providing consulting services for other companies. The company also writes customized software for individual or company use.

What you should know about the manual?

The manual is designed from your perspective, so it is organized by tasks for an easy-to-understand, simple-to-use guide. There are four basics to remember while using the manual:

1. The symbol of two corner brackets on both sides, "<< >>", surrounds actual commands for you to press or letters and symbols to type. For example <<RETURN>> means to push the return key. These brackets are also used to show what is actually on the screen.

2. You must do the first section of the manual, Getting Started, before using B-Stat, since it installs the program and basic data. Getting Started is organized in chronological order, and you must follow this section step by step.
3. Section VII, **Helpful Hints**, provides some basic DOS hints and solutions to possible problems. Section VII also lists all the stats (along with explanations) that B-Stat includes.

4. If you have problems, look in Section VII, **Helpful Hints**, or call Burrell Software Concepts (515) 292-0973.

---

**Do you have further comments or questions?**

If you have comments or questions about B-Stat, please call Burrell Software Concepts at (515) 292-0973, or write to the following address:

Burrell Software Concepts  
P.O. Box 1051  
Welch Avenue Station  
Ames, IA  50010
Section I: Getting Started

Getting Started describes how to set up B-Stat on your computer and enters beginning information. For the program to operate properly, you must complete all five steps in this section first:

- Step 1: Installing B-Stat
- Step 2: Running B-Stat
- Step 3: Entering Team Information
- Step 4: Entering Player Information
- Step 5: Entering Season Games

(Note that you do Step 1 only when you first install the program. You have to do Steps 2-5 before your first use of the program; however, you can also use these steps to alter the data or to set up B-Stat for a new season.)

Step 1: Installing B-Stat

1.0 Set your computer to the DOS mode.

1.1 Place the B-Stat disk in your disk drive.

1.2 Type "<(DRIVE LETTER) : BINSTALL>>" (ie., if the disk is in drive "A" then you would type "<A:BINSTALL>>").

1.3 Press "<RETURN>>.

A welcome message will appear.
1.4 Press <<RETURN>> again.

A message showing available hard drive space will appear. One meg of memory must be available on either the C drive or the D drive for installation to continue.

1.5 Press <<RETURN>> to copy B-Stat on the C drive, or follow the screen directions to change the drive path to D.

1.6 Press <<RETURN>> to complete the installation. (This will take you back to the DOS prompt, but you will still be in the B-Stat directory.)
Step 2: Running B-Stat

If you just installed B-Stat, you will still be in the B-Stat directory (i.e., if you installed B-Stat on the C drive, then your prompt would look like << C:\BSTAT\ >>). If you are starting from the C or D prompt (<< C:\ >> or <D:\ >>), type <<CD BSTAT>> and continue.

2.0 Type <<BASKET>> at the B-Stat directory prompt.

2.1 Press <<RETURN>>.

A team information box will appear.
Step 3: Entering Team Information

3.0 Type your school's name and press <<RETURN>>.

3.1 Type your team's name and press <<RETURN>>.

3.2 Type your coach's name and press <<RETURN>>.

3.3 Type your city and state and press <<RETURN>>.

3.4 Type the season years and press <<RETURN>>.

3.5 If the information is incorrect then
   3.5.1. Type <<N>>.
   3.5.2. Type the letter next to the incorrect entry and press <<RETURN>>.
   3.5.3. Type the correction and press <<RETURN>>.

3.6 If the information is correct type <<Y>>. (Go back to 3.5.1 if information is still incorrect.)

3.7 Press <<F1>> to continue the program.

A Main Menu will appear.
Step 4: Entering Player Information

4.0 Use the arrow keys to highlight "Enter New Data" and press <<RETURN>>.

The Game Menu will appear.

4.1 Use the arrow keys to highlight "Enter Player List" and press <<RETURN>>.

A warning screen will appear. Continuing will erase any previously entered players.

4.2 Type the number of players (maximum of 20) and press <<RETURN>>.

A player information box will appear.

4.3 Follow the screen and type the information, pressing <<RETURN>> after each entry. The program will automatically continue until all the players have been entered.

A chart with all the players' information will appear.
4.4 If the information is incorrect, then

4.4.1 Type «N>>.

4.4.2 Type the number of the player to change and press «RETURN».

4.4.3 Use the cursor keys to highlight the incorrect entry and type the correct information.

4.4.4 Press «RETURN».

4.4.5 Press the escape key when all corrections have been made.

4.4.6 Repeat step 4.4 if any information is still incorrect.

4.5 If all the information is correct, type «Y>>.

The New Data Menu will reappear.
Step 5 - Entering Season Games

5.0 Use the arrow keys to highlight "Enter Season Games" and press «RETURN».

A warning screen will appear. Continuing will erase any previously entered games.

5.1 Press «RETURN» again to enter your opponent teams.

A game information chart will appear.

5.2 Do the following steps for each of your opponents team (36 opponents are allowed):

5.2.1 Type the team's name and press «RETURN».

5.2.2 Type the date of the game in numerical form and press «RETURN». (i.e., December 1, 1991 would be entered 12/1/91.)

5.2.3 Type «H» for home game or «A» for away game.

5.2.4 Type «Y» for conference game or «N» for nonconference.
5.3 Press <<RETURN>> when all the teams have been entered.

A display of entered teams will appear.

5.4 If the information is incorrect press <<N>>, then

5.4.1 Type the number that corresponds to the team you want to change.

5.4.2 Use the arrow keys to highlight the information you want to change.

5.4.3 Type the correct information and press <<RETURN>>.

5.4.4 Repeat step 5.4 until all information is correct.

5.5 If all information is correct the press the escape key.

This will take you back to Game Menu.

5.6 Use the arrow keys to highlight "Quit" on the Game Menu to go back to the Main Menu.
Section II: Entering Stats

Entering Stats explains how to enter your players' stats. B-Stat automatically figures percentages and averages, so all you have to do is enter the raw stats for each player. B-Stat will provide statistical averages and percentages for individual players, the whole team, or season games.

To enter the players' stats, do the following.

1. Go to the Main Menu.

2. Use the arrow keys to highlight "Enter Statistics for a Game" and press <<RETURN>>.

   The Game Menu will appear.

3. Use the arrow keys to highlight "Enter Statistics" and press <<RETURN>>.

   A list of games will appear.

4. Type number next to the game whose stats you want to enter.

   A list of players will appear.
5. Type the number of the player whose stats you want to enter and press <<RETURN>>. (Note that you can enter <<ALL>> and press <<RETURN>> to enter all the players sequentially at the same time.

A list of statistic categories will appear.

6. Follow directions on the screen. (i.e., enter numbers in highlighted box and press <<RETURN>> after each entry.

7. Repeat step 1.6 until all data has been entered for the player.

8. If player stats are incorrect, then
   8.1 Press <<N>> and <<RETURN>>.
   8.2 Type letter next to mistake, and press <<RETURN>>.
   8.3 Type correct data and press <<RETURN>>.
   8.4 Repeat 1.8 until all information is correct.

9. If you enter players using the automatic sequencing, repeat steps 4-8 until you have entered the data for all the players.
10. Press the escape key to return to Game Menu.

11. Use the arrow keys to highlight "Quit" and press <<RETURN>> to return to Main Menu.
Section III: Modifying Games

Modifying Games explains how to modify your schedule by

A. Changing a Game - how to change general game information, such as location, date, or team name (this does not include changing game stats).

B. Adding a Game - how to add a game to your season schedule

C. Deleting a Game - how to delete a game from your schedule

A. Changing Game Information

To change game information, except for stats, about a game do the following steps:

1. Use the arrow keys to highlight "Enter Statistics for a Game" and press <<RETURN>>.

   The Game Menu will appear.

2. Use the arrow keys to highlight "Modify a Game" on the Game Menu and press <<RETURN>>.

   A list of games will appear.
3. Type the number beside the game you want to change.

   The game information chart will appear.

4. Use cursor keys to highlight the information you want to change.

5. Type the correct information and press <<RETURN>>.

6. Repeat steps 2-5 until all information is correct.

7. Press the escape key to return to Game Menu.

8. Type <<Q>> to return to Main Menu.
B. Adding Games

To add games to your roster, do the following steps:

1. Use the arrow keys to highlight "Modify Statistic for a Game" on the Main Menu and press «RETURN>>.

   The Game Menu will appear.

2. Use the arrow keys to highlight "Modify a Game" and press «RETURN>>.

   A list of games will appear.

3. Type the number following last game and press «RETURN>>.

4. Do the following steps for each of the games you want to enter.

   4.1 Type the team's name and press «RETURN>>.

   4.2 Type the date of the game in numerical form and press «RETURN>> (i.e., December 1, 1991, would be entered 12/1/91).

   4.3 Type «H>> for home game or «A>> for away game.

   4.4 Type «Y>> for conference game or «N>> for nonconference.
4.5 Press «RETURN>> when all new teams have been entered.

5. If the information is incorrect type «N>>, then

5.1 Type the number that corresponds to the team you want to change.

5.2 Use the arrow keys to highlight the information you want to change and press «RETURN>>.

5.3 Type the correct information and press «RETURN>>.

5.4 Repeat step 5 until all information is correct.

6. If all information is correct, then press the escape key.

This will take you back to Game Menu.

7. Use the arrow keys to highlight "Quit" on the Game Menus and press «RETURN>> to go back to the Main Menu.
C. Deleting Games

To delete a game from your schedule do the following steps:

1. Use the arrow keys to highlight "Modify Statistic for a Game" on the Main Menu and press <<RETURN>>.

   The Game Menu will appear.

2. Use the arrow keys to highlight "Clear a Game" press <<RETURN>>.

   A list of games will appear.

3. Type <<D>> and then press <<RETURN>> to activate the delete mode.

4. Type the number of the game you want to delete and press <<RETURN>>.

5. Repeat steps 3-4 for every game you want to delete.

6. Press the escape key to return to Game Menu.

7. Type <<Q>> to return to Main Menu.
Section IV: Modifying Players

Modifying Players explains three player manipulations:

A. Adding Players - how to add players to your roster
B. Deleting Players - how to delete players from your roster
C. Changing Player Information - how to change general player information, such as name, jersey number, etc. (this does not include changing stats).

A. Adding Players

To add players to your roster do the following steps:

1. Use the arrow keys to highlight "Modify Players" on Main Menu and press <<RETURN>>.

   A Player Menu will appear.

2. Use the arrow keys to highlight "Add Players" and press <<RETURN>>.

   A player information box will appear.
3. Type the number of players to add and press <<RETURN>>.

4. Follow the screen instructions and type the player's information, pressing <<RETURN>> after each entry. The program will automatically continue until all the players have been entered.

A chart with all the players' information will appear.

5. If the information is incorrect, then
   5.1 Type <<N>>.
   5.2 Type the number of the player to change and press <<RETURN>>.

The player information screen will appear.

5.3 Use the arrow keys to highlight the incorrect entry and type the correct information.
   5.4 Press <<RETURN>>.

6. Repeat step 5 until all information is correct, then type <<Y>>.

7. Press <<RETURN>> to return to Main Menu.
Modifying Players

B. Deleting Players

1. Use the arrow keys to highlight "List Players" on Main Menu.
2. Type <<D>> and press <<RETURN>>.
3. Type the number corresponding to the game you want to delete and press <<RETURN>>.
4. Type <<Q>> and press <<RETURN>> to return to Main Menu.

C. Changing Player Information

To change all player information, except stats, do the following steps:

1. Use the arrow keys to highlight "List Players" on Main Menu and press <<RETURN>>.

   A list of players will appear.

2. Type the number of player you want to change and press <<RETURN>>.

   A player information box will appear.
3. Use the arrow keys to highlight the incorrect entry.

4. Type the correct information and press «RETURN».

5. Press the escape key when all corrections have been made.

   The player information screen will appear.

6. If all information is correct, type «Y» to return to main menu; otherwise repeat steps 2-5.
Section V: Modifying Stats

Modifying Stats explains how to modify game stats. You can modify stats by

A. Modifying Player Stats
B. Modifying Team Stats

A. Modifying Player Stats

To modify a player's stats do the following steps:

1. Use the arrow keys to highlight "Modify Statistics" on Main Menu and press <<RETURN>>.

   A list of games will appear.

2. Type which game has incorrect stats and press <<RETURN>>.

   A list of players will appear.

3. Type the player's number and press <<RETURN>>.

   A list of statistics will appear.
4. Type the letter beside the stat you want to change.

5. Type the new figure and press <<RETURN>>.

6. Enter <<Q>> to return to Main Menu.
B. Modifying Team Stats

Modifying Team Stats only allows you to change the scores of games. To do this complete the following steps:

1. Use the arrow keys to highlight "Modify Statistics" on the Main Menu and press <<RETURN>>.

   A list of games will appear.

2. Type number of game you want to change and press <<RETURN>>.

   A game information box will appear.

3. Type your opponent's score and press <<RETURN>>.

4. Type your score and press <<RETURN>>.

   The list of games will reappear.

5. Type <<Q>> to return to Main Menu.
Section VI: Viewing Stats

Viewing Stats explains how to view your players' stats for each game on the computer screen. To view player stats do the following steps:

1. Use the arrow keys to highlight "View Statistics Menu" on the Main Menu and press <<RETURN>>.

   The View Stat Menu will appear.

2. Type the letter of the stat you want to view and press <<RETURN>>.

   A list of games will appear.

3. Type the number corresponding to the game that you want and press <<RETURN>>.

   A list of players will appear.
4. Type the number corresponding to the player you want to view and press <<RETURN>>.

A list of player's stats will appear.

5. Type <<Q>> to return to Main Menu or press <<RETURN>> to go back to step 2.
Section VII: Graphing Stats

Graphing Stats shows you how to produce online graphs of your player's or your team's stats. You can graph the whole season or each game individually. You can also graph a single stat or all the stats together. To graph statistics do the following steps:

1. Use the arrow keys to highlight "Graph Player's Statistics" on Main Menu and press «RETURN».

   The Stat Menu will appear with different graphing options.

2. Type the letter of the graph you want and press «RETURN».

   A list of players will appear.

3. Type the number corresponding to the player you want to graph the press «RETURN ». If you want to graph the team as a whole, type «Z» then press «RETURN».

   The graph will appear on the screen.

4. Type «Q» to quit and return to the Main Menu or press «C» to return to step 2.
Section VIII: Printing Stats

Printing Stats shows you how to print out stats. The printing function has its own menu, so you only have to designate the game and player. To print out statistics do the following steps:

1. Use the arrow keys to highlight "Print Menu" on the Main Menu and press <<RETURN>>.

   The Print Menu will appear.

2. Type the number of the print function you want and press <<RETURN>>.

   A game list will appear.

3. Type the number of the game you want to print and press <<RETURN>>. Or type <<100>> and press <<RETURN>> for entire season.

   A player list will appear.
4. Type the number of the player you want to print and press <<RETURN>>. Or type <<100>> and press <<return>> to include all the players.

A select printer message will appear.

5. Type the number which corresponds to your type of printer.

Your document should be printing.

6. Type <<Q>> to quit and go back to Main Menu or type <<C>> to return to step 2.
Section IX: Helpful Hints

Helpful Hints gives you some additional information about B-Stat. This section is divided into the following:

A. Operating Problems
B. Printer Problems
C. Data Saving
D. Exiting B-Stat
E. Starting a New Season
F. Explanations of the Stats
G. Running B-Stat from Disk
H. A Closer Look at the Menu System
I. B-Stat Updates

A. Operating Problems

Operating problems are often caused by incorrect installation or equipment problems. The following example questions should answer most of your operating questions:

What does it mean if the message "cannot find," "files missing," or some equivalent prompt comes up on the screen while trying to run B-Stat?

This usually means that you are not in the B-Stat directory. Remember that you must be in the B-Stat directory to run B-Stat. To get into the B-Stat directory simply type "<CD BSTAT>" besides the drive prompt where B-Stat is stored (either "<C:>") or "<D:>").

If you are still having problems finding B-Stat, check your drive directories to make sure B-Stat is still copied on your hard drive.
If B-Stat still isn't working but you do have some B-Stat files on your hard drive, then some of B-Stat's files may have been accidently erased. To avoid losing any data remaining, use DOS commands to copy all the B-Stat files off the B-Stat disk into the B-Stat directory on that same hard drive. Try running B-Stat. If B-Stat runs, then view your files and delete the duplicates (BE SURE TO SAVE ANY FILES CONTAINING DATA).

If B-Stat still isn't running properly, delete any B-Stat files left on your hard drive and reinstall B-Stat. This will erase any data that you have entered, so use this step as your last resort.

What is wrong if the program won't perform all its functions or flashes "files missing" or some similar message on the screen while I'm using B-Stat?

This means that not all of B-Stat's files were copied onto the hard drive or that some of the files were copied under another directory. Check all of your hard drive directories to make sure the B-Stat files are all on the same drive. If you can not tell if files are missing, use the DOS commands and copy all of the files off the B-Stat disk into B-Stat directory. If that still doesn't fix the problem, you will have to reinstall B-Stat, and this will erase any previous data you had entered in the program.
B. Printer Problems

B-Stat works with both dot matrix and laser printers. B-Stat only prints on the standard printer mode for dot matrix printers. The only laser printers B-Stat currently works with are the Hewlett-Packard Laser Jet II and III (program updates will expand the number and types of laser printers B-Stat can use). Note that some other types of laser printers may work with B-Stat, but they may not produce accurate printouts.

If you are having problems printing, check your printer configuration to make sure it can work with B-Stat. Also, make sure you chose the right printer type in the printing process. Finally, check to make sure your printer is hooked up properly.

C. Saving Data

B-Stat automatically saves the data that you enter when you exit the program using the menu exit commands. B-Stat also has an automatic save function that operates at built-in timed intervals. This function may not save all your data if you exit the program without exiting all the menus. In other words, to protect your data you must exit B-Stat by using the exit program commands on the menus.

Whenever you use the "Enter New Data" menu commands (Steps 3, 4 and 5 of Getting Started), all previous data will be erased.
D. Exiting B-Stat

To quit B-Stat follow the exit commands on the menus. Once you have quit B-Stat, you will still be in the B-Stat directory. To return to your main directory type <<CD..>>.

If you shut off your computer or exit B-Stat without using the menu exit commands, any data that you entered or changed before exiting may not be saved.

E. Starting a New Season

To begin keeping stats for a new season repeat Steps 3, 4, and 5 of the Section I: Getting Started. This will erase any previous data.

Note: If you have most of the same players as the previous season, repeat Steps 3 and 5 of Section I: Getting Started. This will erase all game information and keep your player information. Then use Section IV: Modifying Players to make any minor adjustments to your roster.
F. Explanations of the Stats

B-Stat keeps track of the following stats (any explanations are in parentheses):

2 Point Baskets Made
2 Point Baskets Attempted
3 Point Baskets Made
3 Point Baskets Attempted
Free Throws Made
Free Throws Attempted
Assists
Blocked Shots
  (number of shots the player blocked)

Number of Steals
Number of Def. Rebounds
  (number of defensive rebounds)

Number of Off. Rebounds
  (number of offensive rebounds)

Number of Turnovers
Number of Forced Turnovers
  (number of turnovers your player caused
  the opponent to make).

Number of Guts
  (number of charge fouls player forced
  the opponent to make).

Number of Fouls Committed
G. Running B-Stat from Disk

If your computer has enough internal memory, you can run B-Stat directly from the disk. Note that running B-Stat from disk slows down the operation of the software and some computers may not have enough internal memory to properly run B-Stat from disk. Also, your data will be saved on your hard drive or another disk, not the B-Stat disk.

To run B-Stat from disk do the following:

1. Place B-Stat in your disk drive.

2. Type "\" (letter of your disk drive) :)" and press "RETURN" (i.e., if you use drive A, then you would enter "A:\" next to the DOS prompt).

3. Type "BASKET" and press "RETURN".
H. A Closer Look at the Menu System

B-Stat uses easy-to-read menus to help you through the program. The following flow charts give you a brief overview of the menus and where they take you in the program or what they produce (menus are bolded).

### Main Menu

- **Season Games Menu**
  - Enter Statistics for a Game
  - Modifying Statistics

- **Printer Menu**
  - View Statistics Menu
    - Graph Player's Statistics
    - List Games
    - List Players
    - Quit Program

- **Game Menu**
  - Game List
  - Game List

- **Print Menu**
  - Stat. Menu
  - Player list
  - Game list
  - Player list

- **Exit program**
**Game Menu**

- Enter New Data
- Modify a Game
- Clear a Game
- List Games
- Modify Player
- Quit

- Enter Season Games
- Enter Player list
- Quit

A list of games appears.
A list of players appears.
Returns you to **Main Menu**.

Enter Season Games
Enter Player List
Quit

Lets you enter games for the season and erases previous games
Lets you enter players and erases previous player data

Takes you back to the **Game Menu**
Helpful Hints

A list of players will appear. Make your selection.

A list of printers will appear. Enter your printer number.

Your selection will be printing.
Statistics Menu

View Individual Player
View Players Sequentially
View Games and Scores
View Performance Points
View Season Stats for Players
Return to Main Menu

A list of games will appear.

Returns you to the Main Menu.

A list of players will appear. Make your selection.

The stats you selected will appear.
I. B-Stat Updates

Burrell Software Concepts is constantly striving to improve and expand on its products. Currently, we are developing a faster, more expandable B-Stat which can save stats for more than one season. You will receive this first update, along with its documentation, free. We will also notify you about any further updates made on B-Stat. You will be eligible to receive these further updates and their documentation at reduced rates.

B-Stat is protected by copyright laws. Copying the B-Stat software or its documentation without written permission from Burrell Software Concepts is against the law.