(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A FINANCIAL DECISION SUPPORT SYSTEM FOR
U.S. NAVY PUBLIC WORKS DEPARTMENTS

by

S. Keith Hamilton

December 1989

Thesis Advisor Shu S. Liao

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 36	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
A FINANCIAL DECISION SUPPORT SYSTEM FOR U.S. NAVY PUBLIC WORKS DEPARTMENTS

12 PERSONAL AUTHOR(S)
Hamilton, S. Keith

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	December 1989	188

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

To date, most efforts for developing microcomputer based financial applications for small Navy public works departments (PWD's) has occurred using an ad hoc approach, database management software, and independent user development. Recent technological advancements in computer hardware and software provide a cost efficient method of improving the effectiveness of financial decision making in PWD's. This thesis addresses two primary research questions:

1) Can a generally applicable model for financial accounting and reporting be developed for PWD's using a commercially available decision support system generator such as Lotus 1-2-3?

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Shu S. Liao	22b TELEPHONE (Include Area Code) (408) 646-2205 22c OFFICE SYMBOL 54LC

DD Form 1473, JUN 86 — Previous editions are obsolete — S/N 0102-LF-014-6603

19. Continued

2) Can such a model be used to apply decision support system theory to financial management within the PWD?

In this thesis, Lotus 1-2-3 Release 3.0 was used to develop a microcomputer based financial accounting and reporting model for PWD's. A small test data set was used to demonstrate the model and illustrate its use as a decision support system. The software developed in this thesis is available from the thesis advisor upon request.

Approved for public release; distribution is unlimited

A Financial Decision Support System For U.S. Navy Public Works Departments

by

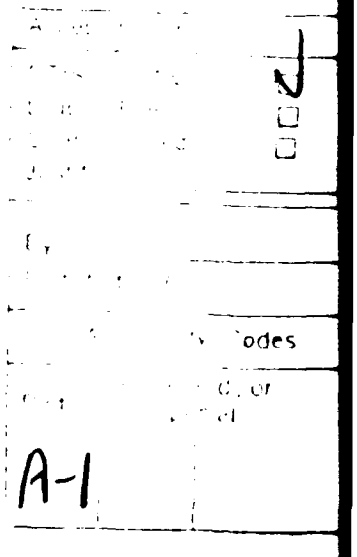
S. Keith Hamilton
Lieutenant, Civil Engineer Corps, United States Navy
B.S.- Ch.E., University of Missouri, 1983

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
December 1989



Author: _____
S. Keith Hamilton

Approved by: _____
Shu S. Liao, Thesis Advisor

_____
Tung X. Bui, Second Reader

_____
David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

To date, most efforts for developing microcomputer based financial applications for small Navy public works departments (PWD's) has occurred using an ad hoc approach, database management software, and independent user development. Recent technological advancements in computer hardware and software provide a cost efficient method of improving the effectiveness of financial decision making in PWD's. This thesis addresses two primary research questions:

- Can a generally applicable model for financial accounting and reporting be developed for PWD's using a commercially available decision support system generator such as Lotus 1-2-3?
- Can such a model be used to apply decision support system theory to financial management within the PWD?

In this thesis, Lotus 1-2-3 Release 3.0 was used to develop a microcomputer based financial accounting and reporting model for PWD's. A small test data set was used to demonstrate the model and illustrate its use as a decision support system. The software developed in this thesis is available from the thesis advisor upon request.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The Naval Facilities Engineering Command (NAVFAC) is responsible for the maintenance and construction of the Naval shore establishment world wide. The current plant value of the Navy's buildings and utilities is $78 billion (Hollinberger, 1988, p. 5). A significant portion of the Navy's annual resource allocation is devoted to the shore establishment and managed by NAVFAC. It is essential that managers be provided with current, accurate financial information as the basis for resource allocation, budget execution, control, and evaluation decisions. This thesis addresses the feasibility of developing a personal computer based, decision support system to meet these needs.

At the field activity level public works centers and public works departments manage the resources necessary to maintain the shore establishment and provide related support. Public works centers, as navy industrial fund activities, have certain unique characteristics that distinguish them from public works departments. The centers are structured to restore the profit motive, which is absent from departments, and to capitalize on the economies of scale which can be recognized through consolidation in certain areas with a high concentration of navy facilities. The size and nature of their business requires sophisticated, computer supported

management systems. The use of such systems is well established for public works centers.

In the absence of the feedback afforded by the profit motive, public works department managers face additional challenges in financial management. In this environment, providing managers with the proper information support is absolutely essential. Computer information systems are playing an increasingly important role in this area. Large and medium size public works departments are able to support the acquisition of such vital systems. However, small public works departments are generally not large enough to support the procurement of these sizable and expensive systems.

Small public works departments manage multimillion dollar annual budgets and face the same type of decision making problems as do their larger counterparts. However, much fewer resources are available to support these decisions in small departments. Recent technological advances in personal computers and related software, along with their now routine existence in small public works departments, provide the necessary tools to address this crucial need.

A. BACKGROUND

1. The External Environment: Technological Development

The continuing rapid development of computers and information technology creates the opportunity for monumental changes in general administrative and decision making processes. Recent technological advancements in personal

computers allows individuals and small businesses to capitalize on these benefits at a nominal cost. The introduction of the 80386 microprocessor by Intel Corp. in 1986 brought the equivalent computing power of a 1980 mainframe computer to small desktop personal computers (PC's) and even briefcase size laptop computers. The exponential growth rate in technology promises even greater progress in the near future. Intel Corp. plans to begin production of its new 80486 microprocessor in the fall of 1989 (Brandt, 1989, p. 123). Meanwhile new advances in reduced instruction-set computing (RISC) chips, optics, computer neural networks, artificial intelligence, the ability to link and network PC's, and the ability for PC's to interface with mainframe computers all attest to the increasingly important role that PC's have in the office environment.

Along with the phenomenal development of computer hardware is an equally remarkable development of computer software. The usefulness of PC's has been greatly extended as word processors, data base management systems, and spreadsheets have rapidly evolved into powerful, integrated computing tools. In addition to these traditional workhorses, the PC software arsenal now includes access to sophisticated programming languages, integrated graphics environment management systems, and multitudes of specialized applications software. Data can be scanned in from existing documents,

read in from bar codes, or even entered by a recorded human voice.

This explosion in technology has enabled today's small businesses to exploit computers in ways that yesterday's industrial giants armed with supercomputers could only dream of. A PC user today can perform complex computing tasks by pressing a few keys or manipulating a mouse. Similar tasks in the past would have required pages of program code written by highly trained specialists. These capabilities allow individual users to easily develop their own models and applications. Given access to a PC, an employee with little computer training can develop and implement a customized system to automate most routine functions.

Along with the countless benefits this technology provides is a serious danger. With the spread of individual PC's comes the spread of individual data bases, systems, and methods. If this is not carefully controlled it can lead to inefficient duplication of effort and the promulgation of incomplete or inaccurate results. The proliferation of independent systems threatens to negate the potential benefits of this important technological advancement.

2. The Internal Environment: Acquisition & Development

The Navy has taken a cautious approach to introducing personal computers into the field activities. Field activities have historically been constrained by low dollar

ceilings on OPN funds[1] and centralized control of computer procurement. By the mid 1980's; however, PC's became generally available to most small Navy public works departments (PWD).[2]

The effort to obtain PC's in the public works departments was achieved on an individual basis over a period of time. No effort was made to provide a central purpose nor to develop any standard software or application modules for these computers. Consequently, individual users began to experiment with various uses for their newly acquired equipment.

During the same time period that PC's were being acquired by PWD's, NAVFAC began developing standardized public works management modules for use on a minicomputer system. However, financial modules were not addressed in this system. Therefore, financial and accounting functions were natural targets for individually developed PC applications. As software improves and user experience increases, so does the number of independent solutions to this common problem. To date no centralized approach to the design of financial

---

[1]Other Procurement Navy (OPN) funds finance the procurement, production, or modernization of equipment not otherwise provided for. OPN ceilings are Congressionally established and controlled. In recent years they have ranged from $1000 to $15,000.

[2]Small Navy public works departments are generally defined as those with fewer than 75 personnel in the maintenance and utilities divisions combined. For the remainder of this thesis small Navy public works departments will be referred to simply as public works departments or PWD's.

accounting and reporting applications modules for PWD's has been promulgated.

B. OBJECTIVES

This research examines the design of financial accounting and reporting systems for the PWD from two perspectives. At the first level, the system must automate the routine accounting and reporting functions of the PWD. From this perspective the system should provide increased efficiency and reliability. At the next level, the system should be useful to public works managers for decision making and control. From this perspective the system should enhance the effectiveness of public works management. This research effort will result in the specification of a decision centered design and a usable financial accounting and reporting model for PWD's.

C. THE RESEARCH QUESTIONS

This thesis attempts to establish the criteria for design of a PWD financial accounting and reporting system. The primary research questions are:

- Can a generally applicable model for financial accounting and reporting be developed for PWD's using a commercially available decision support system generator such as Lotus 1-2-3?
- Can such a model be used to apply decision support system theory to financial management within the PWD?

D. SCOPE, LIMITATIONS, AND ASSUMPTIONS

1. Scope

This research will result in a the development of an automated financial accounting and reporting system for a PWD. The emphasis of the research is to specify the needs of the PWD and the applicable criteria for design of any such system. The system will perform two primary functions:

- To provide the necessary accounting data for external users including the station comptroller, major claimant, and engineering field division.
- To provide timely financial information for decision making and management reports and graphs useful for department evaluation and control.

This research effort will emphasize incorporating the public works manager's perspective in the design of the model. Improving the accuracy and efficiency of the accounting and reporting process is a necessary and important objective for development of the model. However, the focus of the research is on designing a system which will improve the effectiveness of management decision making.

2. Limitations

Decision support system (DSS) theory emphasizes two important considerations which cannot be addressed in this research. System implementation is integral to a DSS design strategy. DSS theory highlights the necessity of properly managing change for such an implementation to be successful. It stresses interaction with the users starting in the predesign phase and continuing throughout the process. Due

to time limitations, it is not possible to implement a system within the scope of this thesis.

DSS theory also gives recognition to the fact that the design of any such system is an evolutionary process. The process of describing the decision making process, identifying key decisions, developing systems to support these decisions, and using the systems that have been developed is iterative by nature. Once the system has been used for some time new needs and applications will become apparent and modifications will be desirable. Users will identify new ways to improve their interaction with the system. As processes and decisions are better understood it will be possible to incorporate new functions in the system. This important part of the process is, of course, beyond the scope of this thesis but hopefully will be carried out by future users and researchers who may find the system useful.

3. Assumptions

For the reasons cited above, any model developed during this research can only be regarded as a first step in an on-going process. Therefore, this model is best described as a prototype. It is assumed that, if this model is deemed useful by those who may ultimately use it, the process will continue. The proper implementation and further development of the system is essential to completing the process and realizing the potential benefits of such a system. Therefore,

maximum flexibility for future modification and expansion are critical elements of the system design.

E. ORGANIZATION OF STUDY

The framework for systems development will be established using decision support system theory. Chapter II will present the DSS literature review used to build this framework. Key personnel in the Naval Postgraduate school (NPS) Public Works Department will be interviewed to provide the basis for defining basic data and system requirements. The NPS PWD has developed a database model, using Enable software, to automate many of the routine accounting functions. This system will be used as a point of departure for development of a DSS model. The Navcompt Manual and Public Works Manual will be used to generalize these requirements. Chapter III will present the results of this effort. Chapter IV will be used to present the resulting model and demonstrated its use. Chapter V will provide two illustrations of how this system can be used for decision support. Finally, chapter VI will present the conclusions and recommendations resulting from this study.

II. LITERATURE REVIEW AND THEORETICAL FRAMEWORK

The development of financial accounting and reporting applications in public works departments has occurred in an environment of tight resource constraints. Large expenditures of time, manpower, or money towards developing these systems at the installation level may strain already scarce resources and significantly detract from a small department's current capability to accomplish its primary mission. Decisions to make such expenditures must carefully balance the demands of current operational requirements against the perceived future benefits of such a system.

It is generally not possible for an individual PWD to justify the large resource allocations associated with using systems development experts to design and implement a computer based financial system. Public works officers, recognizing the potential benefits of these systems, have used numerous innovative approaches to resolve this difficult dilemma. Due to the resource limitations on an individual PWD, the systems which are developed are generally site specific and tend to focus on automation with the goal of improved clerical accuracy and efficiency. Decision support systems theory offers an approach to the design of such systems which may prove useful in overcoming these resource constraints and improving the effectiveness achieved in the employment of

currently available computer technology to assist in both information processing and decision making processes.

DSS theory is a synthesis of technical, behavioral, and managerial perspectives. Its goal is to realize the synergistic effects of a coordinated application of these theories to a system design process in an organizational context. A great deal of work, over many years, has been devoted to advancing and applying each of these individual bodies of knowledge. However, the development of a comprehensive approach to decision making integrating these perspectives has not received as much attention nor progressed to the same level of maturity. DSS theory emerged as an approach to fill this void.

DSS initially developed around Morton's (1971) synthesis of the Carnegie Institute of Technology's 1950's and 60's studies of organizational decision making with Massachusetts Institute of Technology's 1960's work on interactive computer systems. This section discussing DSS theory is primarily based on work by Keen and Morton. (1978)

A. SYSTEMS DEVELOPMENT: A CONTRAST OF APPROACHES

Decision support system theory is best described as an approach -- an approach to improving decision making. It is a way of thinking about the decision making process which has many possible applications. It is, however, primarily concerned with the application of computer systems to improve decision making. Since it is an integrated strategy, it is

useful to consider the key contributions of the principal perspectives it seeks to integrate. Keen and Morton (1978, p. 79) have identified the following:

> ...six rather different viewpoints on the role of computers in organizations and suggested that we need to synthesize from these a strategy for information systems development that exploits their relevant strengths and stresses the managerial and decisionmaking perspective. The viewpoints are:
>
> 1. Computer science: creates technology, both hardware and software. This is a necessary but not sufficient contribution to information systems.
>
> 2. Management science: represents the analytical viewpoint in structuring problems and develops the models so often necessary to drive information systems.
>
> 3. Behavioral science: provides insights into the implementation process and the human and organizational context of the system.
>
> 4. Data processing professional: builds the application systems the organization finally uses.
>
> 5. Management: understands the realities of decision making and thus which systems can be effective.
>
> 6. Decision support: focuses attention on building systems in relation to key decisions and tasks, with the specific aim of improving the effectiveness of the manager's problem-solving process.

These six perspectives form the basis for the development of DSS theory. Keen and Morton describe the traditional perspectives of each of these groups in great detail. The emphasis of this description is the fact that each has developed somewhat in isolation. Each brings a unique approach to problem solving usually resulting in vastly different recommended solutions with significantly different results. DSS theory is built around the premise of

integrating the first five viewpoints within the framework of the sixth.

1. Information vs. Data

Traditional electronic or automatic data processing (EDP or ADP) systems have, as the name implies, focused primarily on data. Tremendous advances have been made in data base management systems (DBMS) and related technology. Both mainframe and PC users have access to large data bases with on line query capabilities. The use of relational data base structures and sophisticated DBMS's has greatly enhanced the development of computer based accounting applications. Moscove and Simkin (1987, pp. 185-188) describe the importance of DBMS's to these applications:

> In typical accounting applications, data-base management systems act as an interface between computerized accounting programs and the accounting data files....From a technical standpoint, data-base management systems are important to accounting information systems because they free the user from the mechanical aspects of file inquiries, file maintenance, file updating, and file reporting, and permit the user to concentrate on the uses of accounting data....Thus the user can spend less time on these functions and more time supporting managerial uses of accounting data.

Integrating the performance of many accounting functions using a single data base and automated processing significantly increases the data processing efficiency. However, DSS theory subordinates the goal of increased efficiency to that of improved effectiveness. From that perspective such technological advances can only be viewed as a means to an end. DBMS, therefore, contributes to a DSS to

the degree that it facilitates and supports managerial decision making.

This viewpoint is supported by Keen and Morton who draw and important distinction between data and information. Data is regarded as a mere collection of facts, while information is a collection of meaningful data. DSS focuses on providing these facts, or data, in such a way that they are useful for some specified purpose. The distinction may be a subtle one but it represents a shift in emphasis from the collection and manipulation of data to the decision making process itself. Thus Keen and Morton view such important improvements in data processing efficiency as necessary but not sufficient contributions to the total system development.

2. Descriptive vs. Prescriptive

Traditionally specialists in management information systems (MIS), operations research (OR), and management science (MS) tend to take a normative approach to systems development. This perspective supports the existence of a superior methodology which yields optimal results. If, in accordance with this viewpoint, optimal solutions can be obtained through certain methods, any other solution would by definition be suboptimal and therefore irrational. Keen and Morton (1978, p. 22) describe this tendency as follows:

> OR/MS and, to a lesser extent, MIS too often imply that any manager who is not "rational" and fully computerized is incompetent. This prescriptive viewpoint assumes that there is a right way to make decisions and that the analyst should therefore act as a missionary converting the ignorant and heathen. The DSS approach begins from

a descriptive perspective and has the additional merit of humility and curiosity about how and why managers behave.

This prescriptive focus which centers around the quest for optimality has proved very fruitful in developing methods and solutions for a wide range of problems. However, these are generally not the same types of problems that a DSS is primarily concerned with. There is a wide variety of problems for which there is no optimal decision. The "best" solution to some problems is defined by the political reality of the situation or the preferences of individual decision makers. The purely normative approach would suggest either that such problems don't really exist or that computers have no role to play in their solution. A preoccupation with prescriptive approaches can result in missing the opportunity to use computers to improve decision making for a whole range of problems faced by managers.

DSS theory stresses a balanced approach which begins with analyzing the organizational environment in which decisions are made and understanding the methods currently used by managers to arrive at those decisions. This understanding is essential to the DSS design process. This is the foundation on which the model for improved decision making is built. As Keen and Morton (1978, p. 77) state:

> Designers need to be sure that they understand the realities of the decision situation and that they have a useful service to offer -- they need a descriptive model as the basis for identifying a normative direction....[DSS theory] is about the design and delivery of systems to help decisionmakers; it seems essential to us that any system builder be as concerned with descriptive realism

> as with normative idealism....[A designer must] make sure she or he understands how managers do in fact make decisions rather than focus on the logic of how they should do so.

The design process developed from DSS theory uses this descriptive approach as the starting point from which normative models are built. The importance of this part of the process should not be overlooked. The implication is that in many types of decision making situations the initial descriptive work is necessary in order to shape the direction for development of a model.

3. Effectiveness vs. Efficiency

The traditional focus on data management has several consequences related to systems development, as discussed above. This primarily technological perspective has another consequence not previously discussed. It leads to viewing computer systems simply as mechanisms to improve efficiency. Well designed computer information systems may enable you to employ fewer clerks and bookkeepers, process more information, or generate reports and documents in a fraction of the time otherwise required. Far to often, however, the reams of computer reports produced by the system are either not understood, not desired, or simply too much for any manager to read or comprehend. This phenomenon resulted in managers of the 1980's coining the term "information overload". If such reports are just filed, thrown away, or passed from person to person without any real use, then the system, no matter how technically competent, is of little real value.

Efficiency is defined in terms of the ratio of inputs to outputs. It involves performing some task using minimal resources or producing maximal product for a given level of resources. An efficient computer system may generate more reports and process more information in less time and at less cost than other comparable systems. However, this says nothing of the usefulness of the output generated.

Effectiveness involves selecting and implementing action which leads to the desired results within the required amount of time. Effectiveness implies a causal relationship. Effectiveness is required for success. Efficient production of an inappropriate product will never be successful. In an ideal world systems would maximize both efficiency and effectiveness. However, in practical situations this does not appear attainable. According to Keen and Morton (1978, p. 7):

> There is often a conflict between efficiency and effectiveness. Effectiveness requires adaptation and learning, at the risk of redundancy and false starts. Efficiency involves a narrowing of focus and minimization of the time, cost, and/or effort required to carry out a given activity. It is essentially programmatic.

Therefore, a balance must be achieved between the pursuit of efficiency and the goal of effectiveness. The particular balance which is appropriate is entirely situation specific. The more stable, or structured, the environment the greater the degree to which decision making can be relegated to operating according to rules and executing standard procedures. In such an environment the focus can

properly shift towards efficiency. However, the more unstable, or unstructured the environment the greater the complexity of decision making and the more the emphasis must shift towards effectiveness.

DSS theory represents a decided shift of focus from the pursuit of efficiency toward the goal of effectiveness. It does not ignore efficiency but it presumes a certain level of efficiency is implicit in the process. The precise level of efficiency is defined by the particular circumstance and the need to go beyond the merely efficient towards a flexible system supporting effective decision making amidst changing requirements. DSS emphasizes developing a detailed understanding of the situation and focuses attention on improving the effectiveness of the decision making process.

B. MANAGEMENT'S ROLE IN THE PROCESS

1. Decision Making Models

DSS theory is based on a management orientation. It focuses on supporting and improving the management decision making process. The descriptive perspective requires designers and technicians to develop a detailed understanding of manager's existing decision making processes. It is based on this perspective that a system to support this process is developed. The system relies on a gradual migration towards a more prescriptive perspective aimed at improving the effectiveness of the decision making process.

Since the decision making process has such a vital roll in a DSS development, it is useful to consider the implication of various decision making models to systems development. Keen and Morton describe five decision making models based on the work of Allison (1971). According to Keen and Morton (1978, p. 63):

> It needs to be stressed that one's concept of the decision process largely predetermines both one's response to other people's logic, behavior, and opinions, and the strategy chosen for design and implementation of any aid to "improve" the quality of decisions....Because of the nultidimensional nature of decisionmaking, it is critical to diagnose which aspect(s) is the most pivotal in any situation....Here we are concerned that the potential contribution of each of these perspectives on decisionmaking is clearly recognized so that if a DSS is warranted, it will be implemented with a strong base.

Each of the five models and its implications are discussed below.

a. The Rational Manager View

The rational view of decision making is economic analysis. It purports that decision makers are rational and completely informed. When faced with a decision the rational manager will develop a set of alternatives and choose that alternative which maximizes the output for a given input. This view leads to a completely prescriptive perspective where traditional operations research approaches may be used to design systems which identify the optimal solution. DSS theory is not particularly relevant to situations in which complete information and unlimited analytical ability exist.

b. The "Satisficing" Process Oriented View

The satisficing view recognizes the informational and analytical limitations involved with most decision making situations. It is based on bounded rationality. When faced with a decision the satisficing manager tends to use heuristics or rules of thumb to arrive at solutions which are considered good enough in the present circumstances. Practical bounds, related to the capabilities and commitment of the decision maker and the resources available to draw from, are placed on the decision making process. By identifying the heuristics used by the decision maker and developing a mechanism to expand and improve on them, DSS theory provides an approach which is especially well suited to this frequently observed situation.

c. The Organizational Procedures View

The organizational process view focuses on the responsibilities and relationships of subunits of the total organization. Each organization has its own domain which is governed by a unique set of operating procedures. Problems are divided functionally and are resolved through the a competitive process which involves each subunit supporting their individual perspectives and goals. A traditional management information system may be used to improve the efficiency of the operating procedures within each subunit. However, a DSS approach may support integration of the individual subunits resulting in joint decisions which are

more compatible with the organization's, rather than the individual subunit's, goals.

d. The Political Process View

The political process view emphasizes the importance of politics in the decision making process. It holds that decisions are made through a process of conflict, bargaining, and consensus building. It recognizes the existence of multiple goals within an organization. Different individuals with different ideas and ambitions all impact the final decision to varying degrees depending on their power, persuasiveness, and alliances. The political reality of organizational decision making has traditionally been ignored by system developers. Although it is not always relevant to the design of a DSS, the political climate should be assessed to determine if in fact it should be accounted for during system development and implementation.

e. The Individual Differences View

The individual differences view stresses the uniqueness of the decision maker. It acknowledges that there is no single approach to decision making but that there is a multitude of approaches applied by different individuals in different situations. There are two subsets of this view. Cognitive complexity theory emphasizes an individual's threshold to assimilate and process information. It holds that too much information, or information that is too complex, is just as detrimental as insufficient information. Cognitive

style theory distinguishes between analytic and intuitive decision makers. Analytic decision makers structure the decision process in a systematic way while intuitive decision makers use diverse trial and error type approaches.

f. Implications for Design

Managerial style and organizational characteristics can significantly affect the decision making process. These factors must be considered when developing a DSS. This has two important implications for DSS design strategy. First, systems should be developed which complement existing management styles rather than attempting to correct them. System developers who incorrectly assume that their own style is the "right" style and design systems accordingly may find that managers ignore the system rather than change their style of decision making. Second, not only will these styles and characteristics differ between organizations but they will also differ over time. Therefore, a DSS must be flexible enough to support a variety of styles and operate in various organizational environments. A system rigidly designed around a single style or decision making process will be useful only for that limited set of conditions and will not be capable of responding to the dynamic changes which occur in all organizations over time. According to Laudon and Laudon (1988, p. 147):

> The design of information systems must accommodate these realities, recognizing that decision making is never a simple process. Information systems can best support managers and decision making if they are flexible, with

multiple analytical and intuitive models for evaluating data and the capability of supporting a variety of styles, skills, and knowledge.

2. The Manager's Role in System Development

The manager's support, input, and participation during the development of a DSS is essential to the success of the project. Clearly, to gain the managerial perspective so central to concept and purpose of a DSS, managers must take an active role in the process. The tremendous advantage of this approach is that managers are no longer just passive customers of the MIS department, they are able to shape the system to meet their specific requirements. The result is that managers receive a product that's not just "a" system, it's "their own" system. Keen and Morton (1978, p. 13) stated that:

> Perhaps the most practical aspect of the DSS approach is that it allows managers to initiate, design, and control the implementation of a system. That is, a DSS is built around a decisionmaking task and while the technical issues may be extremely complex, the main focus is managerial. One main Stumbling block to the diffusion of computer-based methods in organizations has been the dominance of the technical role. Managers have often had trouble in developing the computer resource in relation to their business and decisionmaking needs. They have often been passive, wary consumers, not initiators and innovators.

Systems developed in isolation by technicians will be ignored by the managers they are designed to support. The DSS approach is structured to capitalize on the synergy realized by merging the manager's knowledge about goals and requirements along with his vision for the future direction of the organization with the technician's knowledge of

hardware and software capabilities along with his skill in developing the best ways to employ them. According to Keen and Morton (1978, p. 1):

> A main argument of the DSS approach is that effective design depends on the technician's detailed understanding of management decision processes and the manager's clear recognition of the criteria for developing useful computer-based decision aids.
>
> DSS theory hinges on a meshing of these two crucial perspectives. A DSS cannot achieve the management, perspective so essential to its very existence, without the active participation of managers during the development process.

C. THE DSS FRAMEWORK

1. Focus on Decision Making Support

The DSS framework provides an approach to systems development that focuses on the management perspective. Keen and Morton (1978, p. 1) cite three objectives for DSS:

- Assist managers in their decision processes in semistructured tasks.
- Support, rather than replace, managerial judgment.
- Improve the effectiveness of decisionmaking rather than its efficiency.

These objectives imply a shift in focus from administrative and operational concerns to managerial decision making. This is an extension of the traditional framework for computer support. Attention is redirected, away from automating clerical functions and producing job schedules or production

reports, toward the more subjective types of decisions which require managerial judgement.

A descriptive approach is used to define the key decisions the system is to support and to identify the environment the system must operate in. The system is tailored to requirements of the specific types of decisions it is designed to support. Consideration must also be given to the organizational characteristics, and management styles that exist within the environment in which the system will be used. A generally applicable system must be compatible with a variety of styles and decision processes. Systems development should start from the manager's perspective, focus on the key decisions that managers must make and be developed within the context of those decisions. The system must support an evolutionary movement from this starting point toward the prescriptive position which is ultimately desired.

Managers, not the DSS, make decisions within the organization. The DSS is merely one tool available to the manager to support the decision making process. According to Keen and Morton (1978, p. 58):

> A DSS supports and does not replace the manager. This emphasis on enhancement of decisionmaking exploits those aspects of computers and analytical techniques that are appropriate for the problem and leaves the remainder to the manager. Most, if not all, of managers' key decisions tend to be fuzzy problems, not well understood by them or the organization, and their personal judgment is essential. It is not possible to think of a computer system replacing managers or most of their decisions. Of course, over time, as our level of understanding increases, it may be possible to take some problems that

> we now consider fuzzy and systematize them so that they can be delegated to a computer or a clerk.

In this framework, the manager and the computer form a combined system resulting in better decisions than if either were used alone.

2. Structured vs. Unstructured Decisions

The framework for development of a DSS is built around the three categories of management activity defined by Anthony (1965) and the distinction between programmed and nonprogrammed decisions put forth by Simon (1960). Anthony's three categories of management activity are strategic planning, management control, and operational control. Strategic planning involves defining the objectives of the organization and establishing policies to govern resource management in attaining those objectives. *Management control* is the process by which managers ensure that resources are obtained and used effectively and efficiently in compliance with the established policies and objectives. Operational control is the process of ensuring that the specific tasks and priorities identified by management are carried out in an effective and efficient manner.

Simon defines programmed decisions as those which can be solved by applying predefined procedures. They are recurring, structured types of problems which do not require a unique decision making approach. Nonprogrammed decisions cannot be solved by the use of standard operating procedures. Their structure is not easily definable and they require a

unique analysis or a more intuitive problem solving approach. Keen and Morton retain this foundation in establishing a framework for DSS but they substitute the terms structured and unstructured for programmed and nonprogrammed in order to emphasize their relationship to the decision making process rather than to computer applications. Additionally, they define a third category, semistructured decisions, which is an intermediate between the other two.

Structured decisions do not require managerial judgement. They can be governed by standard operating procedures, delegated to subordinates, or automated through computer solutions. Unstructured decisions require managerial judgement. They may currently be too complex, too unique, or too poorly understood to permit defining any meaningful structured decision making process. The third category, semistructured decisions, is where a DSS can be most useful. These decisions are not structured enough to allow for delegation or automation, but there is sufficient definable structure to permit the use of a DSS. The DSS may provide decision making support by supplying needed information in the desired format, performing complex computational requirements, or implementing problem solving routines selected by the decision maker. However, some degree of managerial judgement will be required to reach the final decision. This judgement may be provided in the form of interpreting information supplied by the DSS, providing appropriate input data, or

selecting the proper solution models to be used. In any case, the interaction between the manager and the DSS is expected to produce more effective decisions than could otherwise be obtained.

Keen and Morton developed a two dimensional table that is useful for integrating and visualizing the application of these two classification schema to the DSS framework. Table 1, below, depicts the DSS framework described in these terms.

TABLE 1. A FRAMEWORK FOR INFORMATION SYSTEMS

	MANAGEMENT ACTIVITY			
TYPE OF DECISION	**OPERATIONAL CONTROL**	**MANAGEMENT CONTROL**	**STRATEGIC PLANNING**	**SUPPORT NEEDED**
Structured	Inventory reordering	Linear programming for manufacturing	Plant location	Clerical EDP or MS model
Semistructured	Bond trading	Setting market budgets for products	Capital acquisition analysis	DSS
Unstructured	Selecting a cover for Time Magazine	Hiring managers	R & D portfolio development	Human intuition

SOURCE: KEEN AND MORTON, 1978, p. 87

D. A FRAMEWORK FOR DSS DESIGN

1. Requirements of the DSS

The approach developed by Keen and Morton suggests that a DSS should be tailored to specific decisions and organizational environments. This approach is designed to encourage users to take an active roll in the development process and to increase the probability of ultimately developing a system which will be used and valued by those it is designed to support. This concept is fundamental to DSS theory. However, this approach must be balanced with the need to develop a system which is generally applicable to more than one specific decision process. Every DSS must operate in an environment of numerous management styles and constantly changing requirements. The very nature of the DSS implies evolutionary development in sophistication of the user; and therefore, increased demands on the system. The DSS must be designed in such a way that it can respond to these demands. In the extreme, a DSS designed specifically for one decision process would have to be redesigned each time any variable in the process changed. Therefore, the designer must balance these two competing goals.

John L. Bennett (1983, p. 18) argues that:

> ...different types of decisions have different data processing requirements. That is, a structured, operational control decision has different requirements than a semi-structured one, and so on. For example, strategic planning decisions tend to require more varied, more aggregated, and more qualitative data than do management control decisions. And structured decisions tend to utilize more data transformations than do

> unstructured decisions....If a DSS is designed for a specific type of decision, any change in the type of decision requires a change in the DSS to accommodate changes in data processing requirements. Therefore we conclude that designing a DSS for a specific type of decision reduces the number of decisions it can support and leads to increased cost if there is a change in the type of decision it is intended to support.

Thus, a DSS should support multiple processes and different types of decisions with different data processing requirements. This argument is not inconsistent with the view presented by Keen and Morton. It is a logical extension of that view which clarifies how to establish the balance between tailoring the system to specific user requirements and making the system flexible enough to survive change. This section is based primarily on the work of Bennett (1983).[3]

Bennett develops requirements for DSS design based on five observations related to management activities. The first observation is that management activities can be classified according to the three categories set forth by Simon (1960). These categories, intelligence, design, and choice provide a useful frame of reference for determining what operations the DSS should perform. Intelligence operations involve gathering information which helps to identify and define a problem or illuminate the relevant aspects of it. Design operations are concerned with developing alternatives and choice operations

---

[3]Bennett's second chapter, which establishes the framework for DSS design used in this thesis, is a version of an article by Eric D. Carlson, originally appearing in "Proceedings of Eleventh Annual Hawaii International Conference on Systems Sciences," 1978, published by Western Periodicals, North Hollywood, California.

involve evaluating and selecting from among those alternatives. The DSS should support all three types of management activities.

The four remaining observations lead to four additional requirements for a DSS. By definition, it is difficult to describe the decision making process for semistructured and unstructured decisions. The DSS should provide representations which help the manager conceptualize the problem. The diverse responsibilities of most managers requires that they deal with a multitude of problems at once rather than allowing them to focus on a single problem at a time. Managers generally rely on a variety of memory aids to help them keep track of their many responsibilities. The DSS should provide the manager with useful memory aids in a familiar format. The DSS should assist managers in using their own management style rather than constrain them by designing a single style into the system. Finally, managers are accustomed to exercising a great deal of personal control over the types of decisions that DSS are designed to support. The DSS should, therefore, provide control aids which help the manager understand the system and interpret its outputs, thus allowing for effective control of the support system.

2. The DSS Design Framework

The requirements described above form the basis for establishing a framework for DSS design. They suggest that the design should not focus on a single decision making

process but should instead provide representations which may be useful for a variety of processes. This type of representation based approach lends itself particularly well to the semistructured and unstructured decisions typical of strategic planning and management control. The representations help managers to conceptualize the problem and provide a context for decision making. In terms of Keen and Morton's two dimensional framework for information systems, the representations help define sufficient structure for unstructured problems to move them toward the semistructured category and thereby make it possible to benefit from the application of a DSS.

Bennett's framework for DSS design was developed based on the requirements discussed above. This framework can be described in terms of four basic questions to be used in DSS design:

- What specific **representations** should be used to enhance conceptualization and provide a frame of reference for using the DSS?
- What **operations** should be used to support intelligence, design and choice activities?
- What type of **memory aids** should be provided?
- What type of **control aids** should be provided?

This framework is used as a focus for systems analysis, as well as a means of structuring the actual design of the DSS.

Representations form the foundation of the DSS design framework. They aid in conceptualization of the problem and provide a context to invoke operations and interpret results.

They may take the form of graphs, charts, tables, equations, or reports. Operations are commands the user selects to execute a desired intelligence, design, or choice activity. Operations compute, modify, analyze, and summarize information contained in representations. It may involve only simple modifications to an existing representation or it may employ a powerful problem solving model to generate alternative solutions. Representations and operations are the core components of the DSS.

Bennett argues that the systems analysis and development should be representation based, not process oriented. A process oriented development would result from a design based on a flow chart of the decision making process. The resulting DSS would likely be process specific thereby reducing its applicability. The representation based approach may use similar techniques to select appropriate representations and operations but they will not be tied to any specific process. This results in a much more generally applicable system than the process oriented approach. However, the increased generality also implies a system which is somewhat more difficult to use since the user must develop processes to employ the system. Memory aids and control aids function to help overcome this difficulty.

Bennett describes seven types of memory aids that can be useful in a DSS (1983, p. 26-27).

- A **data base** contains information from both internal and external sources. "An extracted data base is a memory

for data compiled from sources the decision maker thinks may be relevant to the decision."

- **Views** contain subsets of information found in the extracted data base. This information is partitioned and stored in ways the decision maker thinks may be useful for either making or representing the decision.
- **Workspaces** are temporary storage spaces for accumulating the intermediate results of operations on representations.
- **Libraries** are long term storage spaces for workspace results which may be useful in the future.
- **Links** are storage spaces for data from a workspace or library which may be needed as input for another workspace.
- **Triggers** either invoke necessary operations automatically when needed or prompt the user to do so.
- **Profiles** provide information on system defaults or the status of operations and execution.

These memory aids help to simplify the operation of the DSS by reducing the amount of extraneous information the user must retain. In form, they should closely resemble the manual types of memory aids used by managers so that they provide a familiar, user friendly interface for interacting with the DSS.

Control aids are intended to assist the user in developing decision making processes with the DSS. They also provide the means for the manager to exert direct personal control over the decision making process. These may include the use of menus or function keys; on-line help facilities, instructional prompts, and natural language error messages; default settings or algorithms and the ability to change those defaults; and, programming facilities to allow the user to

modify program steps, define and automate decision processes, and add routines. Together, the memory aids and control aids provide the framework for developing the user interface. User involvement is necessary to develop appropriate memory and control aids. This is a critical part of the design process for a representation based design. If the memory and control aids do not provide the mechanism to easily develop decision processes with the DSS, the system may prove too cumbersome for practical use.

E. SYSTEM INTERNAL CONTROLS

Internal controls for computerized accounting systems is the subject of Statement on Auditing Standards No. 48, issued by the American Institute of Certified Public Accountants. This statement defines internal accounting control procedures by classifying them as either general or application control procedures. General controls are those controls that relate to such things as personnel, file security, system development and maintenance, and the use of internal audits. Application controls are sometimes referred to as transactions controls since they focus on the process of recording, manipulating, and using data generated from accounting transactions. Application controls are an important consideration developing a DSS which uses accounting information. Transaction processing systems, which may be separate from or an integral part of the DSS, frequently generate much of the data used in financial decision making. Application controls can

contribute to significantly improved decision making by increasing the quality of the data provided to the decision maker. Therefore, careful consideration should be given to what types of application controls need to be designed directly into the system. This section is primarily based on the work of Moscove and Simkin (1987) who categorize application controls as one of three types: input controls, processing controls, or output controls.

1. Input Controls

Input controls are those that check the validity and completeness of data before it is entered into system data files. This is the most important type of application control from the standpoint of system design. The most efficient time to detect and correct data errors is before the data enters the system. The alternative is to screen large amounts of data during processing or after processing is complete. Both efficiency and reliability are likely to suffer if either manual screening or after the fact correction of large amounts of data are required. Therefore, emphasis in system design should be placed on detecting data errors at the point of entry where they can be quickly and easily corrected before compromising the integrity of data files. Three important types of input controls will be considered: 1) data transcription, 2) edit tests, and 3) access control.

a. Data Transcription

Data transcriptions is the process of preparing data for computer processing. This is normally done by key punch entry using a computer keyboard and terminal display. One important input control is the use of **preformatted screens** to guide the user through the transcription process. The input screens tell the user what data to enter and should be compatible with source documents to facilitate data input. The input screens may make use of underlines, masks (blinking boxes), or reverse video to help the user know where to input data and how many characters are required. This type of control is the computer equivalent of well designed, preprinted recording forms used in manual accounting systems.

b. Edit Tests

Edit tests examine the contents entered in a data field to determine whether or not it meets some predefined standard of data quality. Since these tests are a type of input control, data which does not meet one or more of the standards is rejected and the user must correct it before it is accepted. There are many potential kinds of tests which could be preformed as edit tests. Several of these are listed below:

- **Tests of Field Content** check for numeric, alphabetic, or alphanumeric field characters in a data field.
- **Tests for Validity** check a data field against an authorized list of acceptable entries.
- **Tests of Reasonableness** check to ensure data is within some range of reasonable possible entries.

- **Tests of Completeness** check to ensure that no fields which require data are left blank.
- **Tests of Consistency** check to ensure that any data which has a related data field is entered consistent with those predefined relationships.

In addition to the edit tests mentioned above, a system of **check digits** can be used to detect transcription errors in long numerical codes. Check digits provide a means of detecting certain types of errors which would not be discovered using other edit tests. There are many ways to use check digits but one of the most common is to add a last digit to the numerical code which is equal to the sum of each of the other digits. The input control would consist of recomputing this sum based on the data keyed in to the input screen and checking to see if it matches what was entered. Check digits cannot detect all types of data entry errors but they are a powerful tool to detect most common transcription errors for important numerical codes. However, substantial additional effort may be required to generate codes with check digits, and therefore, they should only be used if detecting these sorts of errors is critical.

c. Access Controls

The primary type of access control available to system designers is the use of **passwords** to limit access to specific data, files, or systems. This simple control can be very important if data is sensitive or if it is essential that

only certain skilled users be allowed to manipulate the database in order to maintain its integrity.

2. Processing Controls

Processing controls are those which are concerned with manipulation of data after it has already been entered into the system. Two processing controls which may be useful to designers of financial DSS are **control totals** and **record counts**. To use control totals, the user calculates the sum of any relevant numerical field before inputing a group of records. During the processing of that group, the computer calculates the same sum for all the records processed. A mismatch between these two numbers indicates that a processing error has occurred. The user must then try to identify the source of the error. To use a record count, the user simply counts the number of records in a group before processing and this number is checked against the computer generated count of the actual number of records processed. Although record counts are slightly easier to use than control totals, they do not in any way check the results of the actual data processing.

3. Output Controls

Output controls are concerned with the results of the data processing. They involve procedures designed to detect errors in output files and reports. Validating processing results may be accomplished by some form of **activity listing.** These listings provide information about any changes made to

databases or files. In large systems, a detailed activity listing may be not be practical. For these systems, some type of summary listing may be more useful. The type of information which may be recorded in these listings includes:

- number of records at start
- number of records added, modified, and deleted
- number of records at end
- date of last update
- beginning and ending balances or totals

In some cases the distinction between processing controls and output controls may not be very great. The information contained in each may, in fact, be exactly the same. The time at which that information is available to the user distinguishes processing controls from output controls.

III. ACCOUNTING FRAMEWORK AND DATA REQUIREMENTS

In this chapter, the governmental accounting system will be examined from the perspective of the Navy public works department (PWD). The PWD is involved in only a certain portion of the total accounting system. The boundaries of its responsibilities must be defined since this will determine the data requirements for its financial systems. Therefore, the system data requirements will be discussed in the context of the accounting framework.

A. BACKGROUND

Navy public works departments play a large role in managing the financial resources of Navy shore commands. Roughly one half of the annual operating budget for a typical shore activity is provided to cover the costs of civilian labor and the other half is primarily to cover the procurement of goods and services. As one of the largest departments at a typical shore activity, the PWD is responsible for a substantial portion of the civilian work force and normally manages about one half of the non-labor funds in the operating budget. Additionally, the PWD is normally involved in managing a high dollar volume of funds from sources other than the station's operating budget. Tenant commands and other governmental agencies may utilize services of the PWD on a reimbursable basis, Major Claimants provide funds for special

projects to be executed by the PWD, and the Naval Facilities Engineering Command (NAVFAC) provides funds for PWD management of navy family housing. PWD's must maintain detailed accounting records in order to properly fulfil their responsibilities for managing the numerous assets entrusted to them.

The condition of navy shore facilities is a high interest item which receives considerable attention from the Chief of Naval Operations, Secretary of the Navy, Secretary of Defense, and Congress. Therefore, a higher level of detail is usually required on budget submissions from facilities management functions. Beginning with the Defense Appropriation Act of 1963, the congress has included a statutory requirement that a specified minimal portion of the Operations and Maintenance, Navy appropriation be used only for the maintenance of real property facilities.[4] The Secretary of Defense defined cost elements for control of maintenance floor costs to comply with this statute. The importance of the facilities management function, recognized throughout all levels of government, further emphasizes the need for detailed, reliable accounting records and effective financial management at the navy PWD.

---

[4]Real property includes both class 1 (land) and class 2 (buildings, structures, and utilities) plant property. The amount of funds expressly designated by congress as "only for maintenance of real property" is known as the "maintenance floor." The maintenance floor is available for maintenance, repair, and alterations (Functional Category Codes M & R) of class 1 and class 2 plant property.

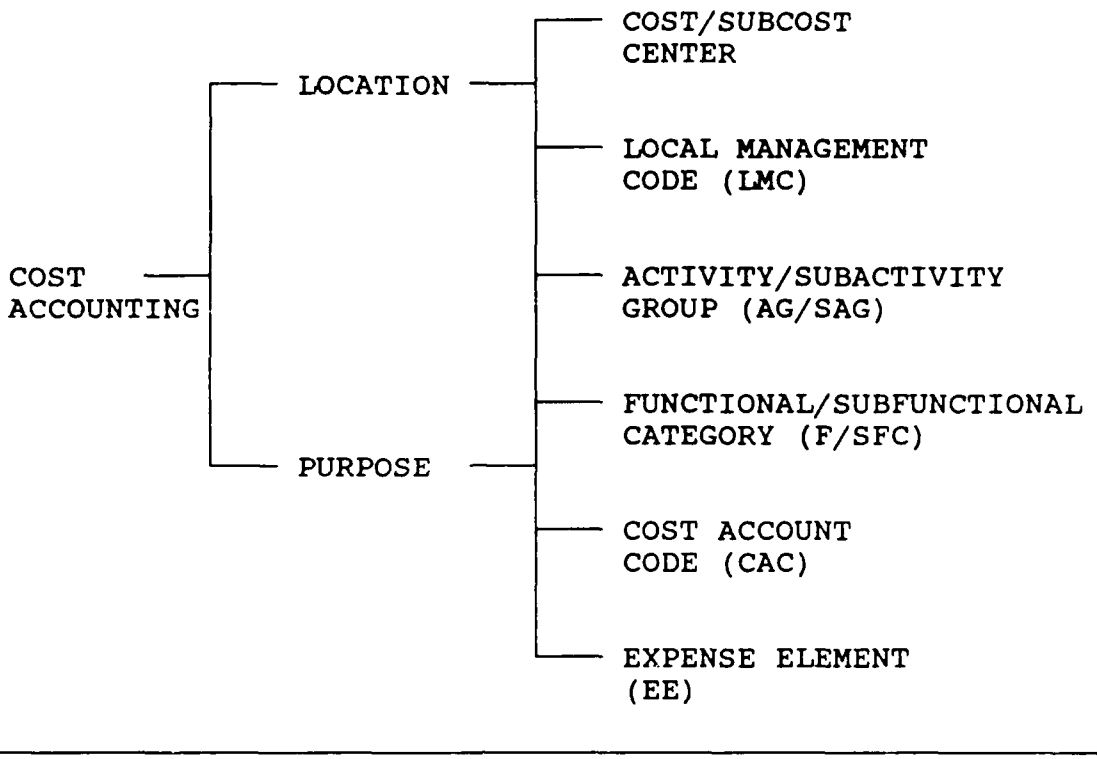
B. COST ACCOUNTING

Data from the cost accounting system is a vital part of financial management within the PWD. Cost accounting data is used for such purposes as budgeting and resource allocation decisions, management control, performance evaluation, and reporting to higher authority. In order to provide the information necessary to fulfil all of these diverse purposes, a very detailed system of cost accounting is required. The basic cost accounting structure used for this purpose is shown below in Figure 1. Funds used for civilian labor are provided and accounted for separate from non-labor funds. In the PWD labor costs are traced to specific work centers within the department; however, material and contract costs are more associated with particular facilities and the purposes for which they were used. Therefore, slightly different accounting schemes are used for labor and non-labor costs to focus on the relevant portions of the cost accounting structure.

1. Labor Accounting

The primary purpose of cost accounting for labor within the PWD is for management control and performance evaluation. To accomplish this, labor funds are accounted for in three ways:

- **Cost/Subcost Center (work center)** - a Responsibility center is the shore activity receiving the operating budget. The responsibility center is broken down into cost centers, departments within a command, and subcost centers and a code is assigned to identify each. In



Source: Practical Comptrollership p. D-12

Figure 1. Cost Accounting Structure

the PWD this is further subdivided and cost accounting by location is accomplished primarily through use of a work center code. This identifies the specific suborganization within the PWD that generated the labor cost (e.g. code 750, maintenance division, electric branch)[5]

[5]The PWD must be able to compare planned and actual manhours for each job by work center and branch for management control purposes. This information is available through records kept in the maintenance control division or under the Base Engineering System, Technical (BEST) management information system. Therefore, it is not duplicated at that level of detail in the accounting records of this system. The accounting system should, however, be able to generate variance reports to identify those jobs which require detailed examination. It seems desirable, and may be possible in the future, to further integrate the maintenance management and accounting systems.

- **Local Management Code (LMC)** - this code allows an activity to further breakdown the cost/subcost center by organizational structure or purpose. This may or may not be used at a particular activity.

- **Labor Class Code (LCC)** - a two digit numerical code used to classify jobs according to various categories of overhead and productive work. Productive work is categorized primarily by the level of management control required. At the lower levels of control data from individual jobs may be aggregated and controlled primarily as a group. At the higher levels of control, normally necessitated by large resource expenditures or for funds provided for a specified purpose, jobs must be controlled individually. Labor class codes are shown in TABLE 2 below.

2. Non-labor Accounting

The primary purpose of cost accounting for non-labor expenses in the PWD is for management control and budgeting. To accomplish this, expenses are accounted for in a tiered schema where each tier provides additional detail about the purpose of the expense (refer to Figure 1).

- **Activity/Subactivity Group (AG/SAG)** - a two digit code which identifies the functional area funds are provided to support. This is primarily used by major claimants for administration of operations and maintenance funds. Some major claimants may control the number of civilian personnel billets authorized in each SAG.

- **Functional/Subfunctional Category (F/SFC)** - a two digit alphanumeric code, the first of which is a letter identifying the functional category and the second a number, or letter, identifying the subfunctional category. A list of functional category codes for naval shore activities and a list of facilities management subfunctional categories are shown below in TABLES 3 and 4 respectively.

- **Cost Account Code (CAC)** - a four digit code used with subfunctional categories to specify the detailed purpose of an expenditure. Cost account codes are prescribed in NAVCOMPT Manual, Volume 2, Chapter 4, Part D.

TABLE 2. LABOR CLASS CODES

DESCRIPTION	LCC
PRODUCTIVE WORK:	
Service Work	01
Emergency Work	02
Preventive Maintenance Inspection	03
Standing Job Orders, Not Estimated	04
Standing Job Order, Estimated	05
Minor Work	06
Specific Job Orders	07
DIRECT SHOP OVERHEAD:	
Rework	40
Supervision	41
Shop Indirect	42
Allowed Time	43
General Office and Clerical	44
Leave	45
INDIRECT OVERHEAD:	
Facilities Management Engineering	60
Engineering Support	61
Administrative Support	62
Indirect Overhead Leave	63
Facilities Support Contract Administration	64
Special Projects Development	65
Miscellaneous	66

SOURCE: NAVCOMPT MANUAL, VOLUME 3, CHAPTER 7, PART C

- **Expense Element (EE)** - a one digit alphabetic code which specifies the type of input used for the purpose specified by the cost account code. Every expense or obligation identified by functional/subfunctional category must also be identified by an expense element. A sample list of expense elements applicable to facilities maintenance is shown in TABLE 5 below.

Additional cost accounting information required by the PWD includes:

- **Segment Code (SEG)** - subfunctional category Z indicates reimbursable work. For example MZ is reimbursable maintenance of real property. Reimbursable jobs are accounted for individually, regardless of size, in the PWD by use of the segment code. Segment codes are normally established at the beginning of the fiscal year for each reimbursable account.

- **Public Works Code** - this code is provided for internal use of the PWD. Up to six alphanumeric characters may be used in this field. This field is intended to provide additional information about the use of funds for budgeting and control. It is suggested that this code include the two digit **Investment Category (IC)** code used to relate shore facilities to their Navy mission contribution.[6] There are 18 primary IC's, numbered 01-18, and a 19th for miscellaneous functions as shown in TABLE 6 below. Other budgeting and control information can be included in this field according to the needs of the individual PWD.

- **Material/Contract** - expense reporting and control requirements dictate that civilian labor, material, and contract costs be accounted for separately. A one digit alphabetic field is used for this purpose. An M is entered for all material costs and a C is entered for all contract costs.

Other information required to complete a journal entry in the public works accounting system includes:

- **Month** - in the accounting system, separate journals are maintained for each subfunctional category except that all reimbursables, including family housing, are maintained in a single journal. Reporting and control requirements necessitate that certain information be routinely extracted on a monthly basis. Those journals, telephone, utilities, and reimbursalble, are

---

[6]Budget submissions for Real Property Maintenance Activities (RPMA) are configured according to investment categories. The primary source of information for this purpose is the station Annual Inspection Summary (AIS) which identifies the backlog of maintenance and repair by investment category. Hollinberger (1988, p. 1) cited the need for accounting systems to describe resources spent by mission area. This will help relate current and past resource expenditures to mission readiness at the activity level as well as provide direction for future resource requirements.

TABLE 3. FUNCTIONAL CATEGORIES

FUNCTIONAL CATEGORY	CODE
Mission Operations	A,B,C
Administration	D
Supply Operations	E
Maintenance of Material	F,G
Property Disposal	H
Medical Operations	J
Oversees Dependent Education	K
Base Services	L
Maintenance of Real Property	M
Utility Operations	N
Other Engineering Support	P
Minor Construction	R
Personnel Support	S
Nontactical Automatic Data Processing Supt.	V

SOURCE: NAVCOMPT MANUAL, VOLUME 2, CHAPTER 4, PART D

TABLE 4. FACILITIES MANAGEMENT SUBFUNCTIONAL CATEGORIES

DESCRIPTION	SFC
Base Communications, Shore Activities	LA
Op. and Maint. of Transportation Equipment	L7
Recurring Maintenance	M1
Nonrecurring Maintenance	M2
Operation of Utilities	N1
General Engineering Support	P1
Minor Construction (within CO authority)	R1
Minor Construction (above CO authority)	R2
Personnel Support	S1

SOURCE: NAVCOMPT MANUAL, VOLUME 2, CHAPTER 4, PART D

TABLE 5. SAMPLE LIST OF EXPENSE ELEMENTS

DESCRIPTION	EE
Travel of Personnel	E
Utilities and Rents	M
Communications	N
Purchased Equipment Maintenance (commercial)	P
Purchased Services, Other	Q
Supplies	T
Civilian Personnel	U

SOURCE: NAVCOMPT MANUAL, VOLUME 2, CHAPTER 4, PART D

TABLE 6. INVESTMENT CATEGORIES

DESCRIPTION	IC
Aviation Operational Facilities	01
Communication Operational Facilities	02
Waterfront Operational Facilities	03
Other Operational Facilities	04
Training Facilities	05
Aviation Maintenance/Production	06
Shipyard Maintenance/Production	07
Other Mainter ;ce/Production	08
RDT&E	09
POL Supply/Storage	10
Ammo Supply/Storage	11
Other Supply/Storage	12
Medical	13
Administrative	14
Troop Housing/Messing	15
Other Personnel Support & Service	16
Utilities	17
Real Estate & Ground Structures	18
Continuing Authority	19

SOURCE: NAVFAC P-72 MANUAL

further subdivided and maintained as separate monthly journals. For these subfunctional categories it is necessary to enter the appropriate month to determine which journal the entry should be recorded in. Only the three letter abreviation for the month is required (e.g. JAN, APR, etc...) but the complete month name may be entered if desired. This is a vlaid field for all subfunctional categories but is only required for the three mentioned above. By always entering the appropriate month the database will support any desired future analysis by month.

- **Obligational Document Control Number (ODC)** - this is a key field in the jounal entry system since it is the single unique field used to distinguish journal entries. This is the unique document number on the applicable obligational document used to procure the materials or services in question.

- **Job Order Number (JON)** - job orders are used to authorize the expenditure of resources to accomplish a specified purpose. The job order number is a number assigned by the activity to fiscally distinquish that job from all other jobs. The job order number is used to relate the databases maintained under the jounal entry system for material and contracts with the labor distribution database maintained under the labor system for civilian labor costs. By relating these two databases complete job costs can be determined, variance reports can be generated, and other management decisions can be supported.

- **Contract Number** - this is the specific contract number assigned to facilities contracts.

- **Amendment Number** - after award, contracts may be modified by making amendments. The amendment may alter the amount of funds committed, obligated, or to be expended in the future. Administrative amendments which do not affect the status of funds are best handled by modifying the exising journal entry to keep track of the number of amendments to date. However, amendments which do affect the status of funds are actually separate transactions; and therefore, are best represented by a separate jounrnal entry. This allows you to keep track of the obligational status of the amendment as well as that of the original contract since the two will normally be different during contract execution. Complete data for any given contract can be obtained by aggregating records by contract number.

- **Description** - a 25 character field used to describe the particular materials or services being procured.
- **Contractor** - a 25 character field used to specify the name of the contractor or vendor providing the goods or services contracted for.

C. OBLIGATIONAL ACCOUNTING

Two types of accounting are used within the cost accounting structure, accrual accounting and obligational accounting. Under the accrual accounting system expenses are recognized at the time operations are incurred. Normally this is the time when goods and services are consumed. Matching expenses with the consumption of assets provides a useful measure of the actual costs associated with accomplishing various missions or providing various services. Obligational accounting is concerned with tracking the progress of budget execution during the limited obligational availability period of funds appropriated by congress. Obligational accounting is a useful tool for monitoring the expenditure of appropriated funds to ensure that they are used for the purposes set forth by congress within a specified period of time.

1. Non-labor Spending

The obligational accounting frame of reference is generally the most relevant for management purposes within the PWD, with the exception of labor accounting. Contracts for the procurement of goods and services are entered in the cost accounting system using the obligational frame of reference

which is tied to the source year of the appropriation. Funds received are accounted for by tracing them through the following three progressive levels of the spending process:

- **Committed** - Individual transactions are first reflected as a commitment when an activity requests that a contract be negotiated for the purchase of goods or services.
- **Obligated** - transactions are then reflected as obligations when a contract is signed for the purchase of goods or services.
- **Expended** - Finally, transactions ar reflected as an expenditure when funds are actually disbursed to pay the vendor for the goods or services contracted for.

Under this system, the accounting equation is expressed as:

- Authorization Held = Status of the Authorization.

This equation may be used to develop a simple dual entry accounting system as is illustrated in Figure 2 below. While there are certain obvious advantages to dual entry accounting, it is not necessarily required at the level of the PWD. In fact, small PWD's do not routinely use dual entry accounting; and therefore, systems should be designed to be flexible enough to accommodate various approaches.

2. Labor Distribution

The primary source document for the labor accounting system is the labor distribution card. On this card civilian employees record the number of hours worked by **Job Order Number**. It should be noted that the primary source document for civilian payroll is the time card. While the labor distribution card and time card must ultimately be reconciled,

#	TRANSACTION	DEBIT	CREDIT
(1)	Authorization Received	1000	
	Uncommitted Balance		1000
	receive $1000.00		
(2)	Uncommitted Balance	300	
	Outstanding Commitment		300
	request contract X for $300.00		
(3)	Uncommitted Balance	200	
	Outstanding Commitment		200
	request contract Y for $200.00		
(4)	Outstanding Commitment	300	
	Outstanding Obligation		300
	sign contract X for $300.00		
(5)	Outstanding Obligation	100	
	Expenditure		100
	disburse $100.00 for contact X		

ACCOUNT	BALANCE	
Authorization Received	$1000	(dr)
Uncommitted Balance	$500	(cr)
Outstanding Commitments	$200	(cr)
Outstanding Obligations	$200	(cr)
Expenditures	$100	(cr)

FIGURE 2. Dual Entry Obligational Accounting

the two documents serve different purposes and it is the labor distribution card which is relevant to cost accounting. The following data is used to account for civilian labor in the cost accounting system:

- **Regular hours** worked by each employee on each job.
- **Overtime hours** worked by each employee on each job.
- **Regular wage** for each employee - from employee records.

- **Overtime wage** for each employee - from employee records.

The social security number (**SSN**) of each employee is used to link labor distribution cards to employee records and labor costs are calculated for each separate job by multiplying the number of hours worked during the period (**PRD ENDING**) by the appropriate wage. The labor costs for individual employees can then be aggregated by job order number for use in management control reports, performance evaluation, and budgeting.

D. PUBLIC WORKS LEDGERS

Responsibility centers receive funds via operating budgets and allotments. Allotments are provided in specific amounts for a specified purpose. They are not part of the operating budget and must be accounted for separately. In the PWD special project funds, subfunctional categories M2 and R2, are received in this manner. Special project funds may only be applied to the particular project they were given to fund and the project cost cannot exceed the funds provided for it. Reimbursable funds are also provided to the PWD in a specific amount for a specified purpose and must be accounted for separately. Therefore, separate ledgers are maintained for each subfunctional category except that M2, R2, and reimbursable ledgers consist of a database which accounts for the status of funds for each uniquely identifiable source.

The data which must be input into the public works ledger system for this purpose is as follows:

- **Subfunctional Category or Segment (SFC/SEG)** - the subfunctional category code, or segment code, as described above, is entered here to define the appropriate ledger for recording the entry. However, since special projects, M2 and R2, are funded by individual project, the special project number must also be entered. The system will prompt the user for this information if required.

- **Annual Planning Figure (APF)** - resource authorizations for operating budgets are normally received by the responsibility center at the beginning of each financial quarter. However, at the beginning of the fiscal year the PWD is given an annual planning figure which identifies the amount of new obligational authority it can expect to receive over the course of the year. For special projects and reimbursables the individual amount of funds issued for each specific purpose should be entered here.

- **Quarterly Operating Target (QTR)** - as the responsibility center receives funds throughout the fiscal year, the PWD will be given current information on the amount of funds made available for obligation. This information is normally distributed in the form of quarterly operating targets or OPTAR's. The amount of funds made available each quarter should be entered in the appropriate quarterly field in the ledger system. Additional funds are occasionally received between quarters, at mid-year, or at the end of the fiscal year. These changes should be made in the ledger system by modifying one of the quarterly operating targets.[7]

E. EMPLOYEE RECORDS

Employee records are integrated with the accounting system for two reasons. The possible need to control civilian

---

[7]The annual planning figure (APF) is provided primarily for planning purposes. The actual total obligational authority will not be known until the end of the fiscal year. Therefore, the ledger system calculates unobligated balances by subtracting obligations for materials and contracts from the sum of the quarterly operating targets.

personnel billets by subactivity group (SAG) was mentioned above. However, even if this type of control is not required, the management to payroll system will require relating employee wages to available funds. Therefore, integrating the employee records with the financial system may serve to facilitate management of civilian personnel billets. Additionally, this integration eliminates redundancy in entering employee wages. By relating the employee data file with the labor card system, the need to enter wages with labor distribution data to calculate labor costs for each job is eliminated, thereby increasing efficiency and accuracy. Other data, not previously mentioned, which must be maintained for each employee is as follows:

- **Name** - **LAST**, 20 characters, **FIRST**, 15 characters, and **MIDDLE INITIAL (MI)** are entered for each employee.
- **Position Description** - a six character code used to identify the position the employee was hired to fill.
- **Series** - a four digit code used to categorize positions by function.
- **Plan** - a two digit code sued to indicate the payment plan applicable to the position (e.g. General Schedule, GS, or Wage Grade, WG)
- **Grade** - a two digit numeric code which normally follows the plan code. The grade indicates a level of seniority within the payment plan, for example a GS-13 position is considered senior to, and will pay a higher salary than, a GS-9 position, other things being equal.
- **Step** - a one or two digit numeric code which specifies a pay level within each plan and grade. An employee may receive a higher step within a particular grade due to such factors as level of experience, special training, performance ratings, or length of service.

- **Status** - a one to four digit code which indicates the employment status of a civil service employee. For career employees, more than three years, enter a C; for career conditional employees, less than three years, enter CC; for temporary employees, unspecified short term position, enter TEMP; and for term employees, specific purpose for a specified period, enter TERM.

- **Work Center** - normally a three digit code which specifies the particular suborganization, or work center, the employee is assigned to (e.g. code 720 - metal trades).

- **Title** - a 25 digit field which indicates the title of the position the employee is filling (e.g. Boiler Plant Foreman).

- **Pay Control Number** - a unique number, similar to a social security number, used to identify each employee for payroll purposes.

- **Social Security Number (SSN)** - the SSN is used to relate employee records to individual labor cards in order to calculate labor costs for each job.

- **Date** - the **HIRE DATE** and **TERMINATION DATE** are entered for each employee. It is important to note that employee records must be kept in the system at least until the end of the fiscal year for use in generating labor reports.

- **Fund Code** - for PWD's this code is used primarily to distinguish positions required to support reimbursable work from those required to support work funded by the station's operating budget.

- **Subactivity Group** - a two digit code which identifies the type of funds used to support each billet.

F. THE PUBLIC WORKS PLANNING SYSTEM

Job order planning data is entered into the planning system to generate a database which will be useful for decision making. This database should include all jobs under consideration for execution for a given fiscal year, regardless of the intended method of accomplishment (MAC).

The purpose of the planning system is to provide the decision maker with a scratch pad for planning along with the tools and models necessary to support various types of manipulation and analysis of the data to generate information useful to the decision making process. The principle types of decisions which this portion of the system is intended to support are related to resource allocation, performance evaluation, and management control. The data requirements in this system have been intentionally limited in order to minimize any redundancy relative to the Base Engineering System, Technical, or BEST, management information system. The data fields necessary to support basic financial decisions are as follows:

- **Job Order Number** - the job order number is entered here to uniquely identify the job in the system and to relate this database to the public works journals to support variance analysis.

- **Priority** - this numerical field is provided to enter the priority number assigned to the job order. This priority number is a key consideration in determining resource allocation. The priority may be assigned by a scheme similar to that presented in the NAVFAC MO-321, Public Works Manual, or by some other method.

- **Status** - this field is provided to distinguish jobs entered in the planning system for planning purposes, P, from those which have been selected for accomplishment during the current fiscal year, S, and those which have already been completed and closed during the current fiscal year, C. The list of completed job orders can be used to support an edit test of job orders entered in the journal entry system so that the user is not allowed to enter job orders which are already closed out. The list of selected job orders can be used to generate a report which projects the expected status of the authorizations based on current planning. This field is not intended for use in tracking the progress of job orders since there are maintenance control systems in place to accomplish that requirement.

- **Investment Category (IC)** - the investment category, as described above, is included in the planning system so that the decision maker can generate graphical representations of the level of resources committed to each investment category under various possible scenarios. This will assist the decision maker in selecting a course of action which best represents the station's needs at that time based on a concrete conception of the impact on various mission areas.

- **Planned Manhours** - estimates of the manhour requirements for each work center; **ELECTRICAL**; **MECHANICAL**, primarily metal trades; **STRUCTURAL**, primarily building trades; and, **OTHER**, any other miscellaneous work centers required for the particular job in question, are entered in this set of fields. This data can be used during planning for comparison with manpower constraints to assist in determining the feasibility of various possible plans.

- **Planned Costs** - estimates of the cost requirements of each job are entered in this set of fields. **TOTAL** costs are divided into **LABOR** costs, **MATERIAL** costs, and **CONTRACT** costs. This data supports decision making by providing financial constraints for each input category for use in assessing the feasibility of various possible plans. Additionally, this data can be used to compare with actual costs for variance analysis or performance evaluation.

Data definition tables are shown below in TABLES 8, 9, 10, 11, and 12 for each of the individual modules in the model. These tables indicate the *field name*, *field type* (A - alphabetic, AN - alphanumeric, or N - numeric), the number of characters displayed in the database, and any edit tests performed during data entry or modification. The relevance and relationships of the various data requirements presented in this chapter can be better seen by examining the input screens used in the three systems, **Accounting**, **Labor**, and **Planning**, and working with a sample data set to demonstrate their use. In addition to demonstrating the mechanics of

using the system, the next chapter will illustrate a few examples of how the system can be used to support decision making.

TABLE 7. LEDGER MODULE FIELDS

NAME	TYPE	DISPLAY	EDIT TESTS
QTR 1	N	14	NUMERIC ENTRY REQUIRED
QTR 2	N	14	NUMERIC ENTRY REQUIRED
QTR 3	N	14	NUMERIC ENTRY REQUIRED
QTR 4	N	14	NUMERIC ENTRY REQUIRED
SFC/SEG	AN	14	NOTES 1,2
APF	N	14	NUMERIC ENTRY REQUIRED

NOTES:
1) SEG codes checked against authorized list.
2) M2, R2, & REIMB. entries checked against ledger to ensure no duplicate entries.
3) No quarterly entries required for M2, R2, and REIMB funds.

TABLE 8. CODES MODULE FIELDS

NAME	TYPE	DISPLAY	EDIT TESTS
SEG CODE	AN	9	USER VERIFICATION

TABLE 9. JOURNAL ENTRY MODULE FIELDS

NAME	TYPE	DISPLAY	EDIT TESTS
ODC	AN	12	ENTRY REQUIRED
CONTRACT #	AN	12	
AMEND #	AN	9	
JON NUMBER	AN	9	ENTRY REQUIRED, NOTE 1
DESCRIPTION	AN	25	CANNOT EXCEED 25 CHAR
CONTRACTOR	AN	25	CANNOT EXCEED 25 CHAR
COST ACCT	AN	10	ENTRY REQUIRED
SFC/SEG	AN	9	ENTRY REQUIRED, NOTE 2
EXP ELMT	AN	4	
PW CODE	AN	4	
COMMITTED	N	12	NUMERIC ENTRY REQUIRED
OBLIGATED	N	12	NUMERIC ENTRY REQUIRED
EXPENDED	N	12	NUMERIC ENTRY REQUIRED
MONTH	A	9	NOTE 3
MATL/CONT	A	9	M OR C REQUIRED

NOTES:
1) Prompted for special project number for M2 & R2 entries.
2) SEG codes checked against authorized list.
3) First 3 characters must be from one of the 12 months for LA, N1, & REIMB entries.

TABLE 10. LABOR CARD MODULE FIELDS

NAME	TYPE	DISPLAY	EDIT TESTS
JOB NUMBER	AN	9	ENTRY REQUIRED
REGULAR HRS	N	9	NOTES 1,2
OVERTIME HRS	N	9	NUMERIC ENTRY REQUIRED
WORK CENTER	AN	9	
LMC	AN	9	
LCC	AN	9	
PERIOD END	AN	9	
SSN	AN	11	NOTE 3

NOTES:
1) Numeric entry required.
2) Cannot exceed 40 hours.
3) Checked against social security numbers in the employee records for validity.

TABLE 11. PERSONNEL MODULE FIELDS

NAME	TYPE	DISPLAY	EDIT TESTS
LAST	AN	20	ENTRY REQUIRED, NOTE 1
FIRST	AN	15	CANNOT EXCEED 15 CHAR
MI	AN	3	
POS DECSRIP	AN	6	
SERIES	AN	7	
REGULAR WAGE	N	9	NUMERIC ENTRY REQUIRED
OVERTIME WAGE	N	9	NUMERIC ENTRY REQUIRED
TITLE	AN	25	CANNOT EXCEED 25 CHAR
WORK CENTER	AN	5	
PLAN	AN	5	
GRADE	AN	6	
STEP	AN	5	
STATUS	AN	5	
PAY CONTROL	AN	12	
SSN	AN	12	NOTE 2
HIRE DATE	AN	10	
TERM DATE	AN	10	
FUND CODE	AN	5	
SAG	AN	4	

NOTES: 1) Cannot exceed 20 characters.
2) 11 characters with dashes, 9 without.

TABLE 12. JOB PLANNING MODULE FIELDS

NAME	TYPE	DISPLAY	EDIT TESTS
JOB NUMBER	AN	7	ENTRY REQUIRED
PRIORITY	N	6	NUMERIC ENTRY REQUIRED
STATUS	AN	6	
IC	AN	3	
ELECTRICAL	N	9	NUMERIC ENTRY REQUIRED
MECHANICAL	N	9	NUMERIC ENTRY REQUIRED
STRUCTURAL	N	9	NUMERIC ENTRY REQUIRED
OTHER	N	9	NUMERIC ENTRY REQUIRED
TOTAL	N	9	NUMERIC ENTRY REQUIRED
LABOR	N	14	NUMERIC ENTRY REQUIRED
MATERIAL	N	14	NUMERIC ENTRY REQUIRED
CONTRACT	N	14	NUMERIC ENTRY REQUIRED
TOTAL	N	14	NUMERIC ENTRY REQUIRED

IV. SYSTEM DEMONSTRATION

Lotus 1-2-3 Release 3.0 was used to develop a prototype financial Decision Support System (DSS) for U.S. Navy Public Works Departments (PWD). The system is composed of three subsystems, the Accounting system, the Labor system, and the Job Planning system. Each of these subsystems is located in a separate file and is composed of various modules which perform a set of related functions. The purpose of this chapter is to present the resulting model by demonstrating its use on a small test data set. The presentation, which will follow a tutorial format, is intended to demonstrate the major features and uses of the model. Therefore, the data and examples used have been simplified to emphasize features of the system and do not necessarily represent reality.

A. SYSTEM DESCRIPTION

1. THE ACCOUNTING SYSTEM

The accounting system is used to maintain a set of journals and ledgers for the PWD. It is composed of four separate modules. The Journal module is used to record transactions involving the procurement of goods and services. Account balances are automatically maintained in the Ledger module after beginning account balances are entered. The Codes module is used to enter a list of authorized segment codes for funds provided to the PWD for reimbursable work.

This list is used to validate segment code entries in the other modules. Finally, the Work module provides a work space for analyzing data and creating reports.

2. THE LABOR SYSTEM

The labor system is used to maintain information relevant to PWD labor expenditures. There are four modules in the Labor system. PWD employee records are maintained in the Personnel module. These records contain information about the employee's billet and wage structure. Labor distribution card data is recorded in the Labor module. Job order accounting for labor costs is based on the data contained in these two modules. A complete labor cost report, by job order number, is automatically generated by the Report module. Finally, the Work module provides a work space for analyzing data and creating reports.

3. THE JOB PLANNING SYSTEM

The job planning module is used to record estimated cost and manhour data on jobs planned for future execution. This information is used to support a variety of functions for PWD management including variance reporting and resource planning and allocation decisions. The first of four modules, the Jobs module, maintains the job estimate records used in this system. The Linear and Dynamic modules are to provide the capability of using linear and dynamic programming for decision support. These modules, which will be discussed

later in this Chapter and again in Chapter IV, are not installed in the Job Planning system at this time. The final module in this system, the Work module, provides a work space for analyzing data and creating reports.

Only a few of the many edit tests the system performs will be demonstrated or explained in the tutorial that follows. For a complete list of data requirements and edit tests refer to the data definition tables at the end of Chapter III. The tutorial will be organized around each of the three subsystems, with a separate section dedicated to each. Chapter V provides two examples of how to integrate the three subsystems and use them as part of a complete financial DSS.[8] Throughout the tutorial, computer commands to be executed by the user will be indicated by bullets, •, and system prompts will be indicated by double angle braces, >>.

B. THE PUBLIC WORKS ACCOUNTING SYSTEM

To begin the tutorial go to the drive and directory that contains the Lotus 1-2-3 Release 3.0 files and start 1-2-3. This illustration assumes that Lotus 1-2-3 Release 3.0 is located on drive C: in subdirectory \123R3 and that the DSS system files are located on drive A: in the root directory.

- C: ←┘ (make C: the current drive)
- CD\123R3 ←┘ (change to the 1-2-3 directory)

---

[8]System requirements: IBM PC compatible computer with an 80286 or 80386 microprocessor, hard disk, and a minimum of 1 megabyte of RAM. For use with large data sets, a minimum of 2 megabytes of RAM is recommended.

- 123 ↵ (start 1-2-3)

Set the default directory to the one containing the DSS system files and retrieve the accounting system file.

- /FD A:\ ↵
- /FR
- highlight the file ACCOUNT.WK3 and press [ENTER]

The accounting system Welcome Screen and main menu should now appear on the screen as shown in Figure 3. The main menu allows the user to select one of four accounting system modules or to exit the accounting system and save the current files by selecting Quit. The instructions at the bottom of the welcome screen explain the use of two important system keys. The [ESC] key is used to exit the accounting system to use 1-2-3 commands. When exiting the system in this way, the current files remain in memory as long as they are not removed using 1-2-3 commands. With the accounting system in memory, pressing ALT-A (holding the Alt key and pressing A) returns the user to the accounting system and restores the main menu. Use the arrow keys to highlight each of the main menu choices and notice the explanation which appears in the panel just below the menu options.

1. The Public Works Ledger Module

When the operating budget is received, the PWD must enter the amount of funds received in each subfunctional category in the ledger module. The ledger module is designed to let the user enter quarterly (QTR) operating targets

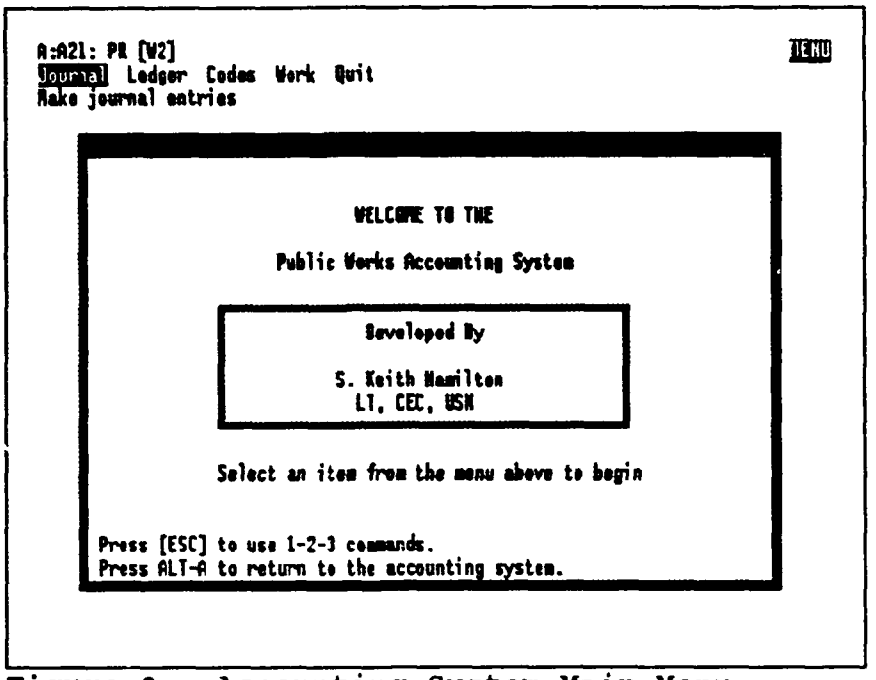


Figure 3. Accounting System Main Menu

(OPTARS) and annual planning figures (APF). After entering this data, total material obligations, contract obligations, and current account balances can be viewed in the ledgers. The procedure is as follows:

- highlight Ledger and press [ENTER] or simply press L

The ledger system input screen should now appear as shown in Figure 4.

The instructions at the top of the input screen explain the use of the four primary system keys which will be used in nearly every module. The [ENTER] key is used to enter data in a particular field and move the cursor to the next field. The [INS] key is used to transfer the completed data shown in the input screen to the database. The [END] key is used to terminate the current module and return to the system

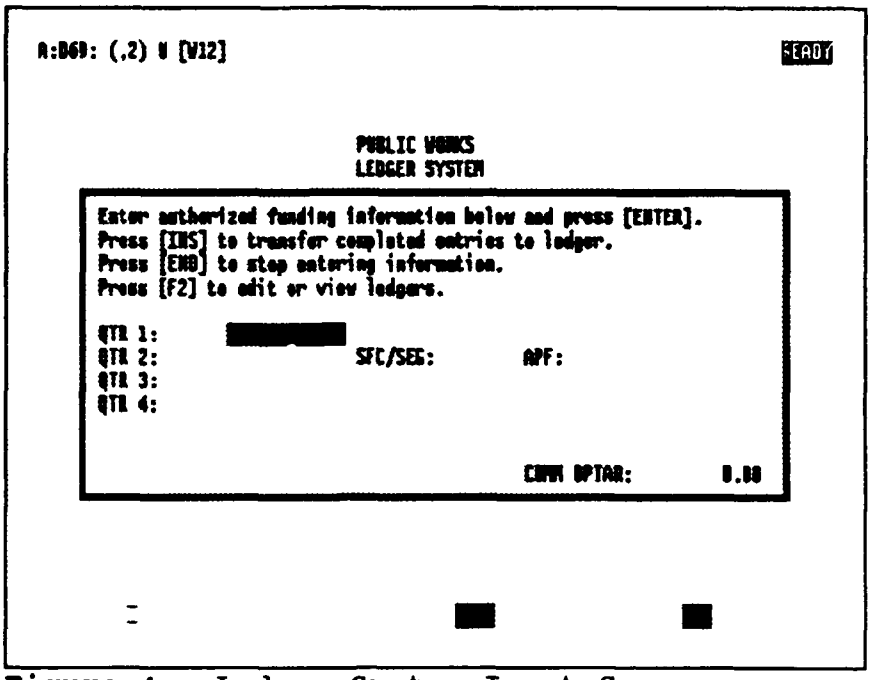


Figure 4. Ledger System Input Screen

main menu. The [F2] key brings up a submenu for editing existing records within a database. The use of each of these keys will be demonstrated in the following sections. The [ENTER] key and the [INS] key will be used to enter the data shown in TABLE 13.

TABLE 13. LEDGER ENTRY DATA

QTR 1	SFC	APF
100,000	LA	400,000
200,000	L7	600,000
300,000	M1	1,200,000
250,000	N1	1,000,000
50,000	P1	250,000
35,000	R1	175,000
10,000	S1	50,000

a. Entering Data

At the start of the fiscal year, an APF and the first quarter's Optar is entered for each subfunctional category based on information provided with the command's operating budget. The first record shown in Figure 13 will be used to demonstrate a system edit test. In the ledger system all four quarterly operating target fields and the APF field require numerical entries. In the first example, no entries will be made in the fields for QTR's 2,3, and 4. With the cursor in the field labeled QTR 1, enter the data for subfunctional category LA as follows.

- 100000 ◂┘
- → LA ◂┘ (LA is entered in the field labeled SFC)
- 400000 ◂┘
- press [INS] (to enter the completed record)

The computer will beep and display the following error message:

>> Error: this field requires a numerical entry. Press [ENTER] to continue.

The cursor has moved to the field labeled QTR 2, the first field with an input error. Zeros should be entered for any QTR fields for which quarterly OPTAR data is not available. The ledgers are then updated as the necessary information becomes available. Perform the following steps to correct the error and insert the completed entry into the database.

- press [ENTER] (to clear the error message)
- type 0 ◂┘
- repeat these steps for QTR 3 and 4

Enter the data for the remaining six subfunctional categories similarly, making sure to enter zeros for QTR's 2, 3, and 4.

b. Browse a ledger

To view one of the ledgers after entering beginning balances for each subfunctional category:

- press [F2]
- highlight Browse and press [ENTER] or simply press B

The system will prompt the user to specify the ledger to browse:

>> Enter SFC or press [ENTER] for reimbursables.

- type L7 ↵

The screen should appear as shown in Figure 5. The ledger cannot be edited in any way from the Browse command. To edit a ledger, the user must return to the ledger system and use the Modify or Delete command. Note the prompt in the panel above the ledger:

>> Press [ENTER] to return to input screen.

Return to the input screen and then end the ledger entry session as follows:

- press [ENTER]
- press [END]

2. The Codes Module

The PWD also receives funds from tenant commands and other activities to fund reimbursable work. At the beginning of the fiscal year the accounting division and the comptroller department establish segment codes to identify these funds.

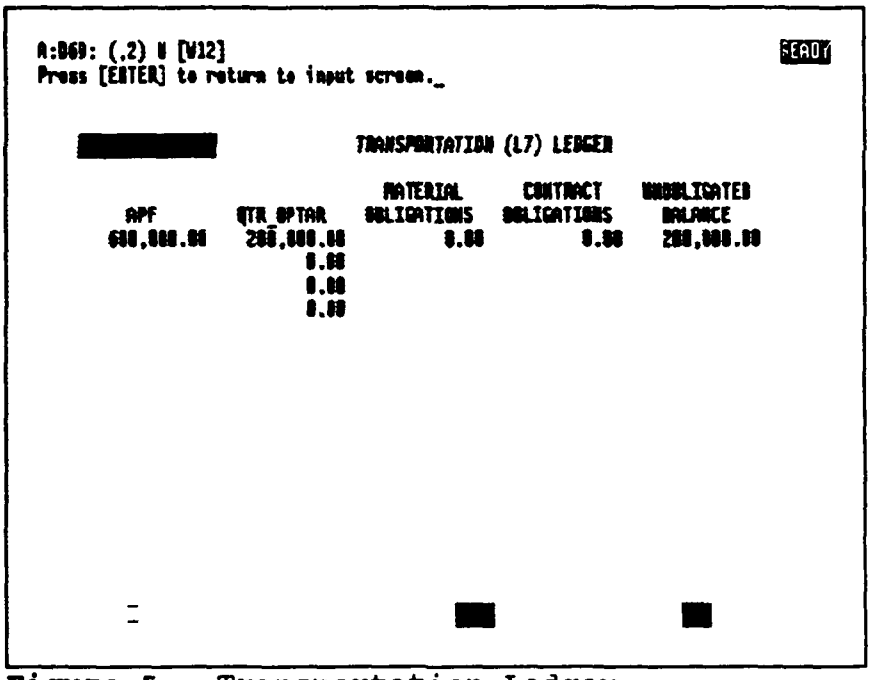


Figure 5. Transportation Ledger

These segment codes are entered into the accounting system to form an authorized list of segment codes against which all future entries will be tested. To enter the codes module from the main menu:

- highlight Codes and press [ENTER] or simply press C

The Codes input screen should appear as shown in Figure 6.

a. Entering Segment Codes

Note that the four primary system keys are used to perform the same functions as before. Enter the data shown in TABLE 14 by typing the segment code and pressing [ENTER]. After pressing [ENTER] the system will prompt the user to verify that the segment code has been keyed in correctly.

>> Verify correct SEG code. Accept? (y/n)

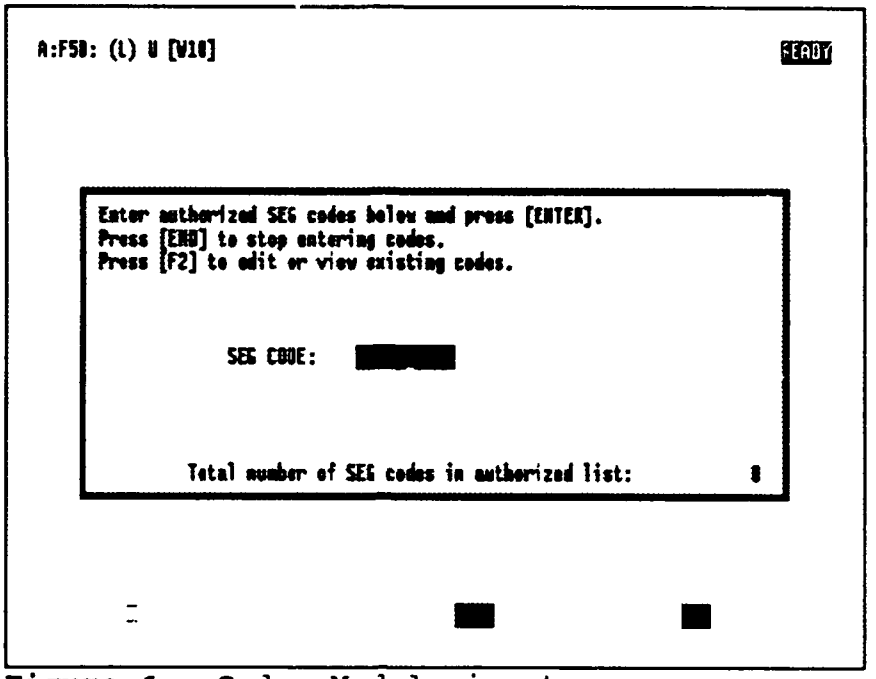


Figure 6. Codes Module input screen

- type Y ↵ (to accept) -- OR --
- type N ↵ (to reenter the correct SEG code)

It is critical that the codes are entered correctly since they will be used to validate all future SEG code entries. Therefore, user confirmation is required before accepting each entry.

TABLE 14. AUTHORIZED SEGMENT CODES

SEG	SEG
10AA	30CA
20AA	31DA
20AB	40AA
20BB	40CA
30AA	50CA
30AB	

The indicator at the bottom of the input screen keeps a running count of the number of segment codes in the authorized list. It should indicate a total of 11 after entering the data in TABLE 14.

b. Browsing the Authorized list of Segment Codes

To view the list of authorized SEG codes:

- press [F2]
- highlight Browse and press [ENTER] or simply press B

The authorized list of SEG codes is a single column on the screen. A prompt should appear in the panel above the list:

>> Press B ◂┘ to begin browsing. Press [ENTER] to return to input screen.

- press B ◂┘

The screen movement keys may now be used to view the entire list ([PgDn], [PgUp], ↑, and ↓) Normally this list would be several pages long. Return to the input screen:

- press [ENTER]

c. Modify and Delete Segment Codes

Suppose that after entering the segment codes it was discovered that one code had been entered incorrectly. Suppose the code entered as 31DA should actually be 31AD. To modify this entry:

- press [F2]
- highlight Modify and press [ENTER] or simply press M

>> SEG code to modify?

- type 31DA ◂┘

>> Change SEG to:

- type 31AD ◂—┘

>> Verify correct SEG code. Accept? (y/n)

- type Y ◂—┘ to accept
- type N ◂—┘ to cancel

Now use the [F2] key and browse the list and verify the changes. Return to the input screen and enter a meaningless segment code , such as TEMP, to demonstrate the use of the Delete command. Note that after entering TEMP the number of SEG codes indicator shows a total of 12 records in the system. To delete TEMP:

- press [F2]
- highlight Delete and press [ENTER] or simply press D

>> SEG code to delete?

- type TEMP ◂—┘

>> Are you sure you want to delete this SEG code? (y/n)

- type Y ◂—┘

Notice that the number of codes indicator now shows only 11 records in the system.

At the beginning of each fiscal year, there may be a need to erase all of the segment codes in the existing list to enter new ones. This is done with the Erase command. To illustrate the use of the Erase command (the authorized list will not actually be erased in this case):

- press [F2]
- highlight Erase and read the description
- press [ENTER]

>> Erase existing list of authorized SEG codes? (y/n)

- type N ◂—┘ to cancel (DO NOT ERASE THE LIST!)

- press [END] (to return to the main menu)

Note that pressing Y ↵ would actually erase the list of authorized segment codes from the file.

3. The Public Works Journal Entry Module

The journal entry system is used to record journal entries for transactions involving material purchases and contract awards. To start the journal entry system module, select Journal from the main menu:

- highlight Journal and press [ENTER] or simply press J

The journal module input screen should appear as shown in Figure 7. Notice the similar use of the primary system keys as indicated by the instructions at the top of the input screen. The middle portion of the input screen is used for data entry and the lower portion shows a running total of the number of journal entries in various journals. The journal system maintains separate journals for each of the subfunctional categories shown at the bottom of the screen. Additionally, the telephone (LA), Utilities (N1), and Reimbursables (REIMB) journals are further divided into separate monthly journals, making a total of 43 journals maintained by the system. The number of journal entries shown for LA, N1, and REIMB represent the total of the 12 monthly journals for each.

a. Making Journal Entries

The data shown in TABLE 15 represents the material costs of a job to repair the air conditioning in Building 23.

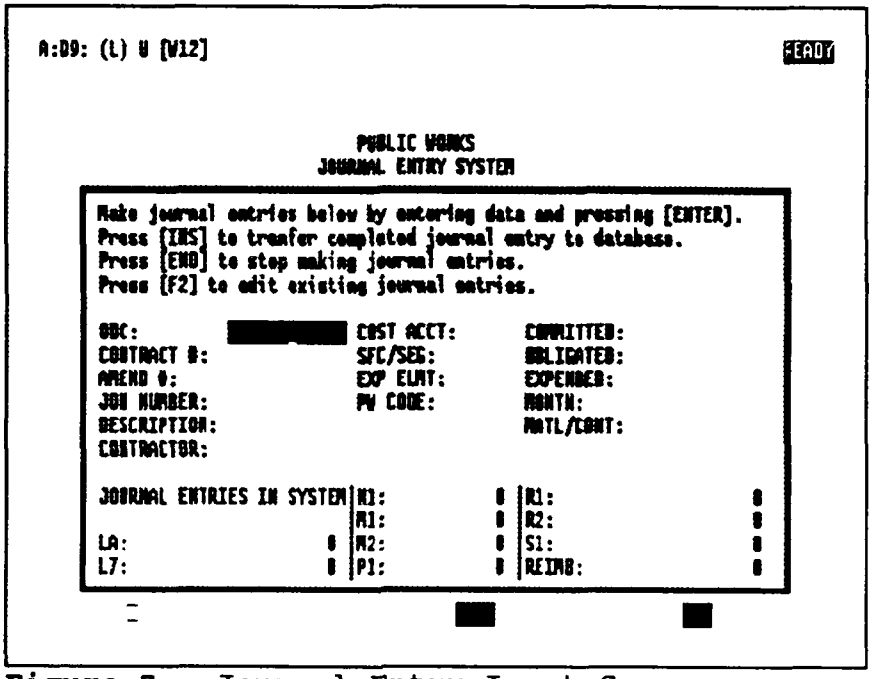


Figure 7. Journal Entry Input Screen

The total job cost was $1320, of which $750 was for material.[9] Enter the data exactly as shown in TABLE 15.

TABLE 15. JOURNAL ENTRY DATA

ODC:	89FA1756	**EXP ELMT:**	T
CONTRACT #:	NA	**PW CODE:**	08
AMEND #:	NA	**COMMITTED:**	750
JON NUMBER:	8970AA	**OBLIGATED:**	750
DESCRIPTION:	RPR A/C BLDG 23	**EXPENDED:**	0
CONTRACTOR:	ART'S A/C SUPPLY	**MONTH:**	JUN
COST ACCT:	7120	**MATL/CONT:**	M
SFC/SEG:	M1		

Notice that Not Applicable (NA) was entered for the contract # and ammend #. No data is required for these fields in this

[9]The labor costs will be entered in the Labor system separately. Labor accounting and material/contract accounting are normally performed by different people.

case; however, it is good practice not to leave blank fields in the database. Therefore, an entry should be made for each field. Add this entry to the maintenance journal after verifying that the data has been entered correctly:

- press [INS] (the M1 indicator now shows one record)

b. Browsing the Journals

Browse the maintenance journal to view this entry:

- press [F2]
- press B

>> Enter desired SFC, or press [ENTER] for reimbursables.

- type M1 ◄┘

>> Press B ◄┘ to begin browsing. Press [ENTER] to return to input screen.

- Press B ◄┘

The journal should look like the one shown in Figure 8.

The screen movement keys (↑,↓,→,←, [PgUp], [PgDn], [TAB], [HOME]) may now be used to view the entire journal. The titles and headers will remain at the top of the screen as you page down through a long journal.

- press [TAB] [TAB] (to view the entire entry)
- press [ENTER] (to return to the input screen)

Figure 9 shows the input screen after typing in journal entry data to record a commitment for a janitorial services contract for the Coast Guard on a reimbursable basis. Enter the data as shown in Figure 9 and add this entry to the reimbursables journal, after verifying that the data has been entered correctly:

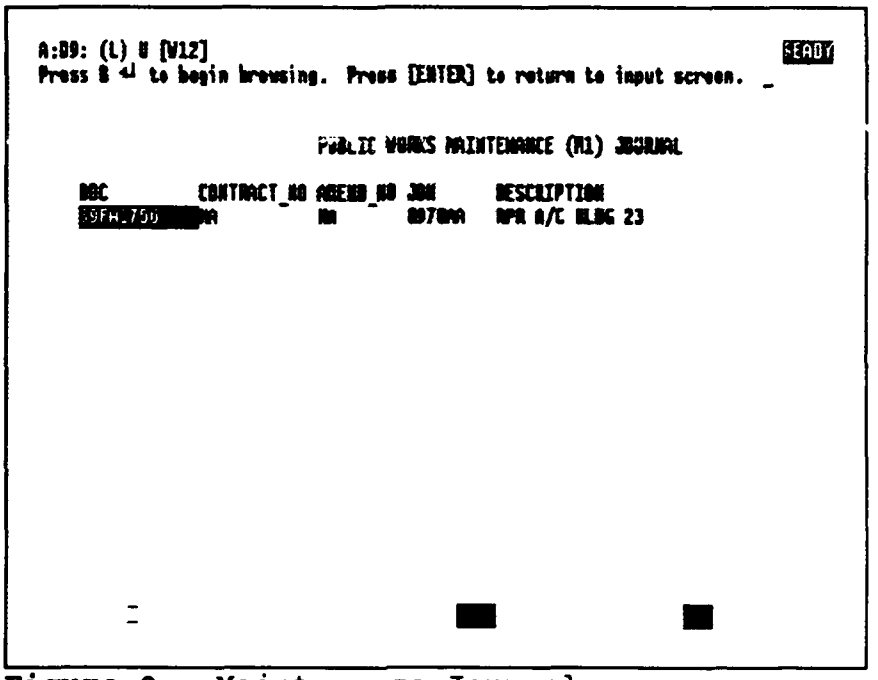


Figure 8. Maintenance Journal

- press [INS]

The computer should beep, move the cursor to the SFC/SEG field and display an error message.

>> Invalid SEG code. Try again? (y/n)

This error is a result of entering a SEG code that was not in the authorized list, 10AB. To enter the correct code:

- type Y ↵
- [F2] [BACKSPACE] A ↵ (to enter 10AA)

The system verifies your entries again and enters the record into the January Reimbursables journal if no errors are found. The REIMB indicator should now show one record.

- press [F2] B (to browse the journal)

>> Enter desired SFC, or press [ENTER] for reimbursables.

- press [ENTER] (to select the reimbursables journal)

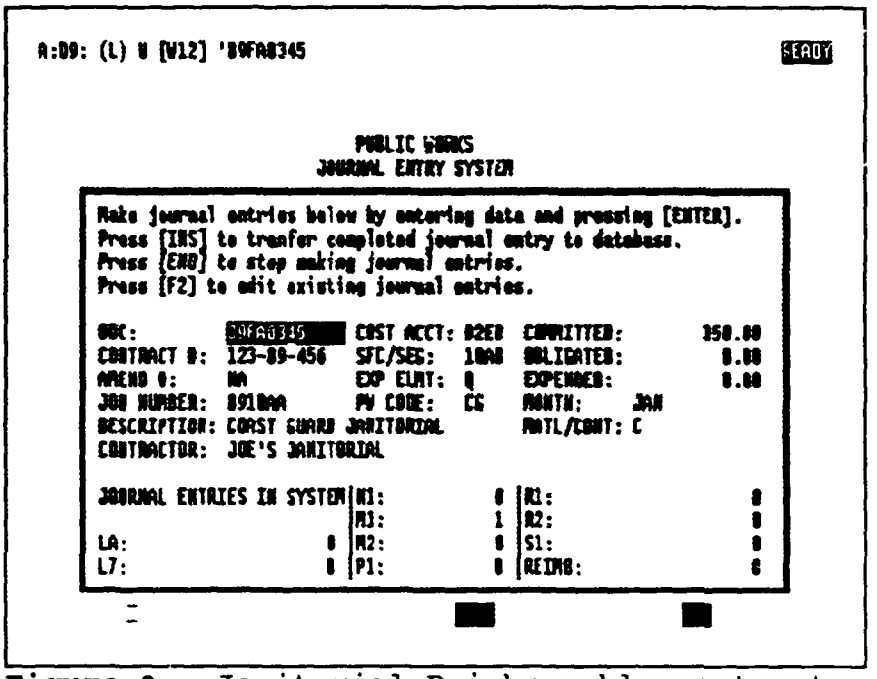


Figure 9. Janitorial Reimbursable contract

>> Enter month.

- type XXX ◄┘ (the computer will beep)

>> Error: month not correctly specified. Type correct month.

- type Jan ◄┘ (or January ◄┘)
- type B ◄┘ (browse the journal)
- press [ENTER] (to return to the input screen)

The prompt to enter the month will only appear when a monthly journal has been specified.

c. Modifying Journal Entries

Now suppose the janitorial contract that was entered has been signed and the funds obligated. The user must modify the journal entry to update the status of funds.

- press [F2]
- press M

>> Enter SFC/SEG.

- type 10AA ◂┘ (to specify the journal)

>> Enter month.

- type Jan ◂┘ (only prompted for monthly journals)

>> Enter ODC.

- type 89FAXXXX ◂┘ (to specify the journal entry)

>> Journal entry not found. Try again? (y/n)

- type Y ◂┘ (to enter the correct ODC)

The same series of prompts will be repeated.

- type 10AA ◂┘
- type Jan ◂┘
- type 89FA0345 ◂┘

All the data previously entered should be restored to the input screen as shown in Figure 10. Read the instructions in the panel above the input screen. Pressing the [END] key will cancel the modification, restore the entry to the journal unchanged, and return to the input screen. Pressing [INS] will verify the entries and replace the old journal entry with the modified one if no errors are found.

- press [ENTER] (to clear the prompt)

Use ↑,↓,→,←, [F2], or [ENTER] to make modifications to any field on the input screen. Use the arrow keys to move to the OBLIGATED field.

- type 350 ◂┘
- press [INS] (to make the modification to the journal)
- press [F2] B [ENTER] Jan ◂┘ (to browse the journal)
- press [TAB] [TAB] (note the change in the entry)
- press [ENTER] (to return to the input screen)

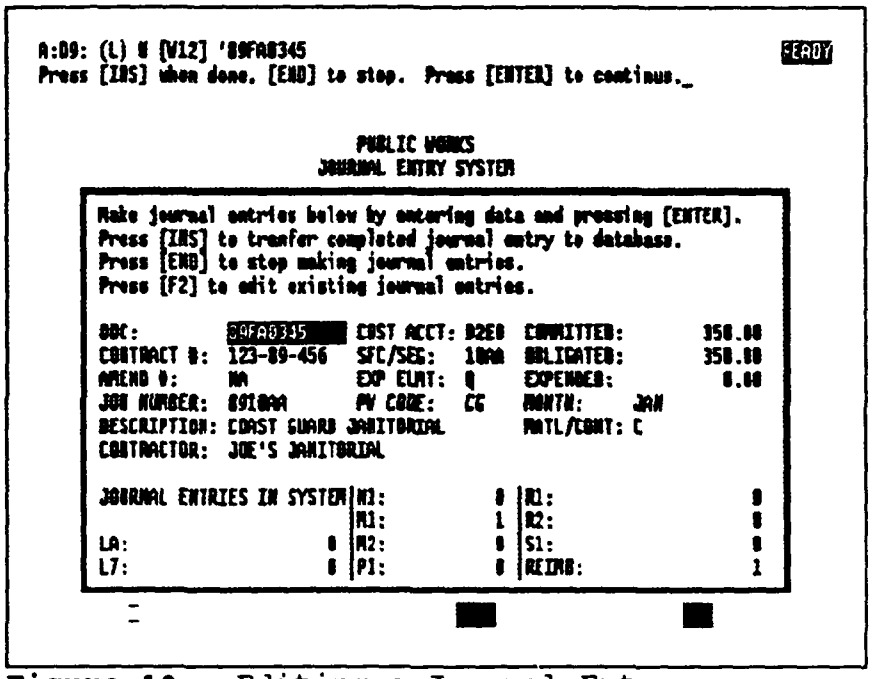


Figure 10. Editing a Journal Entry

d. Deleting Journal Entries

Before demonstrating how to delete journal entries, two meaningless records with the same ODC will be entered into the system. This procedure will illustrate the way the system warns the user of duplicate journal entries when modifying or deleting records. Enter a T in every field except: in SFC/SEG, enter M2, and enter any number in the three fields COMMITTED, OBLIGATED, and EXPENDED, since these three fields require numerical entries. After pressing [INS] the system will perform the edit tests indicated in Chapter III.

- press [INS] (computer will beep and display error msg)

>> Error: enter M for material or C for contracts. Press [ENTER] to continue.

- press [ENTER]

- type C ◂—┘

>> Did you enter special project number for JON? (y/n)

- type Y ◂—┘

When a special project is entered the system prompts the user to verify that the special project number was entered in the job number field. After typing Y ◂—┘, the system enters the record as specified. After typing N ◂—┘, the system prompts the user for the special project number and then enters this number in the job number field before inserting the record in the database. This is done to ensure that the number entered in the journal matches the number entered in the ledger and the planning system. The system performs the edit tests listed in Chapter III each time a new or modified record is inserted. Now that this record has been correctly entered, the M2 indicator should show one record. Now enter another record, identical to this one, so that two records with the same ODC are entered in the M2 journal.

- press [F2]
- press D (delete)

>> Enter SFC/SEG.

- type M2 ◂—┘

>> Enter ODC.

- type T ◂—┘ (computer beeps and displays warning)

>> Warning: multiple journal entries with the same ODC. Press [ENTER] to continue.

This warning alerts the user to the fact that there are duplicate journal entries before modifications or deletions

are made. It is important to correct this situation when discovered or it may result in erroneous balances in the ledger. The record may have been entered twice or only the ODC may be in error. Therefore, the user will normally want to look at the duplicate records before deciding whether to modify or delete them.

- press [ENTER]

>> Do you want to view extracted records? (y/n)

- type Y ↵
- type B ↵ (browse the duplicate records)
- press [ENTER] (to return to the input screen)

>> Are you sure you want to delete this journal entry?

The data shown on the input screen after extracting a record using Modify or Delete is always from the first record found if multiple entries are specified. Modifications performed using the Modify command will affect only the first record, the one on the screen; however, deletions will affect all records meeting the specified criteria.

- type Y ↵ (to delete records matching the criteria)
- press [F2] B
- M2 ↵ B ↵ (to browse the M2 journal)
- press [ENTER] (to return to input screen)

Notice that the M2 indicator showed zero records as soon as Y ↵ was typed to delete the specified records.

e. Erasing the Journals

At the end of each fiscal year the journals must be closed out and new journals must be opened before the

start of the new fiscal year. The Erase command is used to erase the journal entries from one or more journals to start the new fiscal year. Old files should be copied and stored for future reference before using this procedure.

- press [F2]
- highlight Erase (notice the submenu Specified All)
- press [ENTER] or E
- highlight both Specified and All (read description)

Choosing All from the Erase submenu will erase all journal entries from all journals. This is normally done at the end of the fiscal year after closing out old journals and saving the files for future use. Choosing Specified allows the user to erase one journal at a time according to user specifications.

- press S

>> Enter desired SFC, or press [ENTER] for reimbursables.

- type S1 ◂┘ (select journal to erase)

>> Are you sure you want to erase the journal entries? (y/n)

- type N ◂┘ (to cancel procedure)
- press [END] (to stop making journal entries)

>> Stop making journal entries? (y/n)

- type Y ◂┘

The procedures used for working in the journal entry system can now be applied to the special projects and reimbursables ledgers.

4. Additional Features of the Ledger Module

a. Entering Special Projects and Reimbursables

Special projects and reimbursable funds are provided by uniquely identifiable sources for specific purposes. Therefore, the ledgers for these funds include a database listing each specific special project number or segment code and the amount of funds associated with it. These funds are received in one lump sum and so only the APF field is used to record the amount of funds received. The procedures for using these ledgers are analogous to those for using the journal entry system. Select Ledger from the main menu to enter the ledger module.

- press L
- use arrows to move to SFC/SEG field
- type M2 ◄┘ (specify maint. special proj. ledger)
- 750000 ◄┘ (funds provided for specific project)
- press [INS]

>> Enter Special project number and press [ENTER].

- type R-89-001 ◄┘
- press [F2] B (select browse)

>> Enter SFC or press [ENTER] for reimbursables.

- M2 ◄┘ B ◄┘ (browse the M2 ledger)
- press [ENTER] (return to input screen)
- use arrows to move to SFC/SEG field
- type 10AA ◄┘ (to enter reimbursable funds code 10AA)
- 4200 ◄┘ [INS] (to insert the entry)
- use arrows to move to SFC/SEG field
- 20AA ◄┘ 5000 ◄┘ [INS]
- use arrows to move to SFC/SEG field
- 90AA ◄┘ 1000 ◄┘ [INS]

The system will beep and display an error message since SEG code 90AA is not in the authorized list. Every SEG code entry

will be tested against this list before it is entered in a database.

>> Invalid SEG code. Try Again? (y/n)

- type Y ←┘

>> Enter new SEG code.

- type 30AA ←┘ (a valid SEG code)
- press [F2] B [ENTER] (to browse reimbursables ledger)

The indicator panel should now indicate "Updating Ledger." The system is summing all the obligations currently recorded in the 12 monthly reimbursables journals. This updating procedure will always be invoked before the user views or copies any of the ledgers which involve monthly journals (telephone, LA, utilities, N1, and reimbursables). When the reimbursables ledger appears:

- press B ←┘ (to begin browsing)

The screen should look like Figure 11. Notice that a single entry representing the sum of all reimbursable material obligations and a single entry representing the sum of all reimbursable contracts is shown to the right of the funding data. The unobligated balance is calculated as the sum of all reimbursable funds provided, those entered in the APF column, minus material and contract obligations. This summary data is provided in the ledger since it is routinely needed for management purposes. Detailed job order cost records are normally available from other sources. However, detailed job order accounting reports can easily be generated in this

system using procedures similar to those for generating a variance report, as described in section D of this chapter.

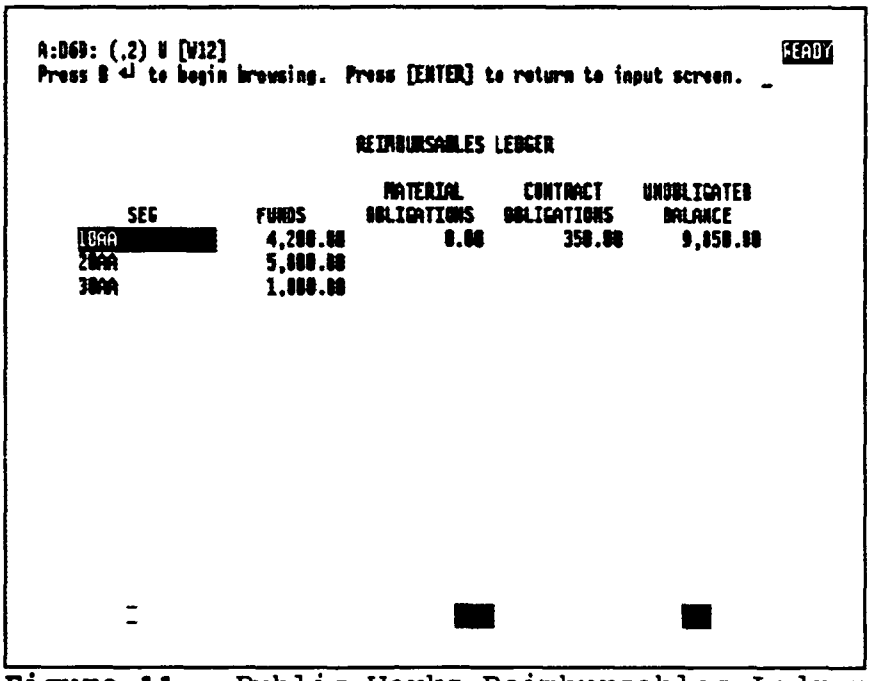


Figure 11. Public Works Reimbursables Ledger

b. Deleting a Ledger Entry

Slightly different procedures are used to delete ledger entries from the standard public works ledgers than from the special projects and reimbursables ledgers. This is necessitated by the fact that the special projects ledgers and reimbursables ledger normally contain numerous entries. Therefore, the user must specify the specific entry to be deleted in these ledgers. Both procedures will be illustrated, starting with a standard ledger and then proceeding to the reimbursables ledger.

- press [F2] D (select Delete)

>> Enter SFC or press [ENTER] for reimbursables.

- type S1 ◂┘

>> Are you sure you want to delete ledger entries?

- N ◂┘ (to cancel deletion)

Typing Y ◂┘ would result in deletion of all entries in the S1 ledger. The procedures for deleting entries in the special projects or reimbursable ledgers are illustrated as follows:

- press [F2] D

>> Enter SFC or press [ENTER] for reimbursables.

- press [ENTER] (delete a reimbursables entry)

>> Enter SEG and press [ENTER].

- type 30AA ◂┘

>> Delete SEG 30AA? (Y/N)

- TYPE Y ◂┘ (to delete the single entry for 30AA)
- press [F2] B (select browse)
- press [ENTER] (select reimbursables journal)
- B ◂┘ (to begin browsing)
- press [ENTER] (to return to input screen)

c. Modifying and Erasing Ledgers

Public works ledgers must be erased at the start of the new fiscal year to enter new data. Before erasing the ledgers you should save the current ledgers in a file for future use. The ledgers are erased by selecting the Erase command from the edit menu, the edit menu appears in the panel above the input screen after pressing [F2], as previously demonstrated for other modules. This procedure is used to erase all ledger entries from all ledgers in the system. As the fiscal year progresses, new information concerning

quarterly Optar approvals or revisions to the annual planning figure will become available. The ledgers are modified to reflect this new information as follows:

- press [F2] M (select Modify)

>> Enter SFC or press [ENTER] for reimbursables.

- type M1 ◄┘

The M1 ledger information should appear on the screen as shown in Figure 12.

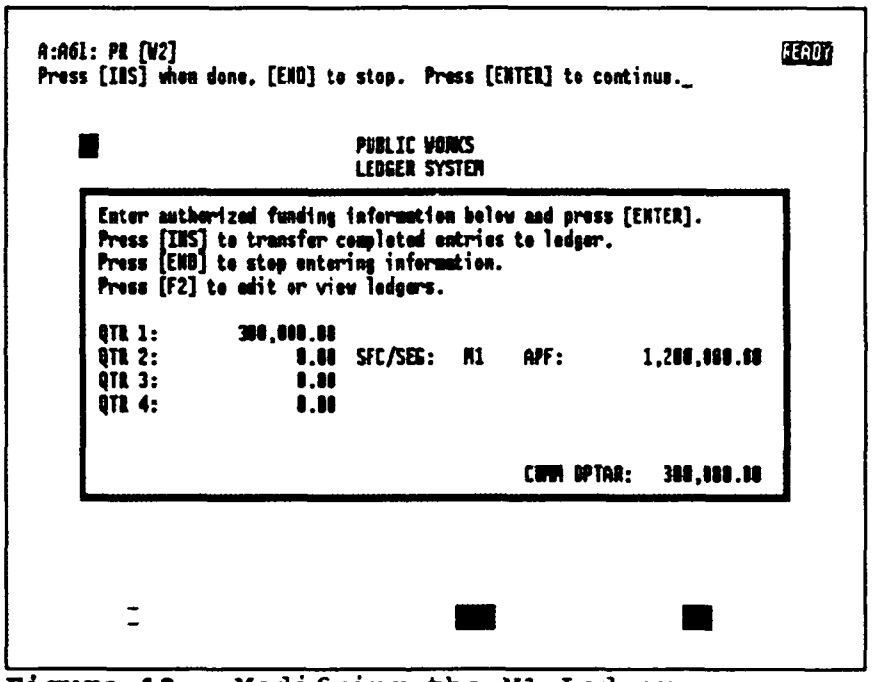


Figure 12. Modifying the M1 Ledger

Enter the following data in the quarterly Optar field indicated and watch the cumulative Optar indicator change as the data is entered.

- QTR 2: 350000 ◄┘
- QTR 3: 350000 ◄┘
- QTR 4: 200000 ◄┘

The cumulative Optar indicator should now agree with the APF, 1,200,000. If the APF does not agree with the cumulative Optar indicator, make the necessary corrections before continuing.

- press [INS] (to update the M1 ledger)
- press [F2] B (select browse)
- M1 ◄┘ (browse the M1 ledger)

Browse the M1 ledger to confirm the changes, as shown in Figure 13. Pressing [END], rather than [INS], after making modifications on the input screen, would have cancelled the modification and the ledger would have been left unchanged.

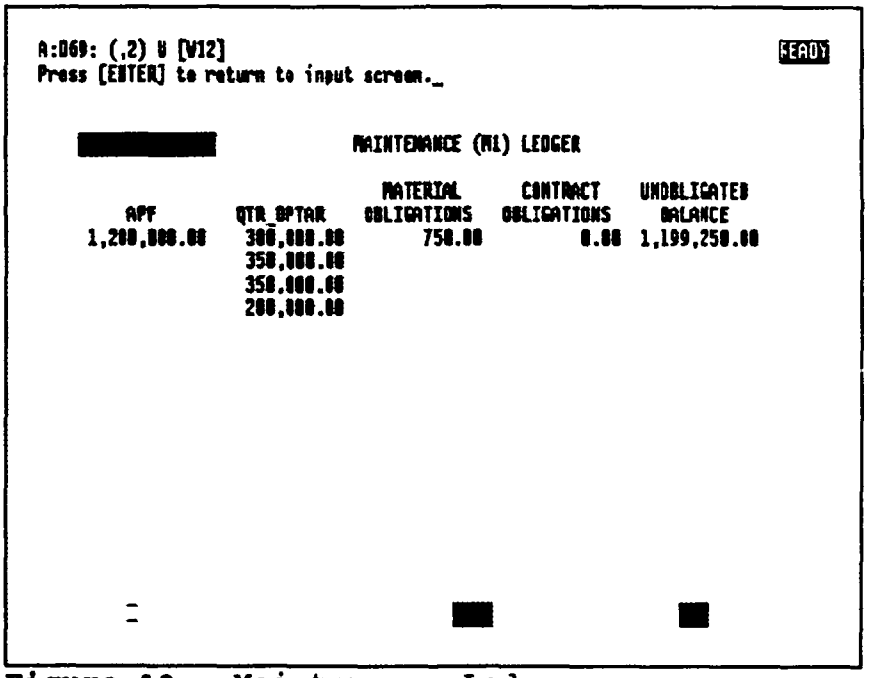


Figure 13. Maintenance Ledger

- press [END] (to return to main menu)

5. The Accounting System Scratch Pad

The scratch pad provides an area for producing reports, analyzing data, constructing graphs, or any other user defined operations. The user may copy any of the journals or ledgers to the scratch pad for further use or analysis simply by making selections from a menu. Once the selected data is copied to the scratch pad, the user is free to use 1-2-3 commands to manipulate the data, create new files for future use, create and print customized reports, create and use customized macros for automating routine operations, or anything else supported by 1-2-3 commands. To use the scratch pad, select Work from the main menu.

- highlight Work (read the submenu)
- press [ENTER] or W
- highlight each submenu command and read description
- press S (exit system and go to scratch pad)

>> Press ALT-A to return to the accounting system. Press [ENTER] to continue.

This prompt reminds you that the user to press ALT-A when ready to return to the accounting system from the scratch pad.

- press [ENTER] (to go to the scratch pad)

Note that the 1-2-3 frame is restored and the current date is displayed in the upper left hand corner of the scratch pad. Any of the 1-2-3 commands may now be used to perform user defined work in the scratch pad area or even to open other files or make new files.

- press ALT-A (return to main menu)

a. Journal Reports

Printed copies of transaction listings are frequently used in the PWD. The procedures to create a report using one or more of the public works journals are as follows:

- press W (select Work)
- O (select Other)

The user can select the journal or set of ledgers to copy to the scratch pad from the "Other" submenu. Once copied to the scratch pad, other journals or ledgers can be added to the scratch pad, or 1-2-3 commands can be used to manipulate the data as desired.

- J (select a journal to copy to the scratch pad)

>> Enter SFC or press [ENTER].

- press [ENTER] (to select the reimbursables journal)

>> Enter month.

- type Jan ←┘

>> Press ALT-A to return to the accounting system. Press [ENTER] to continue.

- press [ENTER] (to go to the scratch pad)

Any of the 1-2-3 commands may now be used to add or delete columns, change headings and titles, or perform statistical analysis. To combine this data with other data, simply use the /Move command to move the January reimbursables journal at least three screens to the right (if it is moved further down the spreadsheet in the same columns, other data may be copied over it) and return to the accounting system to extract another journal or ledger. To save work from the scratch pad

to a new file the following procedures can be used (these procedures are presented for information only and need not be executed at this time):

- /F N (to open a new file in memory)
- /C (to copy the work to the new file)
- use pointing commands to define the copy ranges
- /F S (to save the new file to disk for future use)
- /W D F (to delete the new file from current memory)

These commands may all be performed without ever leaving the scratch pad in the accounting system file.

- press ALT-A (to return to the main menu)

b. Public Works Ledger Reports

The status of funds report is an extremely useful tool for public works managers. These reports may be easily generated using the reports module since the system updates the ledgers automatically. The public works ledgers may be used to create a status of funds report as follows:

- W (select Work)
- E (Erase the old work on the scratch pad)

>> Are you sure you want to erase the scratch pad? (y/n)

- type Y ←┘
- W O (select Work, Other)
- L (use public works ledgers to create a report)

This copies the public works ledgers, from the command's operating budget, to the scratch pad for use in generating a status of funds report. Before the telephone, LA, and utilities, N1, ledgers are copied to the scratch pad, they will be updated. The indicator panel will show "Updating Ledger" while the updating procedure is in progress. This

updating procedure is not necessary for ledgers that do not use data from monthly journals since they are automatically updated each time a new journal entry is made. The ledger report should now appear on the screen as shown in Figure 14. Use the [PgDn] key to view the entire report. This report could now be printed or customized using 1-2-3 commands.

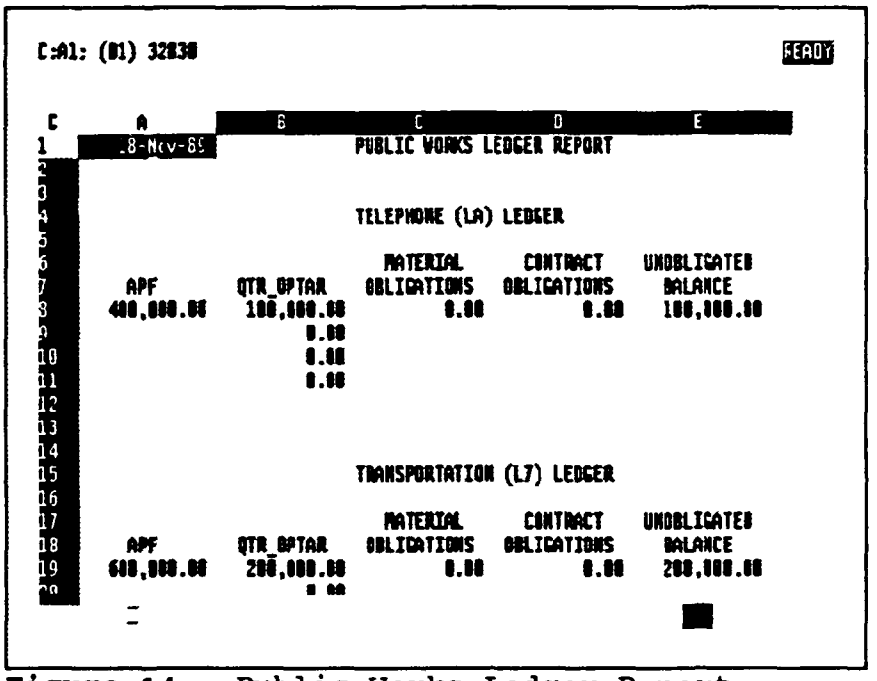


Figure 14. Public Works Ledger Report

c. Special Projects and Reimbursables Ledgers

The special projects and reimbursables ledgers are separated from the other ledgers because they are not part of the station's operating budget and because they may be very lengthy by themselves. The procedures for copying these ledgers to the scratch pad are analogous to those already described.

- press ALT-A (to return to the main menu)
- W E (Erase the scratch pad)

>> Are you sure you want to erase the scratch pad? (y/n)

- type Y ↵
- W O (select Work, Other)
- S (select special projects ledgers for report)

Note that only the M2 ledger was copied to the scratch pad. The R2 ledger was not extracted since there are no entries in this ledger. Only those ledgers which have entries will be extracted. The system will not extract an empty ledger.

- press ALT-A (to return to the main menu)
- W E Y ↵ (to Erase the scratch pad)
- Q (to end the accounting system session)

>> Do you want to save your work? (y/n)

- Y ↵

Choose Y to save the journal entries and ledger entries that have been made during any session. The entries that have just been made will be used again later in this chapter. Selecting N ↵, terminates the session, exits the accounting system and returns to 1-2-3. The accounting system file will still be in memory and can be saved using 1-2-3 commands.

C. THE PUBLIC WORKS LABOR SYSTEM

The public works labor system is used for labor accounting and employee records. The procedures used in this system are very similar to those used in the accounting system. Differences in the two systems will be illustrated during demonstration of the labor system. With the default directory set to A:\ in Lotus 1-2-3: (if the accounting system file is

still in memory, use /Worksheet, Erase to remove it from memory before proceeding)

- /FR
- highlight LABOR.WK3 and press [ENTER]

You should now see the labor system welcome screen and main menu as shown in Figure 15. Any one of the four modules in the labor system can be started simply by selecting it from the main menu. The two system keys, [ESC] and ALT-A, function to exit and return to the system just as they did in the accounting system. Highlight each of the modules on the main menu and read the description in the panel below.

1. The Public Works Personnel Module

The personnel module is used to maintain employee records. The wage rates recorded in this module will be used to calculate labor costs for the labor cost report. Additionally, the social security numbers stored in the employee records are used to validate social security number entries in the labor module.

a. Entering and Browsing Employee Records

Start the personnel module by selecting Personnel from the main menu.

- P

The system keys, explained at the top of the input screen, function essentially the same as they did in the accounting system. Social security numbers may be entered in one of two formats: 1) XXX-XX-XXXX, or 2) XXXXXXXXX. Eleven characters are allowed if format one is used and nine

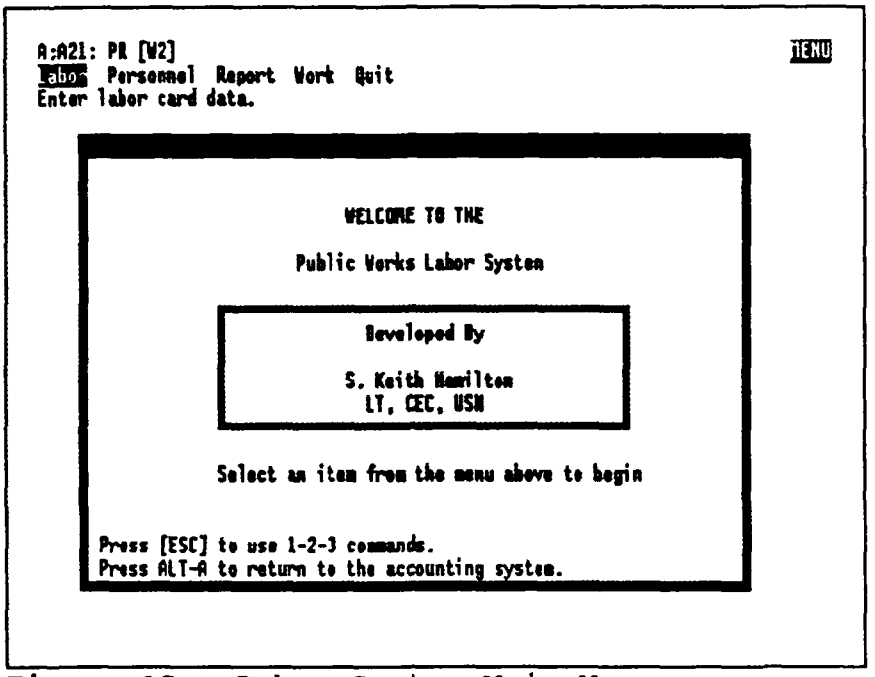


Figure 15. Labor System Main Menu

characters are allowed if format two is used. Enter the data shown in Figures 16, 17, and 18. Use the [ENTER] key after typing the data in each field and then use the [INS] key to transfer the completed employee record to the database. The indicator at the bottom of the input screen keeps a running total of the number of employee records entered in the system. It should show three records in the system after entering the data above. To browse the personnel database:

- [F2] B (select Browse)
- B ◄┘ (begin browsing)
- Use [TAB], or other screen movement keys to view records
- [ENTER] (return to input screen)

b. Modify, Delete, and Erase

Employee records may be specified by the employee's last name or social security number. After

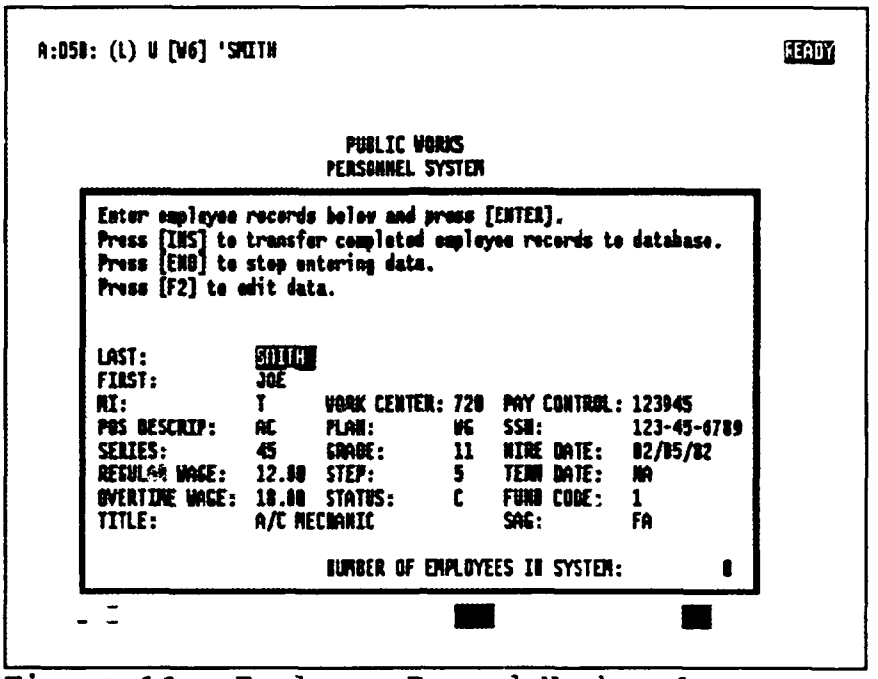


Figure 16. Employee Record Number 1

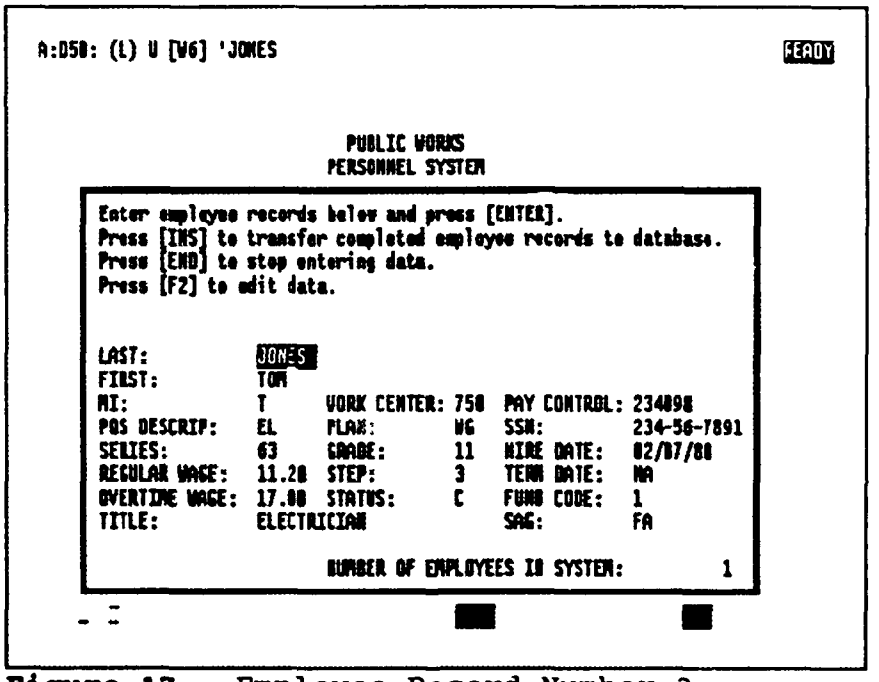


Figure 17. Employee Record Number 2

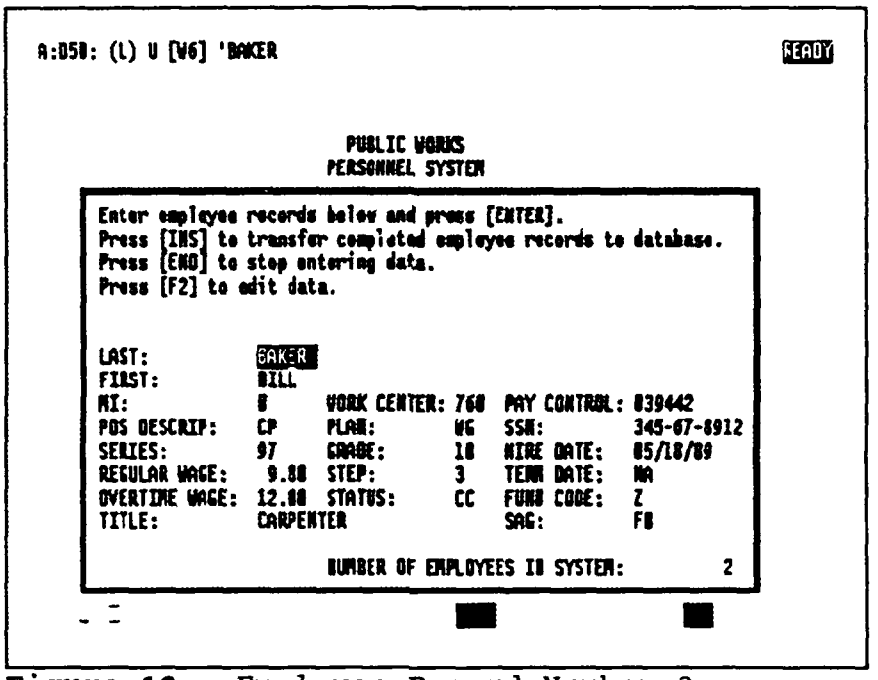


Figure 18. Employee Record Number 3

selecting Modify or Delete from the edit, [F2], menu, the user will be prompted for both last name and social security number. Only enough information to uniquely identify an employee record need be specified. If the information requested by one of the prompts is not necessary, simply press [ENTER] to continue with the procedure. For example, to modify Baker's record:

- press [F2] M (select Modify)

>> Enter employee's last name or press [ENTER] to continue.

- type BAKER ←┘

>> Enter social security number or press [ENTER] to continue.

- press [ENTER] (there is only one Baker in the system)

The input screen should now contain the information entered for Baker and the following prompt should appear in the panel above the input screen.

>> Press [INS] when done, [END] to stop. Press [ENTER] to continue.

- press [ENTER] (to clear the prompt)

Any desired modifications to Baker's record could now be made on the input screen. Pressing [INS] would then replace the old record with the new modified version. In this case no modifications are necessary. To abort the modification procedure:

- press [END] (to abort modification)

>> Cancel modification and return to entry screen? (y/n)

- type Y ◂┘

The Delete command is used in a similar manner. After selecting Delete from the Edit menu, [F2] brings up the menu, the user will be prompted to specify an employee record. The data from the record specified will appear on the input screen and the user will be prompted to confirm the deletion of this record:

>> Are you sure you want to delete this employee record? (y/n)

Type Y ◂┘ to delete the record or N ◂┘ to cancel the deletion and return to the input screen.

The Erase command is used to erase all existing employee records from the personnel system. This command is

selected from the Edit menu. After selecting the Erase command the user will be prompted for confirmation:

>> Are you sure you want to erase all existing personnel data? (y/n)

Type Y ↵ to erase all employee records or N ↵ to cancel the procedure and return to the input screen. If the edit menu is brought up accidentally, simply press [ESC] to remove it. Return to the main menu:

- press [END]

2. The Public Works Labor Module

Labor card data is entered in the labor module for use in labor accounting. The procedures used in the labor module are analogous to those used in the personnel module. To start the labor module from the main menu:

- L

The indicator at the bottom of the input screen keeps a running count of the number of labor card records in the system. Social security numbers entered in this module will be validated against those in the employee records. Therefore, an employee record must be entered in the personnel system before trying to record that employee's labor card data in this module.

a. Entering Labor Card Data

Enter the data shown on the input screen in Figure 19 using the [ENTER] key and [INS] key as before. After pressing [INS], the computer will beep, move the cursor to the REGULAR HRS field, and display an error message:

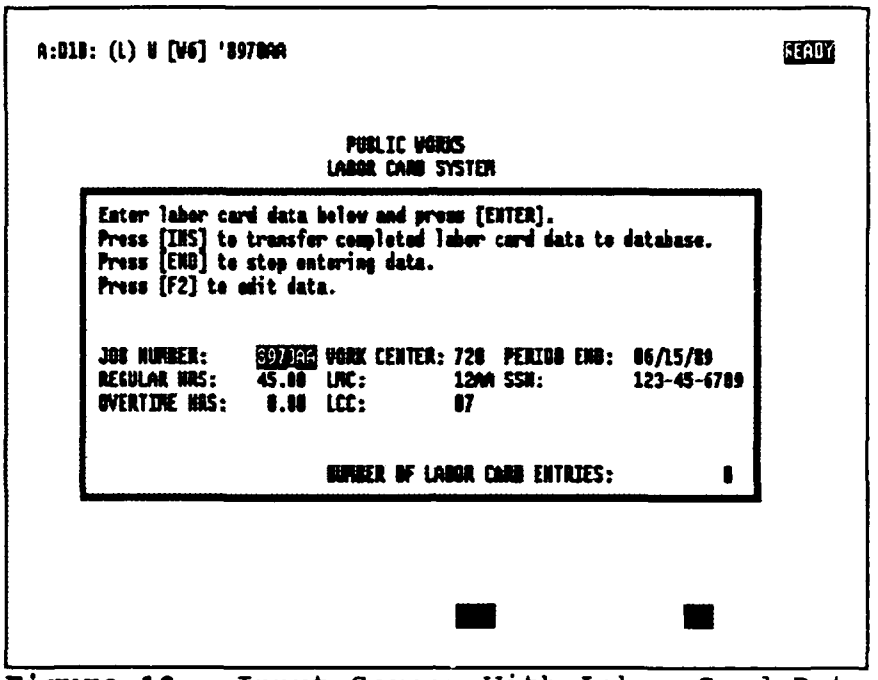


Figure 19. Input Screen With Labor Card Data

>> Error: regular hours cannot exceed 40. Press [ENTER] to continue.

- press [ENTER]
- type 40 ↓
- 5 ◄┘

Since the wages for regular time and overtime are different, the system checks the entry in the REGULAR HRS field to ensure that no more than the maximum number of hours properly charged at the regular time wage, 40, is entered. After correcting this error, the system enters the record and the number of labor cards indicator displays one.

Notice that this job represents the labor portion of the job to repair the air conditioning in Building 23. A journal entry for the material costs associated with this job

was entered in the accounting system earlier. It is critical that the job number for both the material and labor portions of any job be the same, so that the two separate files, labor costs and material costs, can be joined to create a variance report. The procedure for producing variance reports will be illustrated in Chapter V. Browse the labor database to view this record.

- [F2] B (select Browse)
- B ◄┘ (begin browsing)

The screen should appear as shown in Figure 20.

- press [ENTER] (return to input screen)
- Enter the additional labor card data shown in Figure 21.

Notice that the first record shown in Figure 21 has already been entered into the system. Do not duplicate this record.

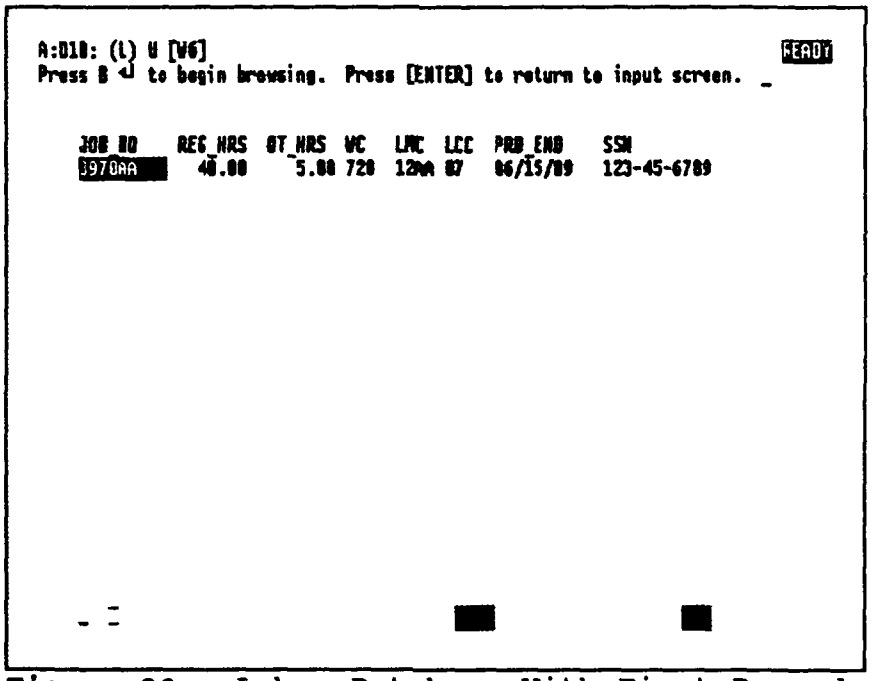


Figure 20. Labor Database With First Record

b. Modifying, Deleting, and Erasing

Labor card records may be specified by the employee's social security number, the job number, and the pay period (period ending date of the labor card). When prompted for this information, simply press [ENTER] if the requested information is not necessary to uniquely specify a labor card record. Normally, only the social security number and job number will be required.

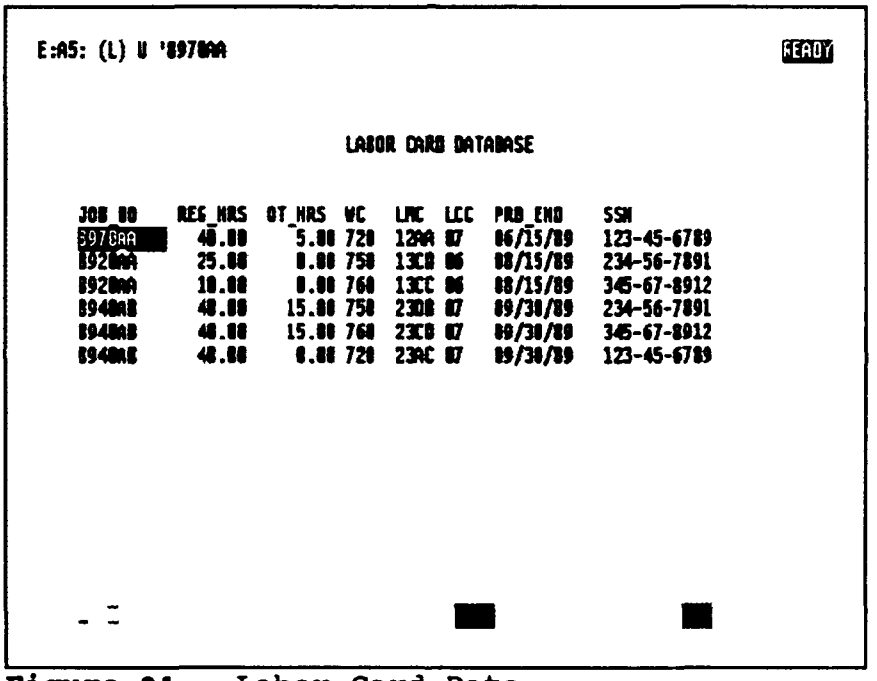


Figure 21. Labor Card Data

To modify a labor card record:

- [F2] M (select Modify)

>> Enter social security number or press [ENTER] to continue.

- type 123-45-6789 ◄┘

>> Enter job number or press [ENTER] to continue.

- type 8940AB ◄┘

>> Enter pay period or press [ENTER] to continue.

- press [ENTER]

The input screen should now appear as shown in Figure 22.

Change the recorded number of overtime hours to two.

- move the cursor to the OVERTIME HRS field
- type 2 ◄┘
- press [INS]

Browse the database to confirm the modification if desired.

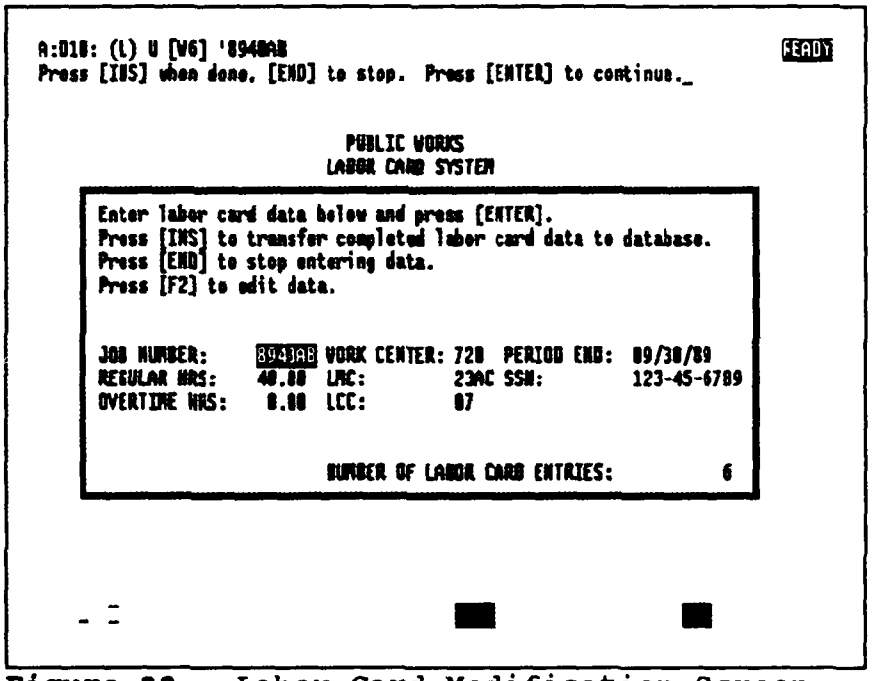


Figure 22. Labor Card Modification Screen

The Delete command, selected from the Edit menu, is used in a similar manner as just described. After specifying the record to delete, the data from that record will appear in the input screen and the user will be prompted to confirm the deletion:

>> Are you sure you want to delete this labor card data? (y/n)

Type Y ◂┘ to delete the specified record or N ◂┘ to cancel the deletion and return to the input screen.

The Erase command is used to erase all the labor card records in the system. This will normally be done at the beginning of each new fiscal year. A copy of the previous year's data should be made on a separate file before using this procedure. After selecting the Erase command from the Edit menu, the user will be prompted for confirmation.

>> Are you sure you want to erase all existing labor card data? (y/n)

Type Y ◂┘ to erase all the labor card records or N ◂┘ to cancel the procedure and return to the input screen. To end the labor card entry session:

- press [END]

>> Stop entering labor card data? (y/n)

- type Y ◂┘ (return to the main menu)

3. The Public Works Labor Cost Report Module

The report module is used to create a labor cost report for all jobs entered in the labor system to date. Select Report from the main menu to create a labor cost report.

- R

The labor card database is joined with the personnel database by the social security number. For each labor card record, the system finds the related employee record, and multiplies

the regular hours by the regular wage and the overtime hours by the overtime wage to calculate the cost of regular and overtime hours for each labor card record. The data is then aggregated by job number to create a labor cost report which shows the cost of regular time, overtime, and the total cost for each job. The cumulative total labor costs are also shown on the report. While the report is being generated, the indicator panel will indicate "CREATING LABOR REPORT." The labor cost report shown in Figure 23 should now appear on the screen.

- B ◂—┘ (to begin browsing)

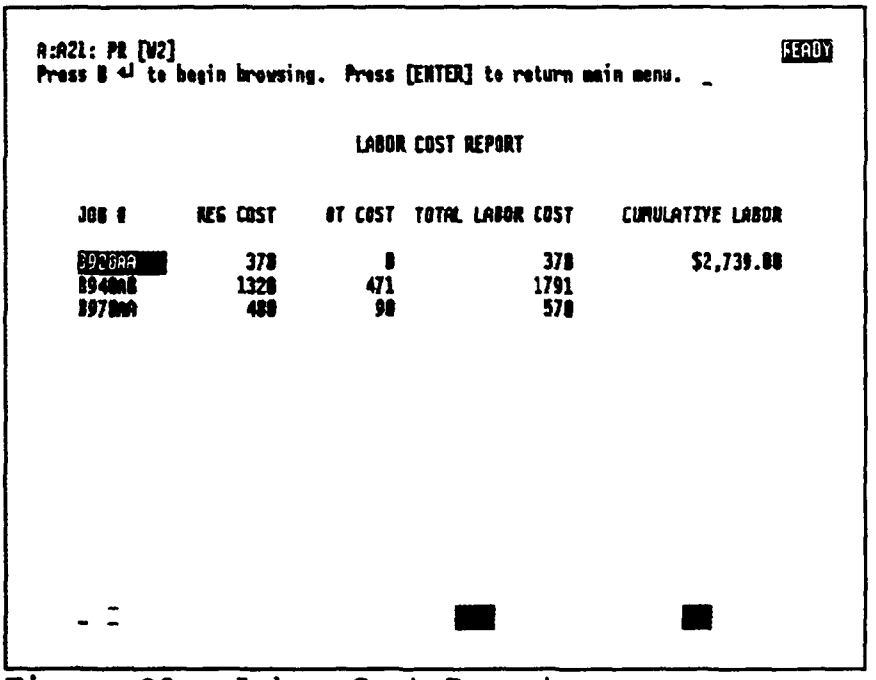


Figure 23. Labor Cost Report

Notice the report shows the labor costs for the three jobs entered in the labor system. The calculations for creating the labor cost report can be verified as shown in TABLE 16.

JOB #	REG COST	OT COST	TOT COST	EMPLOYEE
8920AA	25 x 11.20	0	280.00	234-56-7891
	10 x 9.80	0	98.00	345-67-8912
SUB TOT	378.00	0	378.00	
8940AB	40 x 12.00	2 x 18.00	516.00	123-45-6789
	40 x 11.20	15 x 17.00	703.00	234-56-7891
	40 x 9.80	15 x 12.00	572.00	345-67-8912
SUB TOT	1320.00	471.00	1791.00	
8970AA	40 x 12.00	5 x 18.00	570.00	123-45-6789
SUB TOT	480.00	90.00	570.00	
CUMULATIVE LABOR COST:			2739.00	

TABLE 16 LABOR COST REPORT CALCULATIONS

- press [ENTER] (to return to the main menu)

4. The Labor System Scratch Pad

The functions and purpose of the labor system scratch pad are analogous to those of the accounting system. Highlight Work in the main menu and notice its submenu. The Scratch command is used to exit the labor system and go to the scratch pad. The Erase command is used to erase the contents of the scratch pad. Any or all of the databases or reports in the labor system may be copied to the scratch pad for use in creating a report or performing other analysis. Select Work, highlight Other and read the descriptions of the "Other" submenu.

- W (select Work)
- O (select Other)
- A (copy all databases and reports to scratch pad)

It is important to note that, these procedures copy the current version of the labor cost report to the scratch pad without updating. If the labor cost report is not current, then create a new labor cost report, by selecting Report from the main menu, before copying it to the scratch pad. The screen should appear as shown in Figure 24. The three databases are placed side by side in the scratch pad since each may contain many records. Use the [TAB] key to view the entire contents of the scratch pad. Any of the 1-2-3 commands may now be used to create a customized report, print a report, perform data analysis, and save work to a new file for future use.

- press ALT-A (to return to main menu)
- W E (to Erase the scratch pad)

>> Are you sure you want to erase the scratch pad? (y/n)

- Y ↵
- Q (quit and save the labor system file)

>> Do you want to save your work? (y/n)

- Y ↵ (to save the file)
- /WE (to erase the file from current memory)

D. THE PUBLIC WORKS JOB PLANNING SYSTEM

The job planning system is used to enter data about planned jobs so that this data can be used to support decision making. The primary uses of this data include supporting resource allocation decisions; developing resource allocation

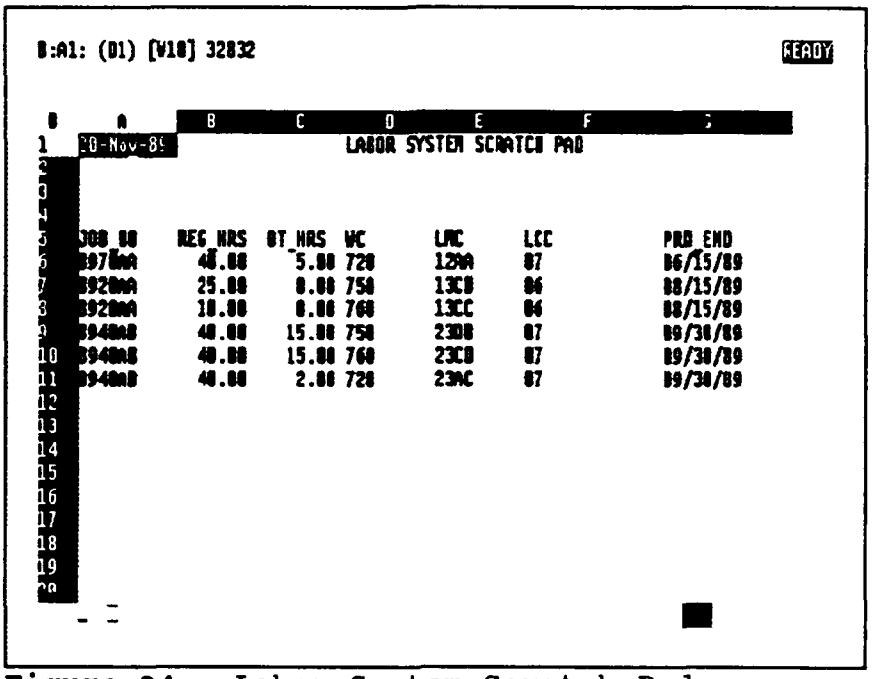


Figure 24. Labor System Scratch Pad

plans and projected spending rates for comparison with actual resource allocations; projecting end of the year status of funds by adding the planned cost of jobs selected for accomplishment to the actual cost of jobs already completed; and, creating variance reports for use in PWD management control. The functions and procedures in the planning system are completely analogous to those of the accounting and labor systems; and therefore, they will be demonstrated only briefly here.

1. The Public Works Job Planning Module

Retrieve the job planning system file to begin the planning session:

- /FR

- highlight PLAN.WK3 and press [ENTER]

The job planning system welcome screen and main menu should appear on the screen as shown in Figure 25.

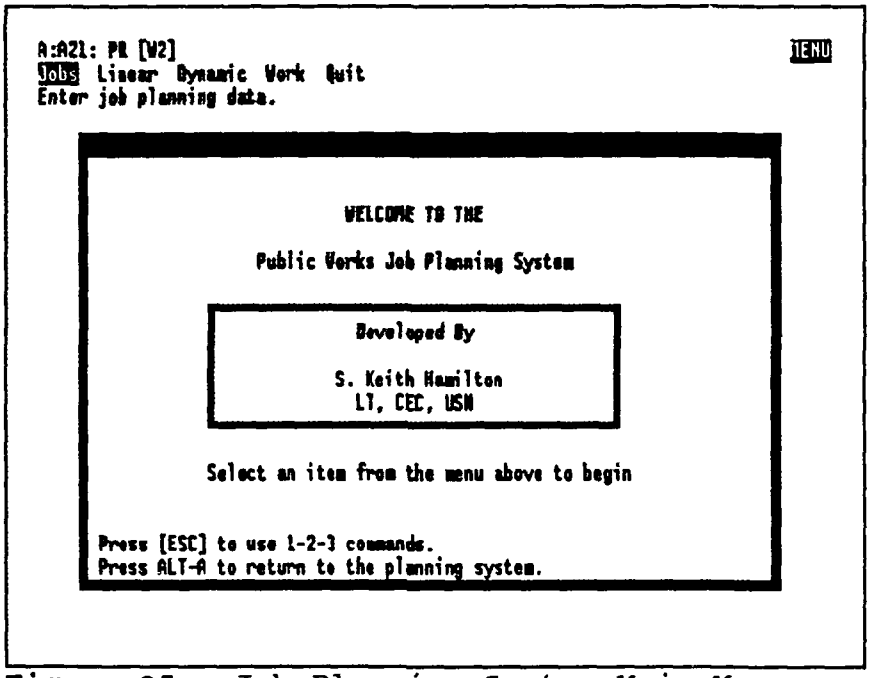


Figure 25. Job Planning System Main Menu

- J (select Jobs module from main menu)

Notice the control totals at the bottom of each column of input data on the job planning input screen. These totals will change as data is entered so that the effects of various actions can be verified as they are executed. Enter the data shown on the input screen in Figure 26.

Enter the additional job planning data shown in Figure 27. Use the [F2] Browse command to view the job planning database and confirm the entries.

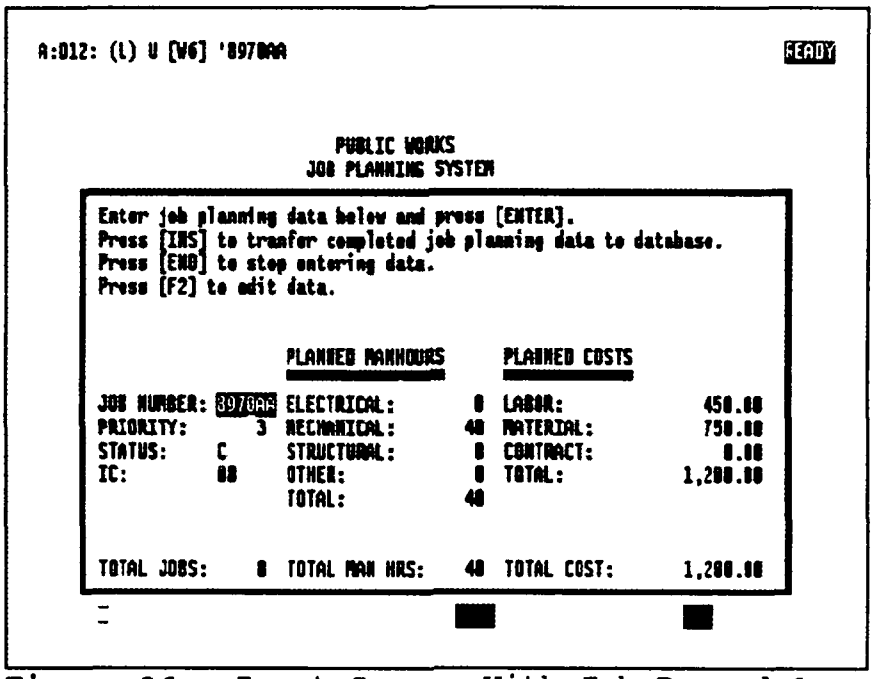


Figure 26. Input Screen With Job Record 1

The Modify and Delete commands, selected from the Edit menu, function in the same manner as they did in the accounting and labor systems. After selecting either Modify or Delete, the user will be prompted to specify the job number of the record to edit. Each job planning record can be uniquely specified by its job number. If the job number specified is not found in the job planning database, the computer will beep, display an error message, and allow the user to try again. The Erase command, selected from the Edit menu, is used to erase all existing records from the job planning database. These functions are completely analogous to those previously demonstrated.

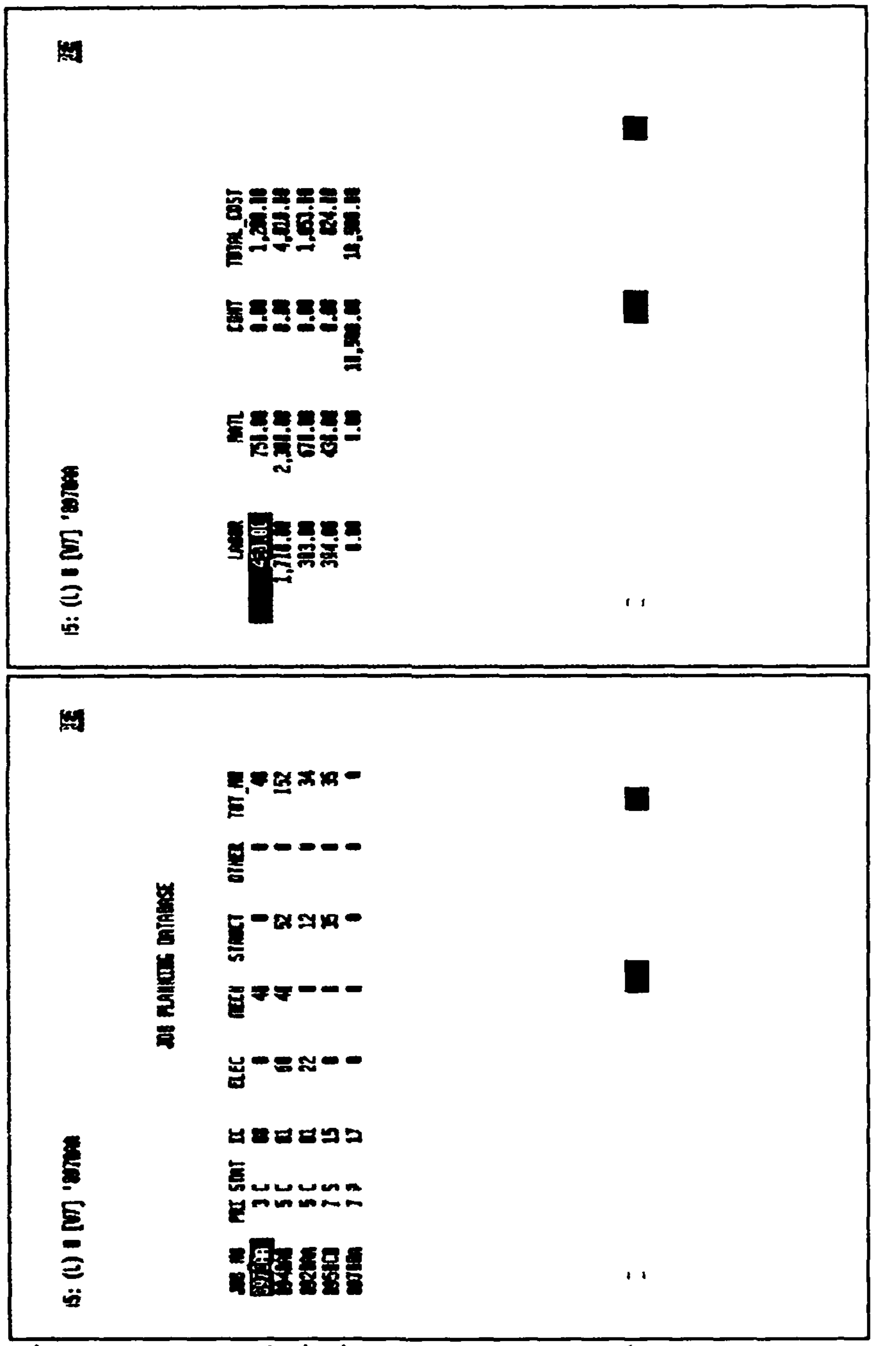


Figure 27. Additional Job Planning Data

2. The Linear and Dynamic Programming Modules

Highlight Linear and then Dynamic on the main menu and read the description of each in the panel below the menu. Note that both of these modules say ** NOT INSTALLED **. After selecting either Linear or Dynamic the following prompt will appear on the screen:

>> This module is not installed. Press [ENTER] to return to main menu.

These menu items are included here only to emphasize their potential for use in this system and to highlight the need for future study in this area. This will be discussed in more detail in Chapter VI.

3. The Job Planning Scratch Pad

Highlight the Work command in the main menu to view the, now familiar, Work submenu. These commands function in the same manner as previously demonstrated. Extract a copy of the job planning database for use in the scratch pad:

- W
- O

Notice that columns L and M must be widened in order to see the numbers entered in these fields. After viewing the extracted database return to the main menu, erase the scratch pad, and save the job planning file:

- ALT-A
- W
- E
- Q

>> Do you want to save your work? (y/n)

- Y ↵ (to save file PLAN.WK3)

The use of all three systems, Accounting, Labor, and Job Planning, has been demonstrated through the use of a simple test data set. The procedures described thus far allow for the automation of many of the accounting and reporting functions in the PWD. This aspect of the system is, perhaps, best described as the transaction processing portion of the system. It can also function as a management information system by producing routine reports such as the status of funds report. However, to completely exploit the power of the system, operations beyond those demonstrated thus far must be employed. Chapter V explores two potential operations which capitalize on the decision support capabilities of this system.

V. ILLUSTRATIVE DECISION SUPPORT CAPABILITIES

One of the primary advantages of the spreadsheet environment is the wide variety of powerful tools available to the user for data manipulation and analysis. The number of potential uses for supporting decision making is virtually limitless. Two illustrative procedures will be demonstrated in this chapter. The procedures chosen are intended to be representative of the types of operations which would be useful to decision makers in the PWD. Two examples cannot begin to scratch the surface of the potential usefulness of this type of system to the public works decision maker. The purpose of the illustration is to provide examples of two common types of decision support capabilities which will be useful to public works decision makers, data analysis and graphical representation, which can serve as a springboard for formulating other useful procedures.

A. Variance Analysis

Variance analysis is one of the fundamental tools used for management control in the PWD. This example will demonstrate the techniques required to integrate data from the three public works systems, accounting, labor, and job planning, and how to manipulate that data to provide information useful for decision making. The procedures demonstrated in this section can then be applied to a myriad

of other possible problems. Creation of a variance report for public works maintenance jobs, subfunctional category M1, will be used to demonstrate these procedures. To begin the procedure:

- retrieve the accounting system file (ACCOUNT.WK3)
- make the journal entries shown in Figures 28 and 29
- press [END] to return to the main menu
- W O J (copy M1 journal to scratch pad)

>> Enter SFC or press [ENTER] for reimbursables.

- M1 ◂┘ (select the M1 journal for copying)
- /RNC (to name the extracted database)

>> Enter name to create:

- type MAINT_JRN ◂┘

>> Enter range:

- B:A5..B:O8 ◂┘ (entire database including field names)

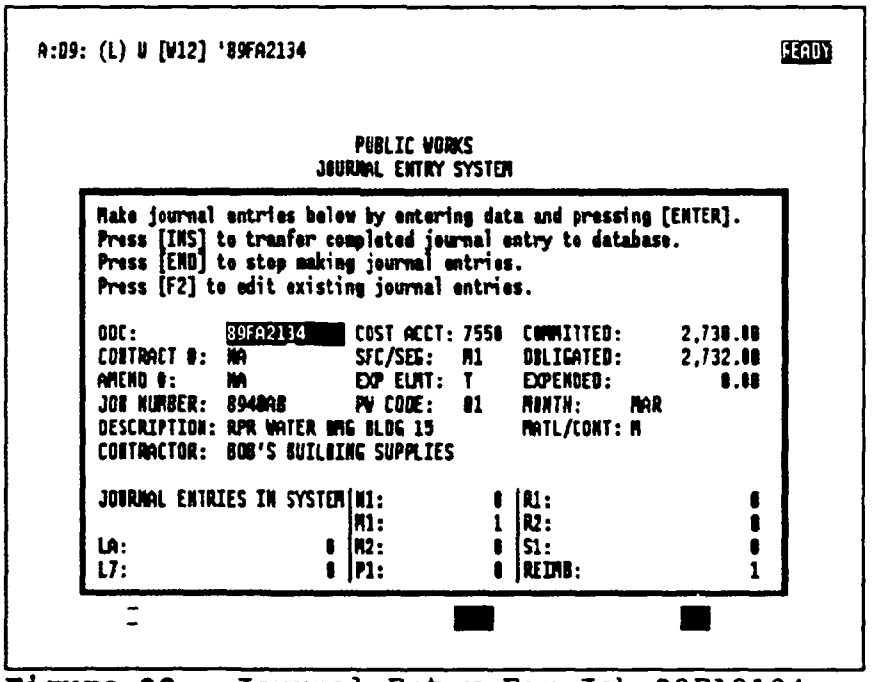


Figure 28. Journal Entry For Job 89FA2134

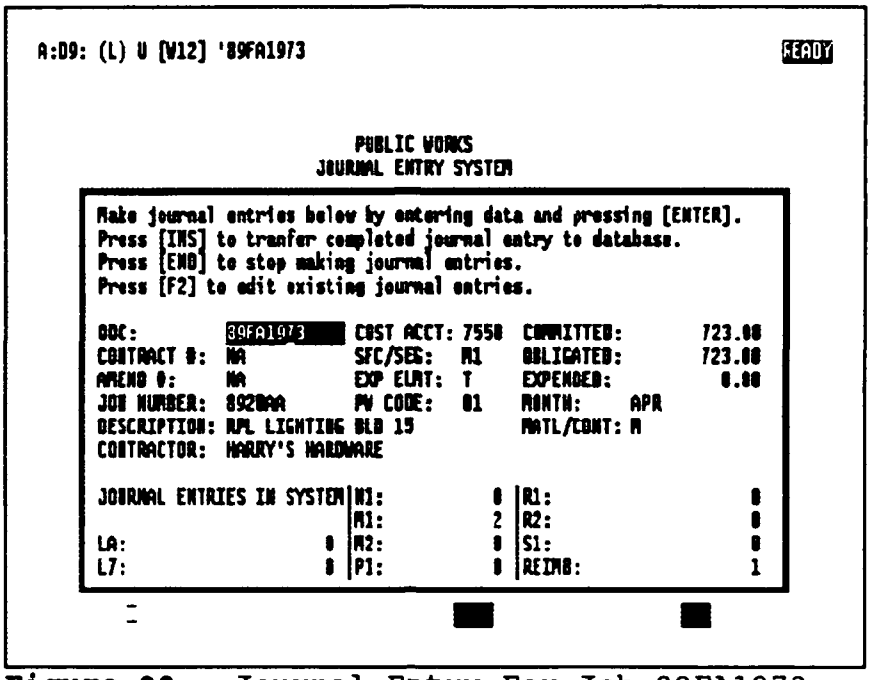


Figure 29. Journal Entry For Job 89FA1973

Note that other journals could have easily been included in the analysis simply by moving the M1 journal three screens to the right, copying the other journal(s) to the scratch pad, and then moving the data from one directly below the other(s) before naming the new database range. It is also important to note that in this simplified example, the M1 journal did not contain any transactions involving contract obligations. In some real world applications it may be desirable to extract only those journal entries related to material obligations. Four simple steps are necessary to accomplish this: In the first step, an output range must be established by copying the field names from the extracted database (M1 journal) to an empty row somewhere in the scratch pad (at least one row or

column away from the existing database). The output range is named using the /RNC command, assigning a name such as MATERIAL, and defining the range to include each of the cells in the new row of field names.

The input range in the existing M1 database which includes transactions involving both materials and contracts and the associated field names. In the second step, it is named using the /RNC command, assigning a name such as BOTH, and defining the range to include the existing field names and associated data.

In the third step, a criteria range must be established to specify that only transactions involving procurement of materials should be extracted. This can be accomplished by typing the field name MATL/CONT in any empty cell above or to the right of the row of field names in the output range (a separation of at least one row or column must be maintained) and then typing M in the cell directly below MATL/CONT. The criteria range is named using the /RNC command, assigning a name such as SPECIFY, and defining the range to include the two cells containing the field name, MATL/CONT, and criteria, M.

The final step in the procedure consists of using 1-2-3 data commands to extract the new database. The new database is extracted using /DQRI, specifying the named input range (BOTH); C, specifying the named criteria range (SPECIFY); O, specifying the named output range; E, extract command; and,

finally Q to remove the data command menu and return to the ready mode. In this case this new database, containing only journal entries related to material procurement, would be used to create the database (named MAINT_JRN) rather than the copy of the M1 journal. However, in the case at hand the M1 journal may be used directly without taking this additional step. Once the database containing material cost data has been extracted and named, it is necessary to save the modified accounting system file, enter the labor system, and create a database containing labor cost information. The procedures necessary to perform these operations are as follows:

- ALT-A (return to the main menu)
- Q Y ◂┘ (save the accounting system file)
- /WEY (erase the worksheet from memory)
- /FR (retrieve the labor system file, LABOR.WK3)
- R B ◂┘ [ENTER] (create a labor cost report)
- W O C [ENTER] (copy the cost report to the scratch pad)
- move the cursor to row 9
- /WDR ◂┘ (delete row 9)
- move the cursor to column B
- /WDC → ◂┘ (delete columns B and C)
- move the cursor to cell B8
- type LABOR ◂┘ (create a field name)
- move the cursor to cell A8
- type JOB_NO ◂┘ (create a field name)
- /RNC (name the new database)

>> Enter name to create:

- COST ◂┘

>> Enter range:

- B:A8..B:B11 ◂┘ (two column database with field names)

The two databases created thus far, MAINT_JRN and COST, are sufficient to determine the labor and material cost for

each M1 job accomplished by the PWD. The final piece of information needed to create a variance report is the planned labor and material costs of each job. This information can be obtained from the planning system. After saving the labor system file, which contains the new database named COST, the a copy of the job planning database must be obtained from the planning system.

- ALT-A (return to main menu)
- Q Y ◂┘ (save labor system file, LABOR.WK3)
- /WEY (erase the worksheet from memory)
- /FR (retrieve the planning system file, PLAN.WK3)
- W O (copy the job planning database to the scratch pad)
- /RNC (name this database)

>> Enter name to create:

- PLANNED ◂┘

>> Enter range:

- B:A5..B:M10 ◂┘ (entire database with field names)
- move the cursor to column L
- /WCCS → ◂┘ 14 ◂┘ (adjust width of columns L & M)

To create a variance report, the three databases just created must be joined to make a new database which shows the job number, planned labor and material costs, actual labor and material costs, and variances. The criteria range used for extracting this new database will include a join formula which determines the way the three databases will be combined. The purpose of the join formula is to specify that the three input databases, MAINT_JRN, COST, and PLANNED, are related by job order number. Therefore, records for which there is a matching job order number in each of the three databases will

be extracted and combined by job order number for inclusion in the variance report. To initialize this procedure, a criteria range must first be established with the required join formula. After establishing the criteria range, the field names for the output range must be defined and the output range must be named. These field names are essentially formulas which determine the way data from the three input databases is combined. Formulas of this type are used to define computed columns in extracted databases.[10] The procedures necessary to accomplish these operations are as follows:

- move the cursor to A15
- type PLANNED.JOB_NO ↓ (field name)
- type +PLANNED.JOB_NO=COST.JOB_NO#AND#PLANNED.JOB_NO= MAINT_JRN.JON ◂┘ (join formula - all on one line)
- /RFT ◂┘ (format cell as text to display formula)
- /RNC (name the criteria range)

>> Enter name to create:

- CRITERIA ◂┘

>> Enter range:

- B:A15..B:A16 ◂┘ (the two cell criteria range)
- move the cursor to A21 to name output database fields
- PLANNED.JOB_NO →
- PLANNED.LABOR →

---

[10]In this procedure, field names in the criteria range and output range include the name of the database from which they came. For example, the field name COST.LABOR refers to the data in a field named LABOR from a database named COST. Field names in the output range which are simple formulas, beginning with +, are used to create a computed column and field names beginning with an @ function are used to create aggregated columns. Refer to the Lotus 1-2-3 reference manual for further explanation.

- COST.LABOR →
- +COST.LABOR-PLANNED.LABOR →
- PLANNED.MATL →
- MAINT_JRN.OBLIGATED →
- +MAINT_JRN.OBLIGATED-PLANNED.MATL →
- +PLANNED.LABOR+PLANNED.MATL →
- +COST.LABOR+MAINT_JRN.OBLIGATED →
- +COST.LABOR+MAINT_JRN.OBLIGATED-
  PLANNED.LABOR-PLANNED.MATL ◄┘
- /C (copy formula to K21)

>> From:

- B:J21 ◄┘ (last cell with formula)

>> To:

- B:K21 ◄┘ (first blank cell)
- move cursor to B:K21
- press [F2] (edit formula)
- [HOME] →
- type (
- [END]
- type)
- type /(COST.LABOR+MAINT_JRN.OBLIGATED) ◄┘
- use /RFT ◄┘ to format cells D21, H21, I21, J21, & K21
- /RNC

>> Enter name to create:

- OUTPUT ◄┘

>> Enter range:

- B:A21..B:K21 ◄┘ (row of new field names)
- move the cursor to cell A21

The screen should now look like Figure 30. The final step necessary to create the variance report is accomplished using 1-2-3 Data commands. These commands will be used to accomplish several tasks. The three databases used as input for the variance report will be joined according to the join formula in the criteria range. The data specified by the

field names in the output database will be entered in the output range, including the computed columns. Two of the databases needed for input are located in separate files which are not in current memory. Lotus 1-2-3 will access these files on the disk if the complete path name, enclosed in double angle brackets (e.g. <<path>>), is included when specifying the database name. The final set of procedures necessary to extract the data required for the variance report is as follows:

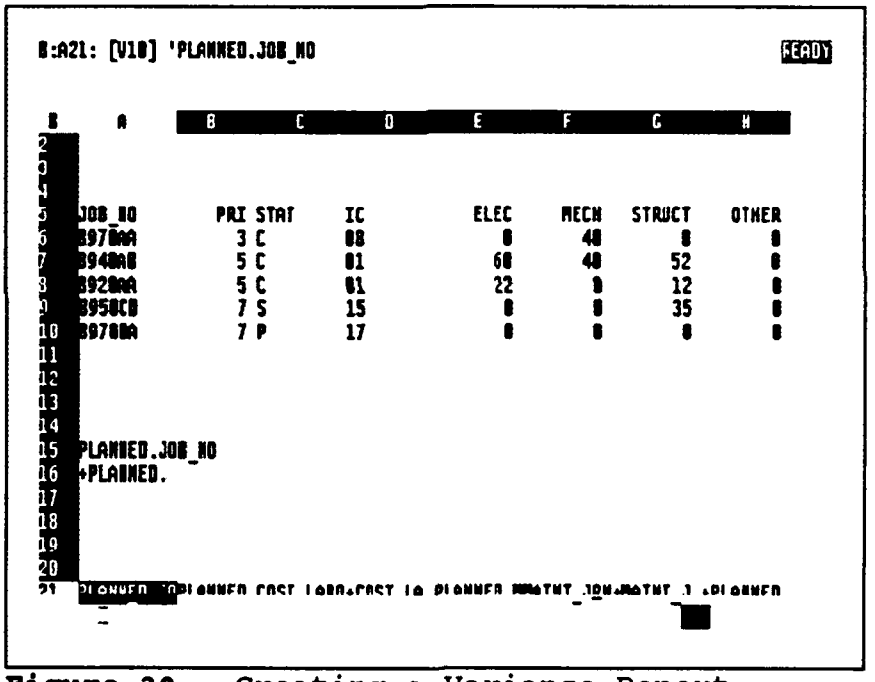


Figure 30. Creating a Variance Report

- /DQR (reset the data query ranges)
- I (specify input range)
- PLANNED,<<LABOR.WK3>>COST,<<ACCOUNT.WK3>>MAINT_JRN ◄┘
- C (specify criteria range)
- CRITERIA ◄┘
- O (specify output range)

- OUTPUT ◂—┘
- E (extract the database and place in output range)
- Q (quit when indicator panel indicates menu)

Data similar to that shown in Figure 31 should have been placed in the output range. If the procedure did not work properly, check the field names and formulas for errors. A copy of the correct field names and formulas is located in a file called VARFORM.WK3. The correct field names and formulas can be viewed in this file by using the / File Open After command to call it into current memory. A copy of the correct format could be obtained from this file and copied to the planning scratch pad for use. The field names and formulas used for this procedure can be copied and saved in a new file for later use. To save the format for later use:

- move the cursor to A:15
- /FNA

>> Enter name of file to create:

- type A:\FORMAT ◂—┘
- press [CTRL-END] [CTRL-PGDN] (return to PLAN.WK3)
- /C [END] ↓ [END] ↓ [END] → ◂—┘
- [CTRL-END] [CTRL-PGUP] ◂—┘
- /FS ◂—┘ R (to save both files to the disk)

The output range now contains a complete variance report for all M1 jobs. New column titles can be created, column widths adjusted, and cell formats changed, as shown in Figure 31, to create a final variance report for printing. An exception report could be created using this new database as an input range. A criteria range would be established to specify the minimal variation requiring further investigation,

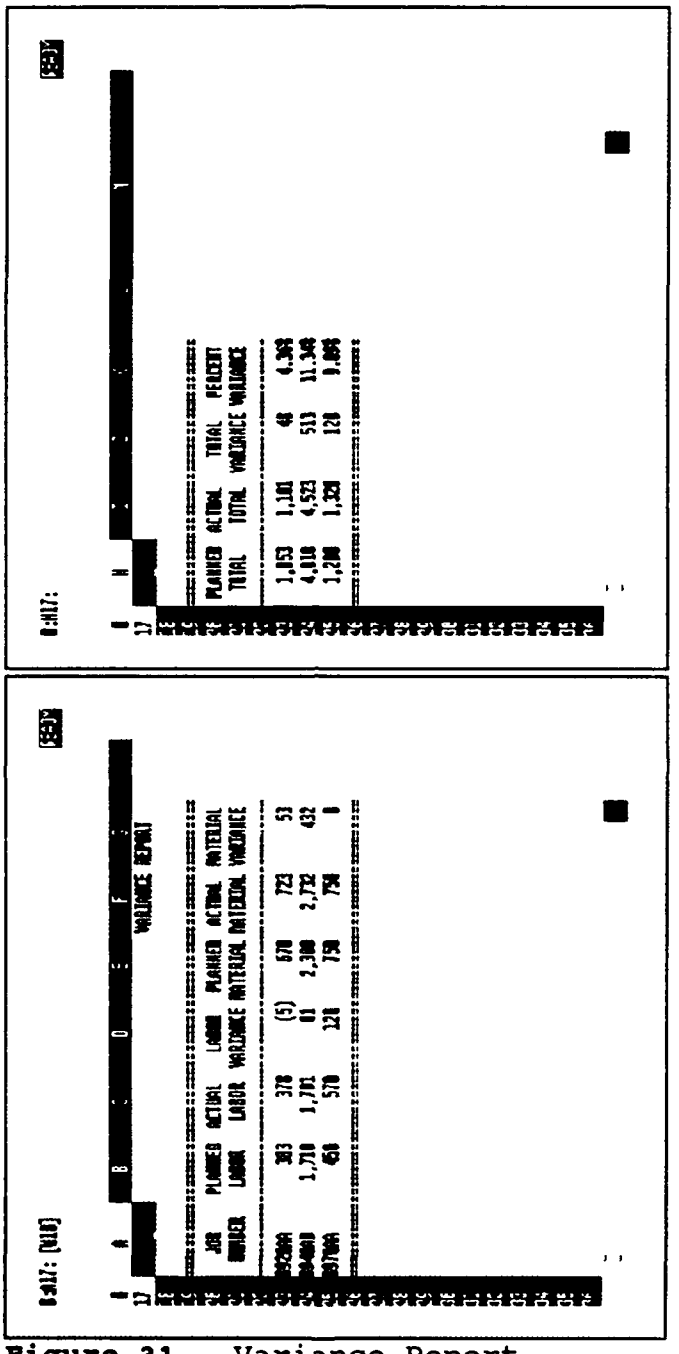


Figure 31. Variance Report

for example a total variance greater than $100 for jobs under $1000, or a percentage variance greater than 10% for jobs $1000 and above. Finally, an output range with appropriate field names would be established and the /Data commands would be used to extract data on only those jobs requiring further investigation. Lotus 1-2-3 also includes numerous commands which could be used to perform various types of statistical analysis on the data in the variance report.

B. Graphical Representations

Graphical representations can greatly enhance the decision maker's ability to conceptualize a problem or spot trends in historical data. There are many areas where this could be useful in the PWD, including the use of bar charts which compare actual vs. planned costs or labor hours for each work center, line graphs showing planned vs. actual spending rates, or pie charts showing the distribution of completed jobs by labor class codes. In this section, the procedures for using on screen graphical representations to support a resource allocation decision will be illustrated. Suppose the PWD would like to review its resource commitments to date to evaluate them for consistency with mission area priorities and guide in the selection of future projects.

- ALT-A (to return to the main menu)
- W E (to erase the scratch pad)
- W O (to extract a new copy of the job planning database)

The first step in the analysis is to aggregate resource allocations by investment category. To initialize the

procedure, input, criteria, and output ranges must be established as before. The criteria for extraction will be all those jobs in the job planning database which have already been completed. These jobs are identified with a C in the STAT (status) field. The input range is the entire job planning database, including field names. The output range will consist of two field names. The first field in the output range will be the Investment Category, IC, and the second will be a formula used to create an aggregated column. The following procedures are used to accomplish theses operations:

- /C B:C5 ◂┘ B:A15 ◂┘ (copy field name STAT)
- /RNC CRITERIA ◂┘ B:A15..B:A16 ◂┘ (criteria range)
- move cursor to A16
- type C ◂┘ (extract only completed jobs)
- move cursor to A5
- /RNC INPUT ◂┘ B:A5..B:M10 ◂┘ (input range)
- move cursor to A21 (to name output fields)
- type IC →
- @SUM(TOTAL_COST) ◂┘
- /RFT (format as text to display formula)
- /RNC OUTPUT ◂┘ B:A21..B:B21 ◂┘ (output range)
- /DQR (reset Data Query ranges)
- I (set input range)
- INPUT ◂┘
- C (set criteria range)
- CRITERIA ◂┘
- O (set output range)
- OUTPUT ◂┘
- E Q (extract aggregated database and remove menu)

Move the cursor to A24 to view the extracted records. The total cost column shows data extracted only for the two investment categories for which jobs have been completed. The

cost of individual jobs is aggregated by investment category. Enter the data shown in TABLE 17, typing over the extracted data, to provide enough data points to generate a meaningful graph. Enter the investment categories (IC) as labels rather than values. For example, IC 01 is entered by typing '01.

TABLE 17 COST DATA BY INVESTMENT CATEGORY

IC	TOTAL COST
01	50,630
05	29,300
06	57,980
08	12,000
14	11,930
15	2,035
17	18,500

- move the cursor to any cell in between A22 and A30
- /GRGTPV (use 1-2-3 auto-graphing to create a pie chart)

A pie chart, showing the percent of resources allocated to each of the investment categories, as shown in Figure 32 should now appear on the screen.

- press any key to return to the worksheet
- Q (to remove the graph menu)

This pie chart, which can easily be printed out using 1-2-3 commands from within the current worksheet, provides a useful representation of the resource allocation distribution by investment category for those jobs completed to date. However, representation which would be much more useful for decision making is one which could be seen on the screen along side the data it represented. In fact this would be a very

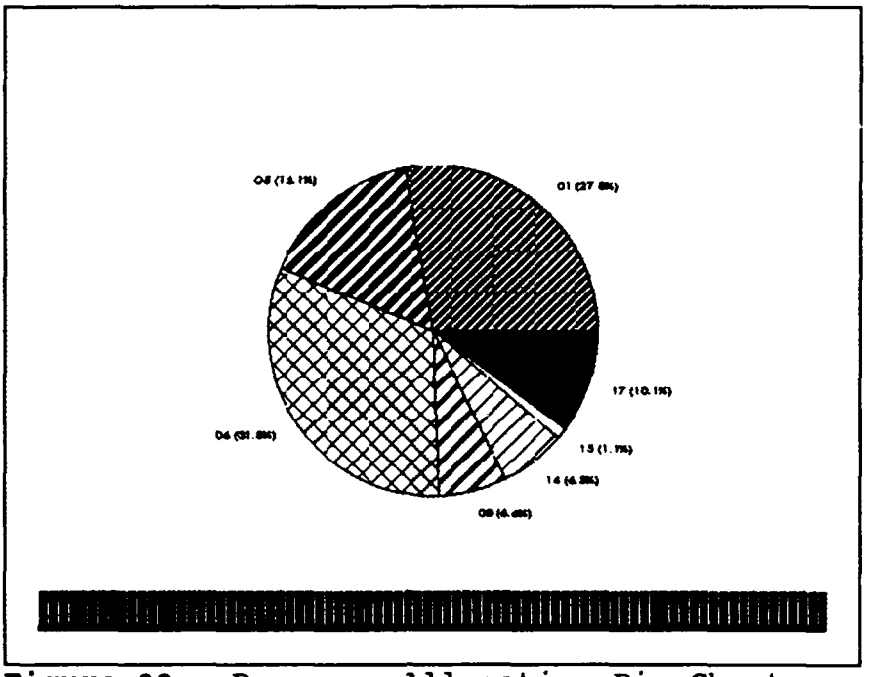


Figure 32. Resource Allocation Pie Chart

powerful tool which could be used to support decision making if the graphical representation, which is shown along side the data it represents, changes on the screen in response to changes in the data. This can be accomplished using the following simple steps:

- move the cursor to column C
- /WWG (create a graph window)

The pie chart should now appear in the graph window on the worksheet as shown in Figure 33. Numerical and graphical results of various resource allocation decisions can now be seen simultaneously. For example, examine the effects of executing a job in investment category 15 for $13,300.

- move the cursor to B:B27
- press [F2] (edit)

- +13300 ◄┘ (add 13,300 to the current entry)

Notice the effects that this change, which appears immediately in the pie chart on the screen, has on the resource allocation distribution. Investment category 15 would receive 7.8%, up from 1.1%, of the total resources if this job was executed.

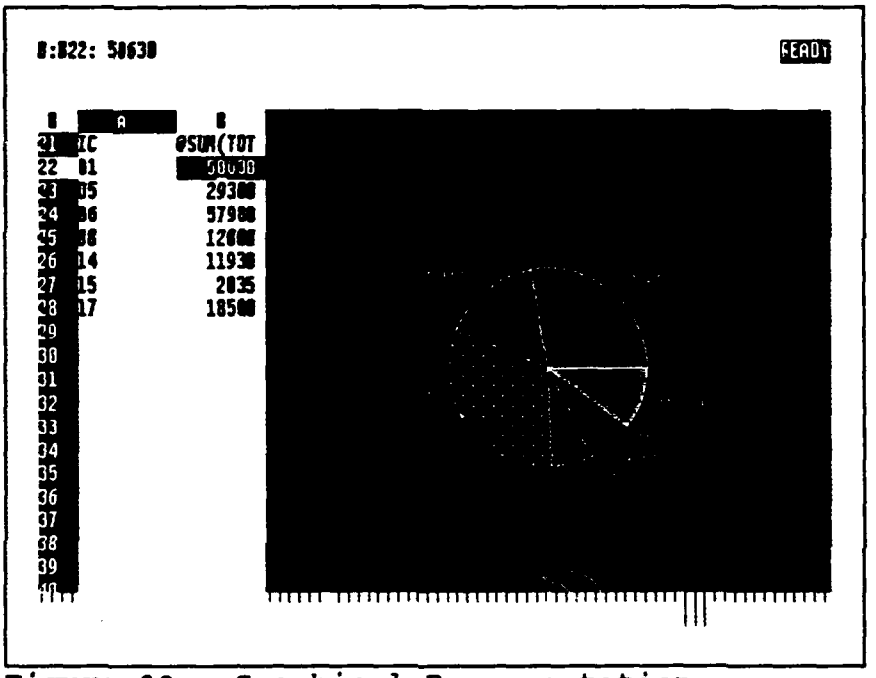


Figure 33. Graphical Representation

- /WWC (to clear the graph window)
- ALT-A (return to main menu)
- W E (to erase the scratch pad)
- Q (end the session and save the system file, PLAN.WK3)

It should again be emphasized that these are only two examples, of a nearly infinite variety, illustrating how this system can be used to support decision making. Additional research is needed to document other uses for this type of system within Navy Public Works Departments. During field

testing of the prototype, and systems implementation, numerous uses will undoubtedly be suggested by users. Research to pull together these suggestions, automate commonly used routines, and document potential operations for common use by PWD's is an integral part of the systems development process which is necessary to fully exploit the benefits offered by a system of this type. This subject will be discussed in more detail in Chapter VI. The software developed in this thesis is available through Dr. Shu S. Liao, Administrative Science Department, Naval Postgraduate School, Monterey California. The macros, range names, and range addresses developed for each of the three system files are presented in Appendixes A through C.

VI. CONCLUSION

This thesis examined the feasibility of developing a generally applicable financial accounting and reporting system for small Navy public works departments (PWD's). A review of the decision support systems (DSS) literature was used to establish the framework for systems development. The Naval Postgraduate School Public Works Department provided much of the information used to define data requirements for the system. Finally, Lotus 1-2-3 Release 3.0 was used to develop a prototype financial decision support system for use in Navy public works departments.

A. RESEARCH RESULTS

This thesis posed two primary research questions.

- Can a generally applicable model for financial accounting and reporting be developed for PWD's using a commercially available decision support system generator such as Lotus 1-2-3?

The model, developed in Lotus 1-2-3 Release 3.0, as a part of this research allows this question to be answered positively in the affirmative. Recent technological developments in microcomputers, along with significant additions to the capabilities of the newly released version of Lotus 1-2-3, have provided the necessary hardware and software tools for development of such a system.

- Can such a model be used to apply decision support system theory to financial management within the PWD?

This question can also be answered in the affirmative. DSS theory provided the framework for systems development of this model. The system supplies a user friendly interface for database management in the powerful model-based, analysis oriented, environment provided by Lotus 1-2-3 Release 3.0. This allows the user tremendous flexibility for data manipulation and analysis, thereby offering the potential to significantly enhance the effectiveness of decision making for financial management in the PWD. However, this thesis does not explore all the implications of this result. Additional research is needed to more fully develop the applications of such a model for financial management in the PWD. This topic will be discussed in more detail later in this chapter.

B. ANALYSIS OF THE RESULTS

The PWD generates a tremendous amount of financial and accounting data. Systems developed to handle this large volume of data have traditionally relied on database management software. This type of software has the advantage of user friendly data entry and efficient data processing mechanisms. The data entry systems normally perform routine edit tests and the database is essentially transparent to the user. These features generally serve to enhance data integrity, in addition to providing for easy data manipulation. In the database management environment, a systems analyst, or experienced user, normally develops the

program steps necessary to make routine calculations and print a report. However, as a result of this purely data processing environment, the manager frequently has little interest in the system, beyond the fact that it may increase efficiency in the accounting section.

The spreadsheet environment, while it has always excelled at providing a flexible tool for calculation and analysis, has not offered the same advantages in database management capabilities. This thesis shows how recent technological advancements in both hardware and software can be exploited to capitalize on the advantages of both environments. Data input screens and menu systems have been developed to provide a user friendly interface. The database is essentially transparent to the user except that in this system the user can view the database and even extract a complete copy of it for manipulation and analysis in the workspace provided. This additional flexibility is provided without compromising database integrity since the user cannot make modifications to the database directly. All modifications must be made through the data input screens so that proper edit tests can be performed before the data is accepted and added to the database.

Most of the disadvantages of working with large amounts of data in a spreadsheet environment have been overcome by this model. Overcoming these problems makes it possible to apply all the power and flexibility of the spreadsheet

environment to the PWD financial management problem. The previously invisible and mysterious database is made directly available to the user along with all the tools necessary for calculation and analysis. The accounting data in this model is organized in the types of journals and ledgers that are familiar to the users. Additionally, the ledger balances can be quickly and easily viewed on the screen without waiting for the creation of some report. An almost endless variety of reports can be easily developed and customized without retyping data from another report.[11] The decision maker is no longer limited by the specific program steps developed in the database management environment. The user has easy access to sophisticated computational functions and tools, statistical analysis, data manipulation and database management functions, graphical representation, the ability to create new files, and the ability to easily develop macros to automate customized procedures. These capabilities allow the system to go far beyond improving the efficiency of the accounting section, they provide a mechanism with the potential to significantly enhance the decision making process.

---

[11]Accounting personnel frequently use reports gererated by database management systems as input, which is then manually typed into a spreadsheet, to generate customized summary reports for management.

C. RECOMMENDATIONS

The model developed in this thesis should be viewed, and used, as a prototype. Due to time limitations, the model developed here was not field tested. Field testing of the prototype should be done to assist in discovering any bugs in the system, ensure general applicability, and to generate the necessary user feedback for complete system development. User comments on the interface and capabilities of the system should be incorporated where feasible. The final system should be developed using the Lotus Add-in Toolkit. This toolkit allows for development of new @functions, macro keywords, and input forms. It provides both the program editor, compiler, and debugger necessary to develop source code in Lotus Programming Language and to compile it into a fully integrated Lotus 1-2-3 Release 3.0 Add-inn program. Once developed, a standard, microcomputer based financial management system should be provided to all Navy PWD's.

The system should be designed around the concept of compatibility with the Base Engineering Support, Technical (BEST) system as well as any other software being developed under the Public Works Management Automation Program. This can be done easily if these other systems use a dBASE, or similar type, file structure since Lotus 1-2-3 is capable of reading in, or manipulating externally, database files of this type. The system should be designed for multiple users in a microcomputer based, local area network environment. This

will reduce data redundancy problems and be consistent with the direction chosen by the Naval Facilities Engineering Command for public works management automation (NAVFAC letter, 1989).

D. AREAS FOR FURTHER RESEARCH

This thesis has brought out the need for further research in several areas. The first area for consideration should be field testing of the prototype and complete development of system requirements definitions. The field testing should emphasize use of the system as a DSS. A researcher knowledgeable of the potential uses and capabilities of the system should work with public works managers to further define the types of decision support desired and identify candidate functions for further automation.

One area which could be easily added to the model, and may have tremendous potential for decision making support in the public works environment, is the use of linear and dynamic programming. One potential area for examination is the use of interactive linear programming for resource allocation decisions. The central research question would focus on the feasibility of developing an annual maintenance plan by optimizing an objective function of assigned job priorities given various manpower and financial constraints. The system should function in an interactive manner so that the decision maker could incorporate knowledge of subjective information regarding job priority and assignment, modify assigned

priorities, and iterate the optimization algorithm until an acceptable "optimal" solution is reached. There are numerous potential applications for dynamic programming in the PWD as well. James Ho (1987) has developed the necessary macros to apply these powerful tools in Lotus 1-2-3. Future research could focus on the applicability of these two models to public works management.

An additional area for future research is to examine the use of various available add-in programs to enhance the capabilities of the system. One area that could be studied as an alternative to developing the system as an add-in using source code, is to use commercially available database add-in programs such as D.A.V.E., a data entry program, or @Base, a data management program. Additionally, there are numerous add-in programs, for such things as construction management, financial management, statistical analysis, and management science applications, which may significantly enhance the ability of the system to provide useful decision support for the public works manager.

APPENDIX A

Appendix A contains the macros and range names used in the accounting system file. The macros are presented in the same spreadsheet format as they were written in the system file. The letter designation in the upper left corner on the first page of macros indicates the worksheet which contains the system macros. Column A list the name of each routine. The number to the left of the routine name is the row number on which that routine begins in the system file. Columns B through F contain the program code, or macros, used to run the system. In the actual system file, each cell may contain a long line of program code, Lotus 1-2-3 allows up to 520 characters per cell. However, a 27 character per cell format has been adopted here for the purpose of presentation. A continuation character, ►, has been used to indicate that the information on the next line is actually located in the same cell as the current line of code. Finally, the range names and addresses are listed in the tables following the macros. Lotus 1-2-3 Release 3.0, a three dimensional spreadsheet, designates cells using the first letter(s) to indicate the worksheet, the letter(s) following the colon to indicate the column, and the number to indicate the row of a specific cell location. A range is specified by the upper left corner cell location, two dots, and the lower right cell location.

AX	A	B	C	D	E	F
1	\0	{FRAMEOFF}{WINDOWSOFF}{FC}{PGDN}{WINDOWSON} {MENUBRANCH SELECT}				
4	SELECT	Journal Make journal entries {BRANCH JOURNAL} {BRANCH \0}	Ledger Enter fund authorizations ► (beginning ► account balances) {BRANCH LEDGER} {BRANCH \0}	Codes Enter a list of ► authorized SEG codes {BRANCH CODES} {BRANCH \0}	Work Scratch Other Erase {PANELOFF}{WINDOWSOFF} ► /RNCSCRATCH~B:A1~ ► /RNCSP~B:D1~/RNCPAD~B:A5~ /RNCRPT_LED~C:A1~ ► /RNCLED_TITL~C:C1~ ► /RNCTEMP~C:A4~ {GOTO}SCRATCH~ ► /RE{END}{HOME}~ {LET SCRATCH,@TODAY} ► {LET SP,ACCOUNTING ► SYSTEM SCRATCH PAD} {MENUCALL WRK} {BRANCH \0}	Quit End session and ► return to 1-2-3. {GETLABEL "Do you want ► to save your work? ► (y/n) ",CHOICE} ► {IF CHOICE="N"}{QUIT} {PANELOFF}/FS{QUIT}
13	\A	{BRANCH \0}				
15	JOURNAL	{FRAMEOFF}{BLANK INPUT_A}{BLANK INPUT_B}{BLANK INPUT_C} {WINDOWSON}{PANELON}{FORM ENTRYFORM,SIGKEYS}				
18	SIGKEYS	{INS} {END} ~ {EDIT}	{ESC}{ESC}{TRANS}{EVAL}{BRANCH JOURNAL} {BRANCH CONFIRM} {D} {ESC}{ESC}{MENUCALL EDIT_JRN}{PANELON}{WINDOWSON}			
23	ERR_MSG					
25	EDIT_JRN	Modify Modify an existing ► journal entry. {LET RETRN,MODIFY}	Delete Delete a specified ► journal entry. {LET RETRN,DELETE}	Browse View a specified journal. {LET RETRN,BROWSE} {GET_DB} {BLANK RETRN}	Erase Specified All {MENUBRANCH ERAS_JRN} {RESTART}{BRANCH JOURNAL}	
28	RETRY	{GETLABEL "Enter ► SFC/SEG. ",SFC} {IF SFC<>"M1"#AND#SFC<> ► "P1"#AND#SFC<>"L7"#AND# ► SFC<>"R1"#AND#SFC<>"S1" ► #AND#SFC<>"M2"#AND#SFC<> ► "R2"}{GETLABEL "Enter ► month. ",MTH} {IF @LEFT(MTH,3)<>"JAN" ► #AND#@LEFT(MTH,3)<>"FEB" ► #AND#@LEFT(MTH,3)<>"MAR" ►	{RETRY} {GETLABEL "Are you sure ► you want to delete this ► journal entry? ► (y/n) ",CHOICE} {IF CHOICE="N"}{PANELOFF} ► /DQMCQ{RESET} ► {RESTART}{BRANCH JOURNAL} {PANELOFF}/DQDDQ ► /DQDDQ{WINDOWSOFF} {RESET} ►	{D}/WTH {BRANCH END_VIEW} {RESTART}{BRANCH JOURNAL}		

```
          #AND#@LEFT(MTH,3)<>"APR" ►  {RESTART}{BRANCH JOURNAL}
          #AND#@LEFT(MTH,3)<>"MAY" ►
          #AND#@LEFT(MTH,3)<>"JUN" ►
          #AND#@LEFT(MTH,3)<>"JUL" ►
          #AND#@LEFT(MTH,3)<>"AUG" ►
          #AND#@LEFT(MTH,3)<>"SEP" ►
          #AND#@LEFT(MTH,3)<>"OCT" ►
          #AND#@LEFT(MTH,3)<>"NOV" ►
          #AND#@LEFT(MTH,3)<>"DEC") ►
          {BRANCH BAD_MTH}
          {GETLABEL "Enter  ►
          ODC.  ",ODC)
          {PICK_DB}
          {ONERROR JRN_ERR,ERR_MSG}
          {PANELOFF}/DQRI
35 PLACE2
          ~CCRIT_JRN~
          OOUT_JRN~MEQ
          {WINDOWSOFF}
          {IF @CELL("TYPE", ►
          BLANK_CHK)="b"} ►
          {BRANCH JRN_ERR}
          {IF @ROWS(OUT_JRN)>2} ►
          {DUP_WARN}
          /RTTRANS_A~INPUT_A~
          /RTTRANS_B~INPUT_B~
          /RTTRANS_C~INPUT_C~
          {WINDOWSON}{PANELON}
          {IF RETRN="DELETE"} ►
          {BLANK RETRN}{RETURN}
          {GETLABEL "Press [INS]  ►
          when done, [END] to stop. ►
            Press [ENTER] to  ►
          continue.",CHOICE)
          {FORM ENTRYFORM,MOD_KEYS}
48 FIN_MOD  {IF SFC<>"M1"#AND#SFC<> ►
          "P1"#AND#SFC<>"L7"#AND# ►
          SFC<>"R1"#AND#SFC<>"S1" ►
          #AND#SFC<>"M2"#AND#SFC<> ►
          "R2"#AND#SFC<>"LA"#AND# ►
          SFC<>"N1"){VALID}
          {WINDOWSOFF}{PANELOFF}
          /DQMRQ{RESET}
          {BLANK RETRN} ►
          {RESTART}{BRANCH JOURNAL}

53 DUP_WARN  {BEEP}{GETLABEL "Warning:  multiple journal entries with same ODC.  Press [ENTER] to continue.",CHOICE}
          {GETLABEL "Do you want to view extracted records?  (y/n)  ",CHOICE}{IF CHOICE="N"}{RETURN}
          {GOTO}OUT_JRN~{WINDOWSON}{GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to return to input screen.  ",CHOICE}
```

```
          (?)(WINDOWSOFF)(FC)(RETURN)

58 RESET  /RNCOUT_JRN~(ESC).(END)(R)~
          (RETURN)

61 MOD_KEYS -                          (D)
          (INS)                        (ESC)(TRANS)(RESTART)(BRANCH FIN_MOD)
          (END)                        (ESC)(END_MOD)

65 END_MOD (GETLABEL "Cancel modification and return to entry screen?  (y/n)  ",CHOICE)
          (IF CHOICE="Y")(WINDOWSOFF)(PANELOFF)/DQMCQ(RESET)(RESTART)(BRANCH JOURNAL)
          (RETURN)

69 PICK_DB (IF SFC="M1")(LET PLACE2,"MAINT_DB")(RETURN)
          (IF SFC="P1")(LET PLACE2,"ENGR_DB")(RETURN)
          (IF SFC="L7")(LET PLACE2,"TRANS_DB")(RETURN)
          (IF SFC="R1")(LET PLACE2,"CONST_DB")(RETURN)
          (IF SFC="S1")(LET PLACE2,"SUPPT_DB")(RETURN)
          (IF SFC="M2")(LET PLACE2,"SP_RPR_DB")(RETURN)
          (IF SFC="R2")(LET PLACE2,"SP_CONST_DB")(RETURN)
          (IF SFC="LA")(LET PLACE2,MTH&"_PHONE")(RETURN)
          (IF SFC="N1")(LET PLACE2,MTH&"_UTIL")(RETURN)
          (LET PLACE2,MTH&"_REIMB")(RETURN)

80 BAD_MTH (BEEP)(GETLABEL "Month not correctly specified.  Try again? (y/n)  ",CHOICE)
          (IF CHOICE="N")(RESTART)(BRANCH JOURNAL)
          (BRANCH RETRY)

84 JRN_ERR (BEEP)(GETLABEL "Journal entry not found.  Try again?  (y/n)  ",CHOICE)
          (IF CHOICE="Y")(BRANCH RETRY)
          /DQMCQ(RESET)(RESTART)(BRANCH JOURNAL)

88 ERAS_JRN Specified                  All
          Erase all journal  ▸         Erase all journal  ▸
          entries from a  ▸            entries from all journals.
          specified journal.           (GETLABEL "Are you sure  ▸
          (LET RETRN,ERASE)            you want to erase all  ▸
          (WINDOWSOFF)(PANELOFF)       journals?  (y/n)  ",CHOICE)
          (GET_DB)                     (IF CHOICE="N") ▸
          (BLANK RETRN)                (RESTART)(BRANCH JOURNAL)
          (GETLABEL "Are you sure  ▸   (PANELOFF)(WINDOWSOFF)
          you want to erase the  ▸     (GOTO)MAINT_DB~(D)/WDR(END)(D)~
          journal entries?  ▸          (GOTO)ENGR_DB~(D)/WDR(END)(D)~
          (y/n)  ",CHOICE)             (GOTO)TRANS_DB~(D)/WDR(END)(D)~
          (IF CHOICE="N") ▸            (GOTO)CONST_DB~(D)/WDR(END)(D)~
          (RESTART)(BRANCH JOURNAL)    (GOTO)SUPPT_DB~(D)/WDR(END)(D)~
          (D)/WDR                      (GOTO)SP_RPR_DB~(D)/WDR(END)(D)~
          (END)(D)~                    (GOTO)SP_CONST_DB~(D)/WDR(END)(D)~
          (FC) ▸                       (LET CHOICE,_PHONE)(ERAS_MTH)
```

```
        {RESTART}{BRANCH JOURNAL}   {LET CHOICE,_UTIL}{ERAS_MTH}
                                    {LET CHOICE,_REIMB}{ERAS_MTH}
                                    {FC}{RESTART}{BRANCH JOURNAL}

105 ERAS_MTH  {LET PLACE,+"JAN"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"FEB"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"MAR"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"APR"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"MAY"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"JUN"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"JUL"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"AUG"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"SEP"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"OCT"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"NOV"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {LET PLACE,+"DEC"&CHOICE}{LOC}{D}/WDR{END}{D}~
              {RETURN}

119 RETRN

121 GET_DB    {PANELOFF}{GETLABEL "Enter desired SFC, or press [ENTER] for reimbursables. ",SFC}
              {IF SFC<>"M1"#AND#SFC<>"P1"#AND#SFC<>"L7"#AND#SFC<>"R1"#AND#SFC<>"S1"#AND#SFC<>"M2"#AND#SFC<>"R2"} ▸
              {GETLABEL "Enter month.  ",CHOICE}{BRANCH VIEW_MTH}
              {IF SFC="M1"}{GOTO}MAINT_DB~{HOME}{D 2}{RETURN}
              {IF SFC="P1"}{GOTO}ENGR_DB~{HOME}{D 2}{RETURN}
              {IF SFC="L7"}{GOTO}TRANS_DB~{HOME}{D 2}{RETURN}
              {IF SFC="R1"}{GOTO}CONST_DB~{HOME}{D 2}{RETURN}
              {IF SFC="S1"}{GOTO}SUPPT_DB~{HOME}{D 2}{RETURN}
              {IF SFC="M2"}{GOTO}SP_RPR_DB~{HOME}{D 2}{RETURN}
130 VIEW_MTH  {IF SFC="R2"}{GOTO}SP_CONST_DB~{HOME}{D 2}{RETURN}
              {IF @LEFT(MTH,3)<>"JAN"#AND#@LEFT(MTH,3)<>"FEB"#AND#@LEFT(MTH,3)<>"MAR" ▸
              #AND#@LEFT(MTH,3)<>"APR"#AND#@LEFT(MTH,3)<>"MAY"#AND#@LEFT(MTH,3)<>"JUN" ▸
              #AND#@LEFT(MTH,3)<>"JUL"#AND#@LEFT(MTH,3)<>"AUG"#AND#@LEFT(MTH,3)<>"SEP" ▸
              #AND#@LEFT(MTH,3)<>"OCT"#AND#@LEFT(MTH,3)<>"NOV"#AND#@LEFT(MTH,3)<>"DEC"}{BRANCH MTH_ERR}
              {IF SFC="LA"}{LET PLACE,CHOICE&"_PHONE"}{LOC}{BRANCH @LEFT(RETRN,4)}
              {IF SFC="N1"}{LET PLACE,CHOICE&"_UTIL"}{LOC}{BRANCH @LEFT(RETRN,4)}
              {LET PLACE,CHOICE&"_REIMB"}{LOC}{BRANCH @LEFT(RETRN,4)}

135 LOC       {ONERROR MTH_ERR}{GOTO}
136 PLACE
              ~{HOME}{D 3}{RETURN}

139 MTH_ERR   {BEEP}{GETLABEL "Error:  month not correctly specified.  Type correct month. ",CHOICE}
              {BRANCH VIEW_MTH}

142 END_VIEW  {GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to return to input screen. ",CHOICE}
              {?}/WTC{PANELON}{RESTART}{BRANCH JOURNAL}

145 CONFIRM   {GETLABEL "Stop making journal entries?  (y/n)  ",CHOICE}
```

```
        {IF CHOICE="Y"}{RESTART}{BRANCH \0}
        {RESTART}{BRANCH JOURNAL}

149 CHOICE

151 TRANS   {IF @CELL("TYPE",INPUT_ODC)="b"}{PANELOFF}{GOTO}INPUT_ODC~{BRANCH EMPTY}
        {IF @CELL("TYPE",INPUT_JON)="b"}{PANELOFF}{GOTO}INPUT_JON~{BRANCH EMPTY}
        {IF @CELL("TYPE",CA)="b"}{PANELOFF}{GOTO}CA~{BRANCH EMPTY}
        {IF @CELL("TYPE",INPUT_SFC)="b"}{PANELOFF}{GOTO}INPUT_SFC~{BRANCH EMPTY}
        {IF @LENGTH(DESCRIPTION)>25}{PANELOFF}{GOTO}DESCRIPTION~{BRANCH CHAR}
        {IF @LENGTH(CONTRACTOR)>25}{PANELOFF}{GOTO}CONTRACTOR~{BRANCH CHAR}
        {IF @CELL("TYPE",COMMITTED)<>"v"}{PANELOFF}{GOTO}COMMITTED~{BRANCH VALUE}
        {IF @CELL("TYPE",OBLIGATED)<>"v"}{PANELOFF}{GOTO}OBLIGATED~{BRANCH VALUE}
        {IF @CELL("TYPE",EXPENDED)<>"v"}{PANELOFF}{GOTO}EXPENDED~{BRANCH VALUE}
        {IF @CELL("TYPE",MC)="b"}{PANELOFF}{GOTO}MC~{BRANCH FIX_MC}
        {IF MC<>"M"#AND#MC<>"C"}{PANELOFF}{GOTO}MC~{BRANCH FIX_MC}
        {PANELOFF}{WINDOWSOFF}
        /RTINPUT_A~TRANS_A~
        /RTINPUT_B~TRANS_B~
        /RTINPUT_C~TRANS_C~
        {PANELON}{WINDOWSON}
        {RETURN}

169 EMPTY   {BEEP}{GETLABEL "Error:  this field requires an entry.  Press [ENTER] to continue.",CHOICE}
        {PANELON}{?}~{BRANCH TRANS}

172 CHAR    {BEEP}{GETLABEL "Error:  field length cannot exceed 25 characters.  Press [ENTER] to continue.",CHOICE}
        {PANELON}{?}~{BRANCH TRANS}

175 VALUE   {BEEP}{GETLABEL "Error:  this field requires a numerical entry.  Press [ENTER] to continue.",CHOICE}
        {PANELON}{?}~{BRANCH TRANS}

178 FIX_MC  {BEEP}{GETLABEL "Error:  enter M for material or C for contracts.  Press [ENTER] to continue.",CHOICE}
        {PANELON}{?}~{BRANCH TRANS}

181 EVAL    {IF @LEFT(SFC_SEG,2)="M1"}{APPENDBELOW MAINT_DB,INPUT_AREA}{RETURN}
        {IF @LEFT(SFC_SEG,2)="LA"}{LAMONTH}{RETURN}
        {IF @LEFT(SFC_SEG,2)="N1"}{N1MONTH}{RETURN}
        {IF @LEFT(SFC_SEG,2)="P1"}{APPENDBELOW ENGR_DB,INPUT_AREA}{RETURN}
        {IF @LEFT(SFC_SEG,2)="L7"}{APPENDBELOW TRANS_DB,INPUT_AREA}{RETURN}
        {IF @LEFT(SFC_SEG,2)="R1"}{APPENDBELOW CONST_DB,INPUT_AREA}{RETURN}
        {IF @LEFT(SFC_SEG,2)="S1"}{APPENDBELOW SUPPT_DB,INPUT_AREA}{RETURN}
        {IF @LEFT(SFC_SEG,2)="M2"}{CHK_SP}{APPENDBELOW SP_RPR_DB,INPUT_AREA}{RETURN}
        {IF @LEFT(SFC_SEG,2)="R2"}{CHK_SP}{APPENDBELOW SP_CONST_DB,INPUT_AREA}{RETURN}
        {VALID}{REIMONTH}
        {RETURN}

193 CHK_SP  {PANELOFF}{GOTO}ENTER_JON~{GETLABEL "Did you enter the special project number in JON?  (y/n)  ",CHOICE}
        {PANELON}{IF CHOICE="Y"}{RETURN}
```

```
        {GETLABEL "Enter special project number and press [ENTER]  ",ENTER_JON}{RETURN}

197 LAMONTH  {IF @LEFT(MONTH,3)="JAN"}{APPENDBELOW JAN_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="FEB"}{APPENDBELOW FEB_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="MAR"}{APPENDBELOW MAR_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="APR"}{APPENDBELOW APR_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="MAY"}{APPENDBELOW MAY_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="JUN"}{APPENDBELOW JUN_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="JUL"}{APPENDBELOW JUL_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="AUG"}{APPENDBELOW AUG_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="SEP"}{APPENDBELOW SEP_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="OCT"}{APPENDBELOW OCT_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="NOV"}{APPENDBELOW NOV_PHONE,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="DEC"}{APPENDBELOW DEC_PHONE,INPUT_AREA}{RETURN}
             {GOTO}MONTH~
             {BEEP}{GETLABEL "Error:  month not correctly specified.  Press [ENTER] to continue.",CHOICE}
             {?}~{TRANS}{BRANCH LAMONTH}

213 N1MONTH  {IF @LEFT(MONTH,3)="JAN"}{APPENDBELOW JAN_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="FEB"}{APPENDBELOW FEB_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="MAR"}{APPENDBELOW MAR_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="APR"}{APPENDBELOW APR_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="MAY"}{APPENDBELOW MAY_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="JUN"}{APPENDBELOW JUN_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="JUL"}{APPENDBELOW JUL_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="AUG"}{APPENDBELOW AUG_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="SEP"}{APPENDBELOW SEP_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="OCT"}{APPENDBELOW OCT_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="NOV"}{APPENDBELOW NOV_UTIL,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="DEC"}{APPENDBELOW DEC_UTIL,INPUT_AREA}{RETURN}
             {GOTO}MONTH~
             {BEEP}{GETLABEL "Error:  month not correctly specified.  Press [ENTER] to continue.",CHOICE}
             {?}~{TRANS}{BRANCH N1MONTH}

229 REIMONTH {IF @LEFT(MONTH,3)="JAN"}{APPENDBELOW JAN_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="FEB"}{APPENDBELOW FEB_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="MAR"}{APPENDBELOW MAR_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="APR"}{APPENDBELOW APR_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="MAY"}{APPENDBELOW MAY_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="JUN"}{APPENDBELOW JUN_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="JUL"}{APPENDBELOW JUL_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="AUG"}{APPENDBELOW AUG_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="SEP"}{APPENDBELOW SEP_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="OCT"}{APPENDBELOW OCT_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="NOV"}{APPENDBELOW NOV_REIMB,INPUT_AREA}{RETURN}
             {IF @LEFT(MONTH,3)="DEC"}{APPENDBELOW DEC_REIMB,INPUT_AREA}{RETURN}
             {GOTO}MONTH~
             {BEEP}{GETLABEL "Error:  month not correctly specified.  Press [ENTER] to continue.",CHOICE}
             {?}~{TRANS}{BRANCH REIMONTH}
```

```
245 VALID     (LET SEG_CRI,SFC_SEG)(IF SFC_SEG=@DGET(AUTH_SEG,"SEG",CRIT_SEG))(RETURN)
              (GOTO)SFC_SEG~(SEG)(RETURN)

248 SEG       (BEEP)(GETLABEL "Error:  Invalid SEG code.  Try again?  (y/n)  ",CHOICE)
              (IF CHOICE="N")(BRANCH JOURNAL)
              (?)~(TRANS)(BRANCH VALID)

252 CODES     (FRAMEOFF)(WINDOWSOFF)(PANELOFF)(FC)(PGDN 2)(BLANK INPUT_SEG)(PANELON)
              (WINDOWSON)(FORM ENTER_CODES,CODEKEYS)

255 CODEKEYS  ~                          ~(GETLABEL "Verify correct SEG code.  Accept?  (y/n)  ",CHOICE)(BRANCH GOODSEG)
              (END)                      (RESTART)(BRANCH \0)
              (EDIT)                     (ESC)(MENUBRANCH EDIT_SEG)(PANELON)(WINDOWSON)

259 GOODSEG   (IF @CELL("TYPE",INPUT_SEG)="b")(BLK_SEG)(RESTART)(BRANCH CODES)
              (IF CHOICE="Y")(APPENDBELOW AUTH_SEG,INPUT_SEG)
              (RESTART)(BRANCH CODES)

263 EDIT_SEG  Modify                     Delete                     Browse                     Erase
              Modify an existing  ►      Delete a specified  ►      View list of authorized  ► Erase existing list of  ►
              SEG code.                  SEG code.                  SEG codes.                 authorized SEG codes.
265 REMOD     (GETLABEL "SEG code to  ►  (GETLABEL "SEG code to  ►  (PANELOFF)(GOTO)AUTH_SEG~  (WINDOWSOFF)(PANELOFF)
              modify?  ",SEG_CRI)        delete?  ",SEG_CRI)        (GETLABEL "Press B ↵ to  ► (GETLABEL "Erase existing ►
              (PANELOFF)  ►              (WINDOWSOFF)               begin browsing.  Press  ►   list of authorized SEG  ►
              (ONERROR MOD_ERR,ERR_MSG) ► (PANELOFF)/DQR            [ENTER] to return to  ►    codes?  (y/n)  ",CHOICE)
              /DQR                       !AUTH_SEG~                 screen.",CHOICE)           (IF CHOICE="Y")(GOTO)  ►
              !AUTH_SEG~                 CCRIT_SEG~Q                (?)(RESTART)(BRANCH CODES) AUTH_SEG~/RNCAUTH_SEG~ ►
              CCRIT_SEG~                 (LET INPUT_SEG,SEG_CRI)  ►                            (ESC)~(D)/RE(END)(D)~
              OOUT_SEG~MEQ               (WINDOWSON)                                           (RESTART)(BRANCH CODES)
              (IF @CELL("TYPE",NEW_SEG) ► (GETLABEL "Are you sure  ►
              ="b")(BRANCH MOD_ERR)      you want to delete this  ►
              (IF @ROWS(OUT_SEG)>2)  ►   SEG code?  (y/n)  ",  ►
              (WINDOWSOFF)(DUP_MSG)      CHOICE)(IF CHOICE="N")  ►
              (LET INPUT_SEG,NEW_SEG)    (RESTART)(BRANCH CODES)
              (GETLABEL "Change SEG  ►   /DQDDQ(RESTART)  ►
              code to:  ",NEW_SEG)       (WINDOWSOFF)(BRANCH CODES)
              (IF @LENGTH(NEW_SEG)=0)  ►
              (BLK_SEG)/DQMCQ(SEG_RANG) ►
              (RESTART)(BRANCH CODES)
              (LET INPUT_SEG,NEW_SEG)  ►
              (WINDOWSOFF)
              (GETLABEL "Verify correct ►
               SEG code.  Accept?    ►
              (y/n)  ",CHOICE)  ►
              (IF CHOICE="N")/DQMCQ  ►
              (SEG_RANG)  ►
              (RESTART)(BRANCH CODES)
              /DQMRQ(SEG_RANG)  ►
```

```
          {RESTART}{BRANCH CODES}

279 MOD_ERR   {BEEP}{GETLABEL "SEG code not found.  Try again?  (y/n)  ",CHOICE}
          {IF CHOICE="Y"}{BLANK INPUT_SEG}{WINDOWSOFF}{PANELOFF}/DQMCQ{SFG_RANG}{BRANCH REMOD}
          {WINDOWSOFF}{PANELOFF}/DQMCQ{SEG_RANG}{RESTART}{BRANCH CODES}

283 DUP_MSG   {BEEP}{GETLABEL "Warning:  not a unique segment code.  Press [ENTER] to continue.",CHOICE}
          {GETLABEL "Do you want to view extracted records?  (y/n)  ",CHOICE}{IF CHOICE="N"}{WINDOWSON}{RETURN}
          {GOTO}OUT_SEG~{WINDOWSON}{GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to return to input screen.  ",CHOICE}
          {?}{WINDOWSOFF}{FC}{PGDN 2}{WINDOWSON}{RETURN}

288 SEG_RANG  /RNCOUT_SEG~{ESC}~{RETURN}

290 BLK_SEG   {BEEP}{GETLABEL "Error:  this field requires an entry.  Press [ENTER] to continue.",CHOICE}{RETURN}

292 LEDGER    {FRAMEOFF}{BLANK LED_QTR}{BLANK LED_SFC}{BLANK LED_APF}
          {WINDOWSOFF}{PANELOFF}{FC}{PGDN 3}
          {WINDOWSON}{PANELON}{FORM LED_FORM,LED_KEYS}

296 LED_KEYS  ~                         {D}
          {INS}                       {ESC}{ESC}{WINDOWSOFF}{PANELOFF}{PICK_LED}{RESTART}{BRANCH LEDGER}
          {END}                       {RESTART}{BRANCH \0}
          {EDIT}                      {ESC}{ESC}{MENUCALL EDIT_LED}

301 CHK_NUM   {IF @CELL("TYPE",QTR1)<>"v"}{WINDOWSON}{PANELOFF}{GOTO}QTR1~{BRANCH NUM}
          {IF @CELL("TYPE",QTR2)<>"v"}{WINDOWSON}{PANELOFF}{GOTO}QTR2~{BRANCH NUM}
          {IF @CELL("TYPE",QTR3)<>"v"}{WINDOWSON}{PANELOFF}{GOTO}QTR3~{BRANCH NUM}
          {IF @CELL("TYPE",QTR4)<>"v"}{WINDOWSON}{PANELOFF}{GOTO}QTR4~{BRANCH NUM}
          {IF @CELL("TYPE",APF)<>"v"}{WINDOWSON}{PANELOFF}{GOTO}APF~{BRANCH NUM}
          {RETURN}

308 APF_NUM   {IF @CELL("TYPE",APF)<>"v"}{WINDOWSON}{PANELOFF}{GOTO}APF~{BRANCH ADD_APF}{RETURN}

310 PICK_LED  {IF RETRN="MOD1_LED"#AND#@CELL("TYPE",RETRN)<>"b"}{BRANCH LED_RPL}
          {IF LED_SFC="LA"}{LET ROWS,@ROWS(PHONE_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW PHONE_APF,LED_APF} ▸
          {APPENDBELOW PHONE_QTR,LED_QTR}/RNCPHONE_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="L7"}{LET ROWS,@ROWS(TRANS_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW TRANS_APF,LED_APF} ▸
          {APPENDBELOW TRANS_QTR,LED_QTR}/RNCTRANS_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="M1"}{LET ROWS,@ROWS(MAINT_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW MAINT_APF,LED_APF} ▸
          {APPENDBELOW MAINT_QTR,LED_QTR}/RNCMAINT_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="N1"}{LET ROWS,@ROWS(UTIL_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW UTIL_APF,LED_APF} ▸
          {APPENDBELOW UTIL_QTR,LED_QTR}/RNCUTIL_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="P1"}{LET ROWS,@ROWS(ENGR_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW ENGR_APF,LED_APF} ▸
          {APPENDBELOW ENGR_QTR,LED_QTR}/RNCENGR_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="R1"}{LET ROWS,@ROWS(CONST_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW CONST_APF,LED_APF} ▸
          {APPENDBELOW CONST_QTR,LED_QTR}/RNCCONST_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="S1"}{LET ROWS,@ROWS(SUPPT_QTR)}{CHK_NUM}{EXIST}{APPENDBELOW SUPPT_APF,LED_APF} ▸
          {APPENDBELOW SUPPT_QTR,LED_QTR}/RNCSUPPT_QTR~{ESC}.{D 4}~{RETURN}
          {IF LED_SFC="M2"}{SP_NUM}{M2_SEG}{APF_NUM}{LET OUT_APF_B,LED_APF}{LET OUT_SEG_B,LED_SFC} ▸
```

```
        {APPENDBELOW SP_RPR_LED,IN_LED}{RETURN}
        {IF LED_SFC="R2"}{SP_NUM}{R2_SEG}{APF_NUM}{LET OUT_APF_B,LED_APF}{LET OUT_SEG_B,LED_SFC} ▸
        {APPENDBELOW SP_CONST_LED,IN_LED}{RETURN}
        {RMB_SEG}{CHK_SEG}{APF_NUM}{LET OUT_APF_B,LED_APF}{LET OUT_SEG_B,LED_SFC} ▸
        {APPENDBELOW REIMB_LED,IN_LED}{RETURN}

322 NUM {BEEP}{GETLABEL "Error:  this field requires a numerical entry.  Press [ENTER] to continue.",CHOICE}
        {PANELON}{?}~{WINDOWSON}{BRANCH CHK_NUM}

325 ADD_APF {BEEP}{GETLABEL "Error:  this field requires a numerical entry.  Press [ENTER] to continue.",CHOICE}
        {PANELON}{?}~{WINDOWSON}{BRANCH APF_NUM}

328 CHK_SEG {IF LED_SFC=@DGET(AUTH_SEG,"SEG",CRIT_SEG)}{RETURN}
        {BAD_SEG}{RETURN}

331 EXIST {IF ROWS=1}{RETURN}
        {GETLABEL "Ledger already exists.  Select Modify or Delete.  Press [ENTER] to continue.",CHOICE}{RESTART}{BRANCH LEDGER}

334 SP_NUM {GETLABEL "Enter special project number and press [ENTER]  ",LED_SFC}
        {RETURN}

337 M2_SEG {LET SEG_CRI,LED_SFC}{IF LED_SFC=@DGET(SP_RPR_LED,"SEG",CRIT_SEG)}{DUP_SEG}
        {RETURN}

340 R2_SEG {LET SEG_CRI,LED_SFC}{IF LED_SFC=@DGET(SP_CONST_LED,"SEG",CRIT_SEG)}{DUP_SEG}
        {RETURN}

343 RMB_SEG {LET SEG_CRI,LED_SFC}{IF LED_SFC=@DGET(REIMB_LED,"SEG",CRIT_SEG)}{DUP_SEG}
        {RETURN}

346 DUP_SEG {BEEP}{GETLABEL "Entry already exists.  Press [ENTER] then Browse, Delete, or try again.",CHOICE}
        {RESTART}{BRANCH LEDGER}

349 BAD_SEG {BEEP}{GETLABEL "Invalid SEG code.  Try again?  (y/n)  ",CHOICE}
        {IF CHOICE="N"}{RESTART}{BRANCH LEDGER}
        {GETLABEL "Enter new SEG code.  ",LED_SFC}{RMB_SEG}{BRANCH CHK_SEG}

353 EDIT_LED Modify                          Delete                          Browse                          Erase
        Modify an existing ▸        Delete a specified ▸        View a specified ledger.    Erase all ledger entries ▸
        ledger entry.               ledger entry.               {WINDOWSOFF}{PANELOFF}      in all ledgers.
        {WINDOWSOFF}{PANELOFF}      {WINDOWSOFF}{PANELOFF}      {GETLABEL "Enter SFC or ▸   {GETLABEL "Are you sure ▸
        {LET RETRN,"MODI_LED"}      {GETLABEL "Enter SFC or ▸   press [ENTER] for ▸         you want to erase all ▸
357 RELED {GETLABEL "Enter SFC or ▸ press [ENTER] for ▸         reimbursables.  ",CHOICE}   ledgers?  (y/n)  ", ▸
        press [ENTER] for ▸         reimbursables.  ",CHOICE}   {LET RETRN,"BROW_LED"}      CHOICE}{IF CHOICE="N"} ▸
        reimbursables.  ",CHOICE} ▸ {LET RETRN,"DELE_LED"}      {GET_LED}                   {RESTART}{BRANCH LEDGER}
        {LET LED_SFC,CHOICE}        {GET_LED}                   {BLANK RETRN}               {WINDOWSOFF}{PANELOFF} ▸
        {IF CHOICE<>"LA"#AND# ▸     {BLANK RETRN}               {GOTO}                      {INDICATE "WAIT"}
        CHOICE<>"L7"#AND#CHOICE<> ▸ {IF @RIGHT(CHOICE,3)<> ▸   @S(LED_LOC)                {LET ROWS,@ROWS(PHONE_QTR)} ▸
        "M1"#AND#CHOICE<>"N1" ▸     "QTR"}{BRANCH LED_DEL}      ~{U 3}{IF LED_LOC= ▸        {GOTO}PHONE_APF~{D}/RE{END} ▸
```

```
                #AND#CHOICE<>"P1"#AND# ▸    (GETLABEL "Are you sure  ▸ "PHONE_APF"#OR#LED_LOC= ▸   (D)~(R)/RE(END)(D)~ ▸
                CHOICE<>"R1"#AND#CHOICE<> ▸ you want to delete ledger ▸ "UTIL_APF"#OR#LED_LOC= ▸   (IF ROWS<>1)/RNCPHONE_APF~ ▸
                "S1")(GETLABEL "Enter  ▸     entries?  (y/n)  ", ▸   "REIMB_LED")(SET_CALC)      (ESC)~/RNCPHONE_QTR~(ESC)~
                SEG or special project  ▸   RETRN)(IF RETRN="N") ▸   (IF @LEFT(LED_LOC,3)= ▸     (LET ROWS,@ROWS(TRANS_QTR)) ▸
                number.  ",SEG_CRI)          (BLANK RETRN) ▸         "SP_"#OR#LED_LOC= ▸       (GOTO)TRANS_APF~(D)/RE(END) ▸
                (GET_LED)                    (RESTART)(BRANCH LEDGER)  "REIMB_LED") ▸          (D)~(R)/RE(END)(D)~ ▸
                (IF ROWS=1)(BRANCH NO_LED)   (GOTO)                    (BRANCH TITL_BRS)       (IF ROWS<>1)/RNCTRANS_APF~ ▸
                (IF @RIGHT(LED_LOC,3)<> ▸    @S(LED_LOC)             (WINDOWSON)           (ESC)~/RNCTRANS_QTR~(ESC)~
                "APF")(BRANCH LED_MOD)       ~(D)/RE(END)(D)~         (GETLABEL "Press [ENTER]  ▸ (LET ROWS,@ROWS(MAINT_QTR)) ▸
                (GOTO)                       (R)/RE(END)(D)~          to return to input  ▸     (GOTO)MAINT_APF~(D)/RE(END) ▸
                @S(LED_LOC)                 (CALC)(IF ROWS=1) ▸       screen.",CHOICE)        (D)~(R)/RE(END)(D)~ ▸
                ~/DQRI                       (RESTART)(BRANCH LEDGER)  (WINDOWSOFF)(IF LED_LOC= ▸ (IF ROWS<>1)/RNCMAINT_APF~ ▸
                .(R)(D 4)~                   /RNC                      "PHONE_APF"#OR#LED_LOC= ▸   (ESC)~/RNCMAINT_QTR~(ESC)~
                CCRIT_LED~                   @S(LED_LOC)             "UTIL_APF"#OR#LED_LOC= ▸  (LET ROWS,@ROWS(UTIL_QTR)) ▸
                OOUT_LED_A~MEQ               ~(ESC)~/RNC               "REIMB_LED")(UNCALC)      (GOTO)UTIL_APF~(D)/RE(END) ▸
368 TITL_BRS    (LET LED_APF,OUT_APF) ▸      @S(CHOICE)              (RESTART)(INDICATE) ▸     (D)~(R)/RE(END)(D)~ ▸
                /COUT_QTR~LED_QTR~           ~(ESC)~                   (BRANCH LEDGER)         (IF ROWS<>1)/RNCUTIL_APF~ ▸
                (FC)(PGDN 3) ▸               (RESTART)(BRANCH LEDGER)  (D 4)/WTH(WINDOWSON)      (ESC)~/RNCUTIL_QTR~(ESC)~
                (WINDOWSON)(PANELON)                                   (GETLABEL "Press B ↵ to ▸ (LET ROWS,@ROWS(ENGR_QTR)) ▸
                (GETLABEL "Press [INS]  ▸                              begin browsing.  Press  ▸ (GOTO)ENGR_APF~(D)/RE(END) ▸
                when done, [END] to stop. ▸                            [ENTER] to return to  ▸   (D)~(R)/RE(END)(D)~ ▸
                  Press [ENTER] to  ▸                                  input screen.  ",CHOICE)  (IF ROWS<>1)/RNCENGR_APF~ ▸
                continue.",CHOICE)                                     (?)(WINDOWSOFF) ▸         (ESC)~/RNCENGR_QTR~(ESC)~
                (FORM LED_FORM,MDA_KEYS)                               (IF LED_LOC="PHONE_APF" ▸ (LET ROWS,@ROWS(CONST_QTR)) ▸
372 LED_RPL     (IF @RIGHT(LED_LOC,3)<> ▸                              #OR#LED_LOC="UTIL_APF" ▸  (GOTO)CONST_APF~(D)/RE(END) ▸
                "APF")(BRANCH LED_REP)                                 #OR#LED_LOC="REIMB_LED") ▸ (D)~(R)/RE(END)(D)~ ▸
                (WINDOWSOFF)(PANELOFF) ▸                               (UNCALC)                  (IF ROWS<>1)/RNCCONST_APF~ ▸
                (CHK_NUM)(WINDOWSOFF) ▸                                /WTC(RESTART)(INDICATE) ▸ (ESC)~/RNCCONST_QTR~(ESC)~
                (PANELOFF)                                             (BRANCH LEDGER)           (LET ROWS,@ROWS(SUPPT_QTR)) ▸
                (LET OUT_APF,LED_APF) ▸                                                          (GOTO)SUPPT_APF~(D)/RE(END) ▸
                /CLED_QTR~OUT_QTR~                                                               (D)~(R)/RE(END)(D)~ ▸
                /DQMRQ(RANG_A)                                                                   (IF ROWS<>1)/RNCSUPPT_APF~ ▸
                (BLANK RETRN) ▸                                                                  (ESC)~/RNCSUPPT_QTR~(ESC)~
                (RESTART)(BRANCH LEDGER)                                                         (LET ROWS,@ROWS(SP_RPR_LED)) ▸
                                                                                                 (GOTO)SP_RPR_SEG~(D)/RE(END) ▸
                                                                                                 (D)~(R)/RE(END)(D)~ ▸
                                                                                                 (IF ROWS<>1) ▸
                                                                                                 /RNCSP_RPR_LED~(ESC).(R)~
                                                                                                 (LET ROWS,@ROWS ▸
                                                                                                 (SP_CONST_LED))(GOTO) ▸
                                                                                                 SP_CONST_SEG~(D)/RE(END) ▸
                                                                                                 (D)~(R)/RE(END)(D)~ ▸
                                                                                                 (IF ROWS<>1) ▸
                                                                                                 /RNCSP_CONST_LED~(ESC).(R)~
                                                                                                 (LET ROWS,@ROWS(REIMB_LED)) ▸
                                                                                                 (GOTO)REIMB_SEG~(D)/RE(END) ▸
                                                                                                 (D)~(R)/RE(END)(D)~ ▸
                                                                                                 (IF ROWS<>1) ▸
                                                                                                 /RNCREIMB_LED~(ESC).(R)~
```

```
                                                                                {INDICATE "READY"}{RESTART} ▸
                                                                                {INDICATE}{BRANCH LEDGER}

378 MDA_KEYS  ~                          {D}
              {INS}                      {ESC}{ESC}{WINDOWSOFF}{PANELOFF}{CHK_NUM}{PICK_LED}{RESTART}{BRANCH LEDGER}
              {END}                      {ESC}{END_MDA}

382 END_MDA   {GETLABEL "Cancel modification and return to entry screen?  (y/n)  ",CHOICE}
              {IF CHOICE="Y"}{WINDOWSOFF}{PANELOFF}/DQMCQ{RANG_A}{BLANK RETRN}{RESTART}{BRANCH LEDGER}
              {RETURN}

386 NO_LED    {GETLABEL "No entries found in this ledger.  Press [ENTER], then make a new entry.",CHOICE}
              {BLANK RETRN}{RESTART}{BRANCH LEDGER}

389 ROWS

391 LED_LOC

393 SET_CALC  {INDICATE "UPDATING LEDGER"}{D 4}{R 2}{EDIT}{HOME}{DEL}~
              {R}{EDIT}{HOME}{DEL}~{L 3}{U 4}{INDICATE "READY"}{RETURN}

396 UNCALC    {GOTO}
              @S(LED_LOC)
              ~{D}{R 2}{EDIT}{HOME}'~
              {R}{EDIT}{HOME}'~{RETURN}

401 GET_LED   {IF CHOICE="LA"}{LET LED_LOC,"PHONE_APF"}{LET CHOICE,"PHONE_QTR"}{LET ROWS,@ROWS(PHONE_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="L7"}{LET LED_LOC,"TRANS_APF"}{LET CHOICE,"TRANS_QTR"}{LET ROWS,@ROWS(TRANS_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="M1"}{LET LED_LOC,"MAINT_APF"}{LET CHOICE,"MAINT_QTR"}{LET ROWS,@ROWS(MAINT_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="N1"}{LET LED_LOC,"UTIL_APF"}{LET CHOICE,"UTIL_QTR"}{LET ROWS,@ROWS(UTIL_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="P1"}{LET LED_LOC,"ENGR_APF"}{LET CHOICE,"ENGR_QTR"}{LET ROWS,@ROWS(ENGR_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="R1"}{LET LED_LOC,"CONST_APF"}{LET CHOICE,"CONST_QTR"}{LET ROWS,@ROWS(CONST_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="S1"}{LET LED_LOC,"SUPPT_APF"}{LET CHOICE,"SUPPT_QTR"}{LET ROWS,@ROWS(SUPPT_QTR)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="M2"}{LET LED_LOC,"SP_RPR_LED"}{LET ROWS,@ROWS(SP_RPR_LED)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {IF CHOICE="R2"}{LET LED_LOC,"SP_CONST_LED"}{LET ROWS,@ROWS(SP_CONST_LED)}{CALC}{BRANCH @LEFT(RETRN,8)}
              {LET LED_LOC,"REIMB_LED"}{LET ROWS,@ROWS(REIMB_LED)}{CALC}{BRANCH @LEFT(RETRN,8)}

412 LED_DEL   {MENUCALL PICK_DEL}{RESTART}{BRANCH LEDGER}

414 PICK_DEL  Specified                  All
              Delete a specified ledger enDelete all ledger entries from ledger.
              {IF @LEFT(LED_LOC,3)="SP_"}{{GOTO}
              {IF @LEFT(LED_LOC,3)="SP_"}{@S(LED_LOC)
              {IF LED_LOC="REIMB_LED"}{GET~{D}/RE{END}{D}~
              {IF LED_LOC="REIMB_LED"}{GET{R}/RE{END}{D}~
              ~/DQRI                     {CALC}{IF ROWS=1}{RESTART}{BRANCH LEDGER}
              @S(LED_LOC)               /RNC
              ~CCRIT_SEG~DDQ             @S(LED_LOC)
              {RETURN}                   ~{ESC}.{R}~
```

```
                                    {RETURN}

426 LED_MOD     /DQRI
                @S(LED_LOC)
                ~CCRIT_SEG~
                OOUT_LED_B~MEQ
                {LET LED_APF,OUT_APF_B}{LET LED_SFC,OUT_SEG_B}
                {ONERROR LED_ERR}{IF @CELL("TYPE",OUT_SEG_B)="b"}{BRANCH LED_ERR}
                {WINDOWSON}{PANELON}
                {GETLABEL "Press [INS] when done, [END] to stop.  Press [ENTER] to continue.",CHOICE}
434 LED_REP     {FORM LED_FORM,MDB_KEYS}
                {WINDOWSOFF}{PANELOFF}{APF_NUM}{WINDOWSOFF}{PANELOFF}
                {LET OUT_APF_B,LED_APF}{LET OUT_SEG_B,LED_SFC}
                /DQMRQ{RANG_B}
                {BLANK RETRN}{RESTART}{BRANCH LEDGER}

440 LED_ERR     {GETLABEL "Ledger entry not found.  Try again?  (y/n)  ",CHOICE}
                {IF CHOICE="N"}/DQMCQ{RANG_B}{RESTART}{BRANCH LEDGER}
                /DQMCQ{RANG_B}{BRANCH RELED}

444 RANG_A      /RNCOUT_LED_A~{ESC}.{R}~{RETURN}

446 RANG_B      /RNCOUT_LED_B~{ESC}.{R}~{RETURN}

448 MDB_KEYS    ~                           {D}
                {INS}                       {ESC}{ESC}{WINDOWSOFF}{PANELOFF}{PICK_LED}{RESTART}{BRANCH LEDGER}
                {END}                       {ESC}{END_MDB}

452 END_MDB     {GETLABEL "Cancel modification and return to entry screen?  (y/n)  ",CHOICE}
                {IF CHOICE="Y"}{WINDOWSOFF}{PANELOFF}/DQMCQ{RANG_B}{BLANK RETRN}{RESTART}{BRANCH LEDGER}
                {RETURN}

456 WRK         Scratch                     Other                       Erase
                Exit the accounting  ►      Journal  Ledger   ►         Erase scratch pad.
                system and go to the  ►     Special  Reimbursables      {GETLABEL "Are you sure  ►
                scratch pad.                {MENUCALL REPORT}           you want to erase the  ►
                {GETLABEL "Press ALT-A  ►   {QUIT}                      scratch pad?  (y/n)  " ►
                to return to the  ►                                     ,CHOICE}
                accounting system.   ►                                  {IF CHOICE="N"} ►
                Press [ENTER] to  ►                                     {RESTART}{BRANCH \0}
                continue.",CHOICE} ►                                    {PANELOFF}{WINDOWSOFF} ►
                {PANELOFF}{GOTO}SCRATCH~ ►                              {GOTO}SCRATCH~
                /WCS12~/RFD1~                                           /RE{END}{HOME}~
                {PANELON}{QUIT}                                         {GOTO}RPT_LED~ ►
                                                                        /RE{END}{HOME}~
                                                                        {RESTART}{BRANCH \0}

465 REPORT      Journal                     Ledger                      Special
                Select a journal to  ►      Use public works ledgers  ► Use special projects  ►
```

```
        create a report.            to create a report.         ledgers to create a report. Reimbursables
467 REWRK  (GETLABEL "Enter SFC or  ► (GOTO)RPT_LED~/WGC14~/RFD1~ (GOTO)RPT_LED~/WGC14~/RFD1~ /RE(END)(HOME)~(D 3) ►
           Press [ENTER] for        ► /RE(END)(HOME)~(D 3) ►    /RE(END)(HOME)~(D 3) ►      /RNDTEMP~/RNCTEMP~~
           reimbursables. ",SFC)       /RNDTEMP~/RNCTEMP~~      /RNDTEMP~/RNCTEMP~~         (LET RPT_LED,@TODAY) ►
           (IF SFC<>"M1"#AND#SFC<> ►   (LET RPT_LED,@TODAY) ►   (LET RPT_LED,@TODAY) ►     (LET LED_TITL, ►
           "P1"#AND#SFC<>"L7"#AND# ►   (LET LED_TITL,"PUBLIC ►  (LET LED_TITL,"SPEC AL ►   "REIMBURSABLES LEDGER ►
           SFC<>"R1"#AND#SFC<>"S1" ►   WORKS LEDGER REPORT")    PROJECTS LEDGER REPORT")    REPORT")
           #AND#SFC<>"M2"#AND#SFC<> ►  (GOTO)PHONE_APF~(D) ►    (GOTO)SP_RPR_LED~(D) ►      (GOTO)REIMB_LED~(D) ►
           "R2")(GETLABEL "Enter  ►    (IF @CELLPOINTER("TYPE") ► (IF @CELLPOINTER("TYPE") ► (IF @CELLPOINTER("TYPE") ►
           month.  ",MTH)              <>"b")(U 4)(SET_CALC) ►  <>"b")(U 4)(LED_RANG)       <>"b")(U 4)(SET_CALC) ►
           (IF @LEFT(MTH,3)<>"JAN" ►   (LED_RANG)(LET LED_LOC, ► (GOTO)SP_CONST_LED~(D) ►  (LED_RANG)(LET LED_LOC, ►
           #AND#@LEFT(MTH,3)<>"FEB" ►  "PHONE_APF")(CALC)(UNCALC) (IF @CELLPOINTER("TYPE") ► "REIMB_LED")(CALC)(UNCALC)
           #AND#@LEFT(MTH,3)<>"MAR" ►  (GOTO)TRANS_APF~(D) ►    <>"b")(U 4)(LED_RANG)       (GOTO)RPT_LED~(PANELON) ►
           #AND#@LEFT(MTH,3)<>"APR" ►  (IF @CELLPOINTER("TYPE") ► (GOTO)RPT_LED~(PANELON) ► (WINDOWSON)(RETURN)
           #AND#@LEFT(MTH,3)<>"MAY" ►  <>"b")(U 4)(LED_RANG)    (WINDOWSON)(RETURN)
           #AND#@LEFT(MTH,3)<>"JUN" ►  (GOTO)MAINT_APF~(D) ►
           #AND#@LEFT(MTH,3)<>"JUL" ►  (IF @CELLPOINTER("TYPE") ►
           #AND#@LEFT(MTH,3)<>"AUG" ►  <>"b")(U 4)(LED_RANG)
           #AND#@LEFT(MTH,3)<>"SEP" ►  (GOTO)UTIL_APF~(D) ►
           #AND#@LEFT(MTH,3)<>"OCT" ►  (IF @CELLPOINTER("TYPE") ►
           #AND#@LEFT(MTH,3)<>"NOV" ►  <>"b")(U 4)(SET_CALC) ►
           #AND#@LEFT(MTH,3)<>"DEC") ► (LED_RANG)(LET LED_LOC, ►
           (BRANCH FIX_MTH)            "UTIL_APF")(CALC)(UNCALC)
           (PICK_DB)                   (GOTO)ENGR_APF~(D) ►
           (GETLABEL "Press ALT-A  ►   (IF @CELLPOINTER("TYPE") ►
           to return to the  ►         <>"b")(U 4)(LED_RANG)
           accounting system.   ►      (GOTO)CONST_APF~(D) ►
           Press [ENTER] to  ►         (IF @CELLPOINTER("TYPE") ►
           continue.",CHOICE) ►        <>"b")(U 4)(LED_RANG)
           (WINDOWSOFF)(PANELOFF) ►    (GOTO)SUPPT_APF~(D) ►
           (GOTO)SCRATCH~/C            (IF @CELLPOINTER("TYPE") ►
           @S(PLACE2)                 <>"b")(U 4)(LED_RANG)
           ~PAD~                       (GOTO)RPT_LED~(PANELON) ►
           (GOTO)SCRATCH~/WCS12~/RFD1~ (WINDOWSON)(RETURN)
           (WINDOWSON)(PANELON) ►
           (RETURN)

479 FIX_MTH (BEEP)(GETLABEL "Month not correctly specified.  Try again? (y/n)  ",CHOICE)
           (IF CHOICE="N")(RESTART)(BRANCH \0)
           (BRANCH REWRK)

483 LED_RANG /RV(R)(END)(D)(END)(D)(R 3)~TEMP~
           (GOTO)TEMP~(R)(END)(D)(END)(D)(L)(D 4)/RNDTEMP~/RNCTEMP~~
```

RANGE NAME	RANGE ADDRESS
ADD_APF	AX:B325..AX:B325
APF	A:L70..A:L70
APF_NUM	AX:B308..AX:B308
APR_PHONE	I:A4..I:O4
APR_REIMB	AL:A4..AL:O4
APR_UTIL	W:A4..W:O4
AUG_PHONE	M:A4..M:O4
AUG_REIMB	AP:A4..AP:O4
AUG_UTIL	AA:A4..AA:O4
AUTH_SEG	D:AA1..D:AA1
BAD_MTH	AX:B80..AX:B80
BAD_SEG	AX:B349..AX:B349
BLANK_CHK	D:A2..D:A2
BLK_SEG	AX:B290..AX:B290
BROW	AX:D29..AX:D29
BROW_LED	AX:D359..AX:D359
CA	A:H9..A:H9
CHAR	AX:B172..AX:B172
CHK_NUM	AX:B301..AX:B301
CHK_SEG	AX:B328..AX:B328
CHK_SP	AX:B193..AX:B193
CHOICE	AX:B149..AX:B149
CODEKEYS	AX:B255..AX:C257
CODES	AX:B252..AX:B252
COMMITTED	A:L9..A:L9
CONFIRM	AX:B145..AX:B145
CONST_APF	AW:Z4..AW:Z4
CONST_DB	AG:A3..AG:O3
CONST_QTR	AW:AA4..AW:AA4
CONTRACTOR	A:D14..A:D14
CONT_CRIT	E:F1..E:F2
CRIT_JRN	E:A7..E:B8
CRIT_LED	E:A12..E:A13
CRIT_SEG	E:A1..E:A2
DB_SEL	A:B16..A:B16
DEC_PHONE	Q:A4..Q:O4
DEC_REIMB	AT:A4..AT:O4
DEC_UTIL	AE:A4..AE:O4
DELE_LED	AX:C359..AX:C359
DESCRIPTION	A:D13..A:D13
DUP_MSG	AX:B283..AX:B283
DUP_SEG	AX:B346..AX:B346
DUP_WARN	AX:B53..AX:B53
EDIT_JRN	AX:B25..AX:B25
EDIT_LED	AX:B353..AX:B353
EDIT_SEG	AX:B263..AX:B263
EMPTY	AX:B169..AX:B169

RANGE NAME	RANGE ADDRESS
END_MDA	AX:B382..AX:B382
END_MDB	AX:B452..AX:B452
END_MOD	AX:B65..AX:B65
END_VIEW	AX:B142..AX:B142
ENGR_APF	AW:U4..AW:U4
ENGR_DB	AF:A3..AF:O3
ENGR_QTR	AW:V4..AW:V4
ENTER_CODES	A:A41..A:M60
ENTER_JON	A:D12..A:D12
ENTRYFORM	A:A1..A:M20
ENTRY_SCREEN	A:A21..A:M40
ERAS	AX:B94..AX:B94
ERAS_JRN	AX:B88..AX:B88
ERAS_MTH	AX:B105..AX:B105
ERR_MSG	AX:B23..AX:B23
EVAL	AX:B181..AX:B181
EXIST	AX:B331..AX:B331
EXPENDED	A:L11..A:L11
FEB_PHONE	G:A4..G:O4
FEB_REIMB	AJ:A4..AJ:O4
FEB_UTIL	U:A4..U:O4
FIN_MOD	AX:B48..AX:B48
FIX_MC	AX:B178..AX:B178
FIX_MTH	AX:B479..AY:B479
GET_DB	AX:B121..AX:B121
GET_LED	AX:B401..AX:B401
GET_SEG	AX:B401..AX:B401
GOODSEG	AX:B259..AX:B259
INPUT_A	A:D9..A:D14
INPUT_AREA	D:A2..D:O2
INPUT_B	A:H9..A:H12
INPUT_C	A:L9..A:L13
INPUT_JON	A:D12..A:D12
INPUT_ODC	A:D9..A:D9
INPUT_SEG	A:F50..A:F50
INPUT_SFC	A:H10..A:H10
IN_LED	D:U2..D:V2
JAN_PHONE	F:A4..F:O4
JAN_REIMB	AI:A4..AI:O4
JAN_UTIL	T:A4..T:O4
JOURNAL	AX:B15..AX:B15
JRN_ERR	AX:B84..AX:B84
JUL_PHONE	L:A4..L:O4
JUL_REIMB	AO:A4..AO:O4
JUL_UTIL	Z:A4..Z:O4
JUN_PHONE	K:A4..K:O4
JUN_REIMB	AN:A4..AN:O4

RANGE NAME	RANGE ADDRESS
JUN_UTIL	Y:A4..Y:O4
LAMONTH	AX:B197..AX:B197
LEDGER	AX:B292..AX:B292
LED_APF	A:L70..A:L70
LED_DEL	AX:B412..AX:B412
LED_ERR	AX:B440..AX:B440
LED_FORM	A:A61..A:M80
LED_KEYS	AX:B296..AX:C299
LED_LOC	AX:B391..AX:B391
LED_MOD	AX:B426..AX:B426
LED_QTR	A:D69..A:D72
LED_RANG	AX:B483..AX:B483
LED_REP	AX:B435..AX:B435
LED_RPL	AX:B372..AX:B372
LED_SFC	A:H70..A:H70
LED_TITL	C:C1..C:C1
LOC	AX:B135..AX:B135
M2_SEG	AX:B337..AX:B337
MACROS	AX:A1..AX:A1
MAINT_APF	AW:K4..AW:K4
MAINT_DB	S:A3..S:O3
MAINT_QTR	AW:L4..AW:L4
MAR_PHONE	H:A4..H:O4
MAR_REIMB	AK:A4..AK:O4
MAR_UTIL	V:A4..V:O4
MATL_CRIT	E:E1..E:E2
MAY_PHONE	J:A4..J:O4
MAY_REIMB	AM:A4..AM:O4
MAY_UTIL	X:A4..X:O4
MC	A:L13..A:L13
MDA_KEYS	AX:B378..AX:C380
MDB_KEYS	AX:B448..AX:C450
MODI_LED	AX:B360..AX:B360
MOD_ERR	AX:B279..AX:B279
MOD_KEYS	AX:B61..AX:C63
MONTH	A:L12..A:L12
MTH	E:C8..E:C8
MTH_ERR	AX:B139 .AX:B139
N1MONTH	AX:B213..AX:B213
NEW_SEG	E:H2..E:H2
NOV_PHONE	P:A4..P:O4
NOV_REIMB	AS:A4..AS:O4
NOV_UTIL	AD:A4..AD:O4
NO_LED	AX:B386..AX:B386
NUM	AX:B322..AX:B322
OBLIGATED	A:L10..A:L10
OCT_PHONE	O:A4..O:O4

RANGE NAME	RANGE ADDRESS
OCT_REIMB	AR:A4..AR:O4
OCT_UTIL	AC:A4..AC:O4
ODC	E:A8..E:A8
OUT_APF	D:S2..D:S2
OUT_APF_B	D:V2..D:V2
OUT_JRN	D:A1..D:O1
OUT_LED_A	D:S1..D:T1
OUT_LED_B	D:U1..D:V1
OUT_QTR	D:T2..D:T5
OUT_SEG	E:H1..E:H1
OUT_SEG_B	D:U2..D:U2
PAD	B:A5..B:A5
PHONE_APF	AW:A4..AW:A4
PHONE_QTR	AW:B4..AW:B4
PICK_DB	AX:B69..AX:B69
PICK_DEL	AX:B414..AX:B414
PICK_LED	AX:B310..AX:B310
PLACE	AX:B136..AX:B136
PLACE2	AX:B35..AX:B35
QTR1	A:D69..A:D69
QTR2	A:D70..A:D70
QTR3	A:D71..A:D71
QTR4	A:D72..A:D72
R2_SEG	AX:B340..AX:B340
RANG_A	AX:B444..AX:B444
RANG_B	AX:B446..AX:B446
REIMB_LED	AW:AT4..AW:AU4
REIMB_SEG	AW:AT4..AW:AT7
REIMONTH	AX:B229..AX:B229
RELED	AX:B357..AX:B357
REMOD	AX:B265..AX:B265
REPORT	AX:B465..AX:B465
RESET	AX:B58..AX:B58
RETRN	AX:B119..AX:B119
RETRY	AX:B28..AX:B28
REWRK	AX:B467..AX:B467
RMB_SEG	AX:B343..AX:B343
ROWS	AX:B389..AX:B389
RPT_LED	C:A1..C:A1
SCRATCH	B:A1..B:A1
SEG	AX:B248..AX:B248
SEG_CRI	E:A2..E:A2
SEG_RANG	AX:B288..AX:B288
SELECT	AX:B4..AX:B4
SEP_PHONE	N:A4..N:O4
SEP_REIMB	AQ:A4..AQ:O4
SEP_UTIL	AB:A4..AB:O4

RANGE NAME	RANGE ADDRESS
SET_CALC	AX:B393..AX:B393
SFC	E:B8..E:B8
SFC_SEG	A:H10..A:H10
SIGKEYS	AX:B18..AX:C21
SP	B:D1..B:D1
SP_CONST_DB	AV:A3..AV:O3
SP_CONST_LED	AW:AO4..AW:AP4
SP_CONST_SEG	AW:AO4..AW:AO4
SP_NUM	AX:B334..AX:B334
SP_RPR_DB	AU:A3..AU:O3
SP_RPR_LED	AW:AJ4..AW:AK4
SP_RPR_QTR	AW:AK4..AW:AK7
SP_RPR_SEG	AW:AJ4..AW:AJ5
SUPPT_APF	AW:AE4..AW:AE4
SUPPT_DB	AH:A3..AH:O3
SUPPT_QTR	AW:AF4..AW:AF4
TEMP	C:A4..C:A4
TEST_A	D:A2..D:A2
TEST_CRIT	AW:AZ1..AW:AZ2
TEST_OUT	AW:BC1..AW:BD1
TITL_BRS	AX:D368..AX:D368
TRANS	AX:B151..AX:B151
TRANS_A	D:A2..D:F2
TRANS_APF	AW:F4..AW:F4
TRANS_B	D:G2..D:J2
TRANS_C	D:K2..D:O2
TRANS_DB	R:A3..R:O3
TRANS_QTR	AW:G4..AW:G4
UNCALC	AX:B396..AX:B396
UTIL_APF	AW:P4..AW:P4
UTIL_QTR	AW:Q4..AW:Q4
VALID	AX:B245..AX:B245
VALUE	AX:B175..AX:B175
VIEW_MTH	AX:B130..AX:B130
WRK	AX:B456..AX:B456
\0	AX:B1..AX:B1
\A	AX:B13..AX:B13

APPENDIX B

Appendix B contains the macros and range names used in the labor system file. The presentation format is the same as for the accounting system file.

H	A	B	C	D	E	F
1	\0	{FRAMEOFF}{WINDOWSOFF}{FC}{PGDN}{WINDOWSON} {MENUBRANCH SELECT}				
4	SELECT	Labor Enter labor card data. {BRANCH LABOR} {BRANCH \0}	Personnel Enter personnel data ► for each employee. {BRANCH PERSON} {BRANCH \0}	Report Create labor cost report. {BRANCH REPORT} {BRANCH \0}	Work Scratch Other Erase {WINDOWSOFF}{PANELOFF} ► /RNCSCRATCH~B:A1~ ► /RNCSP~B:D1~/RNCPAD~B:A5~ {LET SCRATCH,@TODAY} ► {LET SP,LABOR ► SYSTEM SCRATCH PAD} {GOTO}SCRATCH~/WCS10~/RFD1~ {MENUCALL WRK} {BRANCH \0}	Quit End session and return ► to 1-2-3. {GETLABEL "Do you want ► to save your work? ► (y/n) ",CHOICE} ► {IF CHOICE="N"}{QUIT} {PANELOFF}/FS{QUIT}
12	\A	{BRANCH \0}				
14	LABOR	{FRAMEOFF}{BLANK INPUT_A}{BLANK INPUT_B}{BLANK INPUT_C} {WINDOWSON}{PANELON}{FORM ENTER_LAB,LAB_KEYS}				
17	LAB_KEYS	{INS} {END} ~ {EDIT}	{ESC}{ESC}{TRANS}{VALID}{APPENDBELOW LAB_DB,INPUT_LAB}{RESTART}{BRANCH LABOR} {BRANCH CONFIRM} {D} {ESC}{ESC}{MENUCALL EDIT_LAB}{PANELON}{WINDOWSON}			
22	ERR_MSG					
24	EDIT_LAB	Modify Modify an existing ► labor card entry.	Delete Delete a specified ► labor card entry.	Browse View labor card data. {WINDOWSOFF}{PANELOFF} {GOTO}LAB_DB~{HOME} {D 4}/WTH {WINDOWSON} {END_VIEW} {RESTART}{BRANCH LABOR}	Erase Erase all existing ► labor card data. {GETLABEL "Are you sure ► you want to erase all ► existing labor card ► data? (y/n) ",CHOICE} {IF CHOICE="N"}{RETRUN} {WINDOWSOFF}{PANELOFF} {GOTO}LAB_DB~{D} /RE{END}{HOME}~ {U}/RNCLAB_DB~ ► {ESC}.{END}{R}~ {FC}{RESTART}{BRANCH LABOR}	
27	RETRY	{LET RETRN,"MODIFY"} {GETLABEL "Enter social ► security number or press ► [ENTER] to continue. " ► ,SSN_CRI}{IF SSN_CRI=""} ► {BLANK SSN_CRI} {GETLABEL "Enter job ► number or press [ENTER] ► to continue. ",JOB_CRI} ► {IF JOB_CRI=""} ► {BLANK JOB_CRI} {GETLABEL "Enter pay ► period or press [ENTER] ► to continue. ",PRD_CRI} ► {IF PRD_CRI=""} ► {BLANK PRD_CRI}	{LET RETRN,"DELETE"} {RETRY} {GETLABEL "Are you sure ► you want to delete this ► labor card entry? ► (y/n) ",CHOICE} {IF CHOICE="N"}{PANELOFF} ► /DQMCQ{RESET}{BRANCH LABOR} {PANELOFF}/DQDDQ ► {WINDOWSOFF} {RESET} {RESTART}{BRANCH LABOR}			

```
        {ONERROR LAB_ERR,ERR_MSG}
        {PANELOFF}/DQR
        ILAB_DB~
        CCRIT_LAB~
        OOUT_LAB~MEQ
        {WINDOWSOFF}
        {IF @CELL("TYPE", ►
        BLANK_CHK)="b"; ►
        {BRANCH LAB_ERR,
        {IF @ROWS(OUT_LAB)>2) ►
        {DUP_WARN}
        /RTTRANS_A~INPUT_A~
        /RTTRANS_B~INPUT_B~
        /RTTRANS_C~INPUT_C~
        {WINDOWSON}{PANELON}
        {IF RETRN="DELETE"} ►
        {BLANK RETRN}{RETURN}
        {GETLABEL "Press [INS]  ►
        when done, [END] to stop. ►
          Press [ENTER] to  ►
        continue.",CHOICE}
        {FORM ENTER_LAB,MOD_KEYS}
45 FIN_MOD  {VALID}
        {WINDOWSOFF}{PANELOFF}
        /DQMRQ{RESET}
        {BLANK RETRN} ►
        {RESTART}{BRANCH LABOR}

50 DUP_WARN  {BEEP}{GETLABEL "Warning:  not a unique labor card entry.  Press [ENTER] to continue.",CHOICE}
        {GETLABEL "Do you want to view extracted records?  (y/n)  ",CHOICE}{IF CHOICE="N"}{RETURN}
        {GOTO}OUT_LAB~{WINDOWSON}{GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to retrurn to input screen.  ",CHOICE}
        {?}{WINDOWSOFF}{FC}{RETURN}

55 MOD_KEYS  ~                          {D}
        {INS}                       {ESC}{TRANS}{RESTART}{BRANCH FIN_MOD}
        {END}                       {ESC}{END_MOD}

59 END_MOD  {GETLABEL "Cancel modification and return to entry screen?  (y/n)  ",CHOICE}
        {IF CHOICE="Y"}{WINDOWSOFF}{PANELOFF}/DQMCQ{RESET}{RESTART}{BRANCH LABOR}
        {RETURN}

63 LAB_ERR  {BEEP}{GETLABEL "Labor card entry not found.  Try again?  (y/n)  ",CHOICE}
        {IF CHOICE="Y"}{WINDOWSOFF}/DQMCQ{RESET}{WINDOWSON}{BRANCH RETRY}
        /DQMCQ{RESET}{RESTART}{BRANCH LABOR}

67 RESET    /RNCOUT_LAB~{ESC}.{END}{R}~
        {RETURN}

70 RETRN
```

```
 72 END_VIEW  {GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to return to input screen.  ",CHOICE}
              {?}/WTC{RETURN}

 75 CONFIRM   {GETLABEL "Stop entering labor data?  (y/n)  ",CHOICE}
              {IF CHOICE="Y"}{RESTART}{BRANCH \0}
              {RESTART}{BRANCH LABOR}

 79 CHOICE    Y

 81 TRANS     {IF @CELL("TYPE",LAB_REG_HRS)<>"v"}{PANELOFF}{GOTO}LAB_REG_HRS~{BRANCH VALUE}
              {IF @CELL("TYPE",LAB_OT_HRS)<>"v"}{PANELOFF}{GOTO}LAB_OT_HRS~{BRANCH VALUE}
              {IF LAB_REG_HRS>40}{PANELOFF}{GOTO}LAB_REG_HRS~{BRANCH HRS}
              {IF @CELL("TYPE",LAB_JON)="b"}{PANELOFF}{GOTO}LAB_JON~{BRANCH EMPTY}
              {PANELOFF}{WINDOWSOFF}
              /RTINPUT_A~TRANS_A~
              /RTINPUT_B~TRANS_B~
              /RTINPUT_C~TRANS_C~
              {PANELON}{WINDOWSON}
              {RETURN}

 92 VALUE     {BEEP}{GETLABEL "Error:  this field requires a numerical entry.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRANS}

 95 HRS       {BEEP}{GETLABEL "Error:  regular hours cannot exceed 40.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRANS}

 98 EMPTY     {BEEP}{GETLABEL "Error:  this field requires an entry.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRANS}

101 VALID     {LET VAL_CRI,SSN_LAB}{IF SSN_LAB=@DGET(PERSON_DB,"SSN",CRIT_VAL)}{RETURN}
              {PANELOFF}{GOTO}SSN_LAB~{BAD_SSN}{RETURN}

104 BAD_SSN   {BEEP}{GETLABEL "Error:  SSN not found in personnel database.  Try again?  (y/n)  ",CHOICE}
              {IF CHOICE="N"}{RESTART}{BRANCH LABOR}
              {PANELON}{?}~{TRANS}{BRANCH VALID}

108 PERSON    {FC}{PGDN 2}{FRAMEOFF}{BLANK INPUT_D}{BLANK INPUT_E}{BLANK INPUT_F}
              {WINDOWSON}{PANELON}{FORM ENTER_PSN,PSN_KEYS}

111 PSN_KEYS  {INS}                    {ESC}{ESC}{TRAN_PSN}{APPENDBELOW PERSON_DB,INPUT_PSN}{RESTART}{BRANCH PERSON}
              {END}                    {RESTART}{BRANCH \0}
              ~                        {D}
              {EDIT}                   {ESC}{ESC}{MENUCALL EDIT_PSN}{PANELON}{WINDOWSON}

116 EDIT_PSN  Modify                   Delete                   Browse                   Erase
              Modify an existing  ►    Delete a specified  ►    View personnel data.     Erase all existing  ►
              personnel data record.   personnel data entry.    {WINDOWSOFF}{PANELOFF}   personnel data.
119 REMOD     {LET RETRN,"MODIFY"}     {LET RETRN,"DELETE"}     {GOTO}PERSON_DB~{HOME}   {GETLABEL "Are you sure  ►
```

```
        {GETLABEL "Enter ►          {REMOD}                      {D 4}/WTH                    you want to erase all ►
        employee's last name or ►   {GETLABEL "Are you sure ►    {WINDOWSON}                  existing personnel ►
        press [ENTER] to ►          you want to delete this ►    {END_VIEW}                   data? (y/n) ",CHOICE}
        continue. ",LAST_CRI} ►     employee record? ►           {RESTART}{BRANCH PERSON}     {IF CHOICE="N"}{RETURN}
        {IF LAST_CRI=""} ►          (y/n) ",CHOICE}                                           {WINDOWSOFF}{PANELOFF}
        {BLANK LAST_CRI}            {IF CHOICE="N"}{PANELOFF} ►                               {GOTO}PERSON_DB~{D}
        {GETLABEL "Enter social ►   /DQMCQ{RERANGE} ►                                         /RE{END}{HOME}~
        security number or press ►  {BRANCH PERSON}                                           {U}/RNCPERSON_DB.. ►
        [ENTER] to continue. " ►    {PANELOFF}/DQDDQ ►                                        {ESC}.{END}{R}~
        ,SSNO_CRI} ►                {WINDOWSOFF}                                              {FC}{PGDN 2} ►
        {IF SSNO_CRI=""} ►          {RERANGE}                                                 {RESTART}{BRANCH PERSON}
        {BLANK SSNO_CRI}            {RESTART}{BRANCH PERSON}
        {ONERROR PSN_ERR,ERR_MSG}
        {PANELOFF}/DQR
        IPERSON_DB~
        CCRIT_PSN~
        OOUT_PSN~MEQ
        {WINDOWSOFF}
        {IF @CELL("TYPE", ►
        CHK_BLNK)="b"}{PSN_ERR}
        {IF @ROWS(OUT_PSN)>2} ►
        {DUP_WRN}
        /RTTRANS_D~INPUT_D~
        /RTTRANS_E~INPUT_E~
        /RTTRANS_F~INPUT_F~
        {WINDOWSON}{PANELON}
        {IF RETRN="DELETE"} ►
        {BLANK RETRN}{RETURN}
        {GETLABEL "Press [INS] ►
        when done, [END] to ►
        stop. Press [ENTER] to ►
        continue.",CHOICE}
        {FORM ENTER_PSN,MDF_KEYS}
136 FIN_MDF {WINDOWSOFF}{PANELOFF}
        /DQMRQ{RERANGE}
        {BLANK RETRN} ►
        {RESTART}{BRANCH PERSON}

140 RERANGE /RNCOUT_PSN~{ESC}.{END}{R}~
        {RETURN}

143 DUP_WRN {BEEP}{GETLABEL "Warning: multiple employee records specified. Press [ENTER] to continue.",CHOICE}
        {GETLABEL "Do you want to view extracted records? (y/n) ",CHOICE}{IF CHOICE="N"}{RETURN}
        {GOTO}OUT_PSN~{WINDOWSON}{GETLABEL "Press B ↵ to begin browsing. Press [ENTER] to retrurn to input screen. ",CHOICE}
        {?}{WINDOWSOFF}{FC}{PGDN 2}{RETURN}

148 MDF_KEYS ~                      {D}
        {INS}                       {ESC}{TRAN_PSN}{RESTART}{BRANCH FIN_MDF}
        {END}                       {ESC}{END_MDF}
```

```
152 END_MDF   {GETLABEL "Cancel modification and return to entry screen?  (y/n)  ",CHOICE}
              {IF CHOICE="Y"}{WINDOWSOFF}{PANELOFF}/DQMCQ{RERANGE}{RESTART}{BRANCH PERSON}
              {RETURN}

156 PSN_ERR   {BEEP}{GETLABEL "Employee record not found.  Try again?  (y/n)  ",CHOICE}
              {IF CHOICE="Y"}{BRANCH REMOD}
              /DQMCQ{RERANGE}{RESTART}{BRANCH PERSON}

160 TRAN_PSN  {IF @CELL("TYPE",LAST)="b"}{PANELOFF}{GOTO}LAST~{BRANCH BLNK}
              {IF @LENGTH(LAST)>20}{PANELOFF}{GOTO}LAST~{BRANCH LAST_PSN}
              {IF @LENGTH(FIRST)>15}{PANELOFF}{GOTO}FIRST~{BRANCH FRST_PSN}
              {IF @LENGTH(TITLE)>25}{PANELOFF}{GOTO}TITLE~{BRANCH TITL_PSN}
              {IF @CELL("TYPE",PSN_REG_WAGE)<>"v"}{PANELOFF}{GOTO}PSN_REG_WAGE~{BRANCH VALU}
              {IF @CELL("TYPE",PSN_OT_WAGE)<>"v"}{PANELOFF}{GOTO}PSN_OT_WAGE~{BRANCH VALU}
              {IF @CELL("TYPE",SSN_PSN)="b"}{PANELOFF}{GOTO}SSN_PSN~{BRANCH FIX_SSN}
              {IF @IF(@CODE(@RIGHT(SSN_PSN,5))=45,@IF(@LENGTH(SSN_PSN)=11,"R","W"),@IF(@LENGTH(SSN_PSN)=9,"R","W"))="W"} ▸
              {PANELOFF}{GOTO}SSN_PSN~{BRANCH FIX_SSN}
              {PANELOFF}{WINDOWSOFF}
              /RTINPUT_D~TRANS_D~
              /RTINPUT_E~TRANS_E~
              /RTINPUT_F~TRANS_F~
              {PANELON}{WINDOWSON}
              {RETURN}

175 LAST_PSN  {BEEP}{GETLABEL "Error:  field length cannot exceed 20 characters.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRAN_PSN}

178 FRST_PSN  {BEEP}{GETLABEL "Error:  field length cannot exceed 15 characters.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRAN_PSN}

181 TITL_PSN  {BEEP}{GETLABEL "Error:  field length cannot exceed 25 characters.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRAN_PSN}

184 VALU      {BEEP}{GETLABEL "Error:  this field requires a numerical entry.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRAN_PSN}

187 FIX_SSN   {BEEP}{GETLABEL "Error:  SSN not properly specified.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRAN_PSN}

190 BLNK      {BEEP}{GETLABEL "Error:  this field requires an entry.  Press [ENTER] to continue.",CHOICE}
              {PANELON}{?}~{BRANCH TRAN_PSN}

193 REPORT    {WINDOWSOFF}{PANELOFF}{INDICATE "CREATING LABOR REPORT"}
              /DQRILAB_DB,PERSON_DB~
              CCRIT_JOIN~
              OOUT_JOIN~EQ
              {GOTO}OUT_JOIN~{R}/M{END}{R}~{U 2}~/M{ESC}{U}.{END}{R}~~
              {L}/RNCOUT_JOIN~{ESC}.{END}{R}~
```

```
(GOTO)OUT_JOIN~/RNCIN_AGG~(ESC).(END)(R)(END)(D)~
/DQRIIN_AGG~
CCRIT_AGG~
OOUT_AGG~EQ
(GOTO)OUT_AGG~/RFH(END)(R)~(PGUP)(D 5)/WTH(INDICATE "READY")(WINDOWSON)
(GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to return main menu.  ",CHOICE)(?)
(INDICATE)(WINDOWSOFF)/WTC(GOTO)OUT_AGG~/RFR~(R)/RFT(END)(R)~
(GOTO)OUT_JOIN~(R)/M(END)(R)~(U)~(U 2)/M(END)(R)~(D 2)~
(D 2)(L)/RNCOUT_JOIN~(ESC).(END)(R)~
(GOTO)OUT_AGG~(HOME)/RNDLCR~/RNCLCR~(END)(D)(END)(D)(END)(R 3)~
(RESTART)(BRANCH \0)
```

211 WRK

```
Scratch
Exit the labor system ►
and go to the scratch pad.
(GETLABEL "Press ALT-A ►
to return to the labor ►
system.  Press [ENTER] ►
to continue.",CHOICE) ►
(PANELOFF)(GOTO)SCRATCH~
(PANELON)(QUIT)
```

```
Other
Labor  Personnel  Cost  All
(WINDOWSOFF)(MENUCALL RPT)
(WINDOWSON)(PANELON)(QUIT)
```

```
Erase
Erase scratch pad.
(GETLABEL "Are you sure ►
you want to erase the ►
scratch pad?  (y/n)  " ►
,CHOICE)
(IF CHOICE="N") ►
(RESTART)(BRANCH \0)
(PANELOFF)(WINDOWSOFF) ►
(GOTO)PAD~
/RE(END)(HOME)~
(RESTART)(BRANCH \0)
```

219 RPT

```
Labor
Use Labor Card Database ►
to generate report.
(GETLABEL "Press ALT-A ►
to return to the labor ►
system.  Press [ENTER] ►
to continue.",CHOICE) ►
(PANELOFF)(GOTO)SCRATCH~
/CLAB_DB~PAD~
(HOME)(RETURN)
```

```
Personnel
Use Personnel Database ►
to generate report.
(GETLABEL "Press ALT-A ►
to return to the labor ►
system.  Press [ENTER] ►
to continue.",CHOICE) ►
(PANELOFF)(GOTO)SCRATCH~
/CPERSON_DB~PAD~
(HOME)(RETURN)
```

```
Cost
Use Labor Cost Report ►
to generate a new report.
(GETLABEL "Press ALT-A ►
to return to the labor ►
system.  Press [ENTER] ►
to continue.",CHOICE) ►
(PANELOFF)(GOTO)SCRATCH~
/CLCR~PAD~
(GOTO)PAD~
(D)/RE(END)(R)~
```

```
All
Use all three databases ►
to generate a new report.
(GETLABEL "Press ALT-A ►
to return to the labor ►
system.  Press [ENTER] ►
to continue.",CHOICE) ►
(PANELOFF)(GOTO)SCRATCH~
/CLAB_DB~PAD~
(GOTO)PAD~(END)(R 3)
/CPERSON_DB~~
```

RANGE NAME	RANGE ADDRESS
BAD_SSN	H:B104..H:B104
BLANK_CHK	C:A2..C:A2
BLNK	H:B190..H:B190
CHK_BLNK	D:A2..D:A2
CHOICE	H:B79..H:B79
CONFIRM	H:B75..H:B75
CRIT_AGG	G:W1..G:W2
CRIT_JOIN	G:U1..G:U2
CRIT_LAB	C:Q1..C:S2
CRIT_PSN	D:W1..D:X2
CRIT_VAL	C:U1..C:U2
CUM_LAB	G:D6..G:D8192
DUP_WARN	H:B50..H:B50
DUP_WRN	H:B143..H:B143
EDIT_LAB	H:B24..H:B24
EDIT_PSN	H:B116..H:B116
EMPTY	H:B98..H:B98
END_MDF	H:B152..H:B152
END_MOD	H:B59..H:B59
END_VIEW	H:B72..H:B72
ENTER_LAB	A:A1..A:M20
ENTER_PSN	A:A41..A:M60
ERR_MSG	H:B22..H:B22
FIN_MDF	H:B136..H:B136
FIN_MOD	H:B45..H:B45
FIRST	A:D51..A:D51
FIX_SSN	H:B187..H:B187
FRST_PSN	H:B178..H:B178
HRS	H:B95..H:B95
INPUT_A	A:D10..A:D12
INPUT_B	A:H10..A:H12
INPUT_C	A:L10..A:L11

RANGE NAME	RANGE ADDRESS
INPUT_D	A:D50..A:D57
INPUT_E	A:H52..A:H56
INPUT_F	A:L52..A:L57
INPUT_LAB	C:A2..C:H2
INPUT_PSN	D:A2..D:S2
IN_AGG	G:H5..G:K11
JOB_CRI	C:S2..C:S2
LABOR	H:B14..H:B14
LAB_DB	E:A4..E:H4
LAB_ERR	H:B63..H:B63
LAB_JON	A:D10..A:D10
LAB_KEYS	H:B17..H:C20
LAB_OT_HRS	A:D12..A:D12
LAB_REG_HRS	A:D11..A:D11
LAST	A:D50..A:D50
LAST_CRI	D:W2..D:W2
LAST_PSN	H:B175..H:B175
LCR	G:A1..G:F8
MACROS	H:A1..H:A1
MDF_KEYS	H:B148..H:C150
MOD_KEYS	H:B55..H:C57
OUT_AGG	G:A5..G:D5
OUT_JOIN	G:H5..G:K5
OUT_LAB	C:A1..C:H1
OUT_PSN	D:A1..D:S1
PAD	B:A5..B:A5
PERSON	H:B108..H:B108
PERSON_DB	F:A4..F:S4
PRD_CRI	C:Q2..C:Q2
PSN_ERR	H:B156..H:B156
PSN_KEYS	H:B111..H:C114
PSN_OT_WAGE	A:D56..A:D56

RANGE NAME	RANGE ADDRESS
PSN_REG_WAGE	A:D55..A:D55
REMOD	H:B119..H:B119
REPORT	H:B193..H:B193
RERANGE	H:B140..H:B140
RESET	H:B67..H:B67
RETRN	H:B70..H:B70
RETRY	H:B27..H:B27
RPT	H:B219..H:B219
SCRATCH	B:A1..B:A1
SELECT	H:B4..H:B4
SP	B:D1..B:D1
SSNO_CRI	D:X2..D:X2
SSN_CRI	C:R2..C:R2
SSN_LAB	A:L11..A:L11
SSN_PSN	A:L53..A:L53
TITLE	A:D57..A:D57
TITL_PSN	H:B181..H:B181
TRANS	H:B81..H:B81
TRANS_A	C:A2..C:C2
TRANS_B	C:D2..C:F2
TRANS_C	C:G2..C:H2
TRANS_D	D:A2..D:H2
TRANS_E	D:I2..D:M2
TRANS_F	D:N2..D:S2
TRAN_PSN	H:B160..H:B160
VALID	H:B101..H:B101
VALU	H:B184..H:B184
VALUE	H:B92..H:B92
VAL_CRI	C:U2..C:U2
WRK	H:B211..H:B211
\0	H:B1..H:B1
\A	H:B12..H:B12

APPENDIX C

Appendix B contains the macros and range names used in the job planning system file. The presentation format is the same as for the accounting system file.

E	A	B	C	D	E	F
1	\0	{FRAMEOFF}{WINDOWSOFF}{FC}{PGDN}{WINDOWSON} {MENUBRANCH SELECT}				
4	SELECT	Jobs Enter job planning data. {BRANCH JOB} {BRANCH \0}	Linear Use linear programming ► model. ► ** NOT INSTALLED ** {GETLABEL "This module ► is not installed. Press ► [ENTER] to return ► to main menu.",CHOICE} {BRANCH \0}	Dynamic Use dynamic programming ► model. ► ** NOT INSTALLED ** {GETLABEL "This module ► is not installed. Press ► [ENTER] to return ► to main menu.",CHOICE} {BRANCH \0}	Work Scratch Other Erase {WINDOWSOFF}{PANELOFF} ► /RNCSCRATCH~B:A1~ ► /RNCSP~B:D1~/RNCPAD~B:A5~ {LET SCRATCH,@TODAY} ► {LET SP,PLANNING ► SCRATCH PAD} {GOTO}SCRATCH~/WCS10~/RFD1~ {MENUBRANCH WRK} {BRANCH \0}	Quit End session and return ► to 1-2-3. {GETLABEL "Do you want ► to save your work? ► (y/n) ",CHOICE} ► {IF CHOICE="N"}{QUIT} {PANELOFF}/FS{QUIT}
12	\A	{BRANCH \0}				
14	JOB	{FRAMEOFF}{BLANK INPUT_A}{BLANK INPUT_B}{BLANK INPUT_C} {WINDOWSON}{PANELON}{FORM ENTER_JOB,JOB_KEYS}				
17	JOB_KEYS	{INS} {END} ~ {EDIT}	{ESC}{ESC}{TRANS}{APPENDBELOW JOB_DB,INPUT_JOB}{RESTART}{BRANCH JOB} {BRANCH CONFIRM} {D} {ESC}{ESC}{MENUCALL EDIT_JOB}{PANELON}{WINDOWSON}			
22	ERR_MSG					
24	EDIT_JOB	Modify Modify an existing ► job entry. {LET RETRN,"MODIFY"}	Delete Delete a specified ► job entry. {LET RETRN,"DELETE"}	Browse View job planning data. {WINDOWSOFF}{PANELOFF} {GOTO}JOB_DB~{HOME} {D 4}/WTH {WINDOWSON} {END_VIEW} {RESTART}{BRANCH JOB}	Erase Erase all existing job ► planning data. {GETLABEL "Are you sure ► you want to erase all ► existing job planning ► data? (y/n) ",CHOICE) {IF CHOICE="N"}{RETURN} {WINDOWSOFF}{PANELOFF} {GOTO}JOB_DB~{D} /RE{END}{HOME}~ {U}/RNCJOB_DB~ ► {ESC}.{END}{R}~ {FC}{RESTART}{BRANCH JOB}	
27	RETRY	{GETLABEL "Enter ► job number. ",JOB_CRI} {ONERROR JOB_ERR,ERR_MSG} {PANELOFF}/DQR IJOB_DB~ CCRIT_JOB~ OOUT_JOB~MEQ {WINDOWSOFF} {IF @CELL("TYPE", ► BLANK_CHK)="b"}{JOB_ERR} {IF @ROWS(OUT_JOB)>2} ► {DUP_WARN} /RTTRANS_A~INPUT_A~ /RTTRANS_B~INPUT_B~ /RTTRANS_C~INPUT_C~	{RETRY} {GETLABEL "Are you sure ► you want to delete this ► job entry? (y/n) " ► ,CHOICE} {IF CHOICE="N"}{PANELOFF} ► /DQMCQ{RESET} ► {RESTART}{BRANCH JOB} {PANELOFF}/DQDDQ ► {WINDOWSOFF} {RESET} {RESTART}{BRANCH JOB}			

```
          (WINDOWSON)(PANELON)
          (IF RETRN="DELETE") ►
          (BLANK RETRN)(RETURN)
          (GETLABEL "Press [INS]  ►
          when done, [END] to stop. ►
            Press [ENTER] to  ►
          continue.",CHOICE)
          (FORM ENTER_JOB,MOD_KEYS)
43 FIN_MOD (WINDOWSOFF)(PANELOFF)
          /DQMRQ(RESET)
          (BLANK RETRN) ►
          (RESTART)(BRANCH JOB)

47 DUP_WARN (BEEP)(GETLABEL "Warning:  multiple entries with same job number.  Press [ENTER] to continue.",CHOICE)
          (GETLABEL "Do you want to view extracted records?  (y/n)  ",CHOICE)(IF CHOICE="N")(RETURN)
          (GOTO)OUT_JOB~(WINDOWSON)(GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to retrurn to input screen.  ",CHOICE)
          (?)(WINDOWSOFF)(FC)(RETURN)

52 MOD_KEYS ~                          (D)
          (INS)                        (ESC)(TRANS)(RESTART)(BRANCH FIN_MOD)
          (END)                        (ESC)(END_MOD)

56 END_MOD (GETLABEL "Cancel modification and return to entry screen?  (y/n)  ",CHOICE)
          (IF CHOICE="Y")(WINDOWSOFF)(PANELOFF)/DQMCQ(RESET)(RESTART)(BRANCH JOB)
          (RETURN)

60 JOB_ERR (BEEP)(GETLABEL "job entry not found.  Try again?  (y/n)  ",CHOICE)
          (IF CHOICE="Y")(WINDOWSOFF)/DQMCQ(RESET)(WINDOWSON)(BRANCH RETRY)
          /DQMCQ(RESET)(RESTART)(BRANCH JOB)

64 RESET  /RNCOUT_JOB~(ESC).(END)(R)~
          (RETURN)

67 RETRN

69 END_VIEW (GETLABEL "Press B ↵ to begin browsing.  Press [ENTER] to return to input screen.",CHOICE)
          (?)/WTC(RETURN)

72 CONFIRM (GETLABEL "Stop entering job planning data?  (y/n)  ",CHOICE)
          (IF CHOICE="Y")(RESTART)(BRANCH \0)
          (RESTART)(BRANCH JOB)

76 CHOICE Y

78 TRANS  (IF @CELL("TYPE",JON)="b")(PANELOFF)(GOTO)JON~(BRANCH EMPTY)
          (IF @CELL("TYPE",PRI)<>"v")(PANELOFF)(GOTO)PRI~(BRANCH VALUE)
          (IF @CELL("TYPE",ELEC)<>"v")(PANELOFF)(GOTO)ELEC~(BRANCH VALUE)
          (IF @CELL("TYPE",MECH)<>"v")(PANELOFF)(GOTO)MECH~(BRANCH VALUE)
          (IF @CELL("TYPE",STRUCT)<>"v")(PANELOFF)(GOTO)STRUCT~(BRANCH VALUE)
```

```
          {IF @CELL("TYPE",OTHER)<>"v"}{PANELOFF}{GOTO}OTHER~{BRANCH VALUE}
          {IF @CELL("TYPE",TOT_MH)<>"v"}{PANELOFF}{GOTO}TOT_MH~{BRANCH VALUE}
          {IF @CELL("TYPE",LABOR)<>"v"}{PANELOFF}{GOTO}LABOR~{BRANCH VALUE}
          {IF @CELL("TYPE",MATL)<>"v"}{PANELOFF}{GOTO}MATL~{BRANCH VALUE}
          {IF @CELL("TYPE",CONT)<>"v"}{PANELOFF}{GOTO}CONT~{BRANCH VALUE}
          {IF @CELL("TYPE",TOT_COST)<>"v"}{PANELOFF}{GOTO}TOT_COST~{BRANCH VALUE}
          {PANELOFF}{WINDOWSOFF}
          /RTINPUT_A~TRANS_A~
          /RTINPUT_B~TRANS_B~
          'RTINPUT_C~TRANS_C~
          {PANELON}{WINDOWSON}
          {RETURN}

 96 EMPTY {BEEP}{GETLABEL "Error:  this field requires an entry.  Press [ENTER] to continue.",CHOICE}
          {PANELON}{?}~{BRANCH TRANS}

 99 VALUE {BEEP}{GETLABEL "Error:  this field requires a numerical entry.  Press [ENTER] to continue.",CHOICE}
          {PANELON}{?}~{BRANCH TRANS}

102 WRK   Scratch                    Other                      Erase
          Exit the labor system  ►   Use the Job Planning  ►    Erase scratch pad.
          and go to the scratch pad. Database to create  ►      {GETLABEL "Are you sure  ►
          {GETLABEL "Press ALT-A  ►  a report.                  you want to erase the  ►
          to return to the labor  ►  {GETLABEL "Press ALT-A  ►  scratch pad?  (y/n)  " ►
          system.  Press [ENTER]  ►  to return to the labor  ►  ,CHOICE}
          to continue.",CHOICE} ►    system.  Press [ENTER]  ►  {IF CHOICE="N"} ►
          {PANELOFF}{GOTO}SCRATCH~   to continue.",CHOICE} ►    {RESTART}{BRANCH \0}
          {PANELON}{QUIT}            {PANELOFF}{GOTO}SCRATCH~   {PANELOFF}{WINDOWSOFF} ►
                                     /CJOB_DB~PAD~              {GOTO}SCRATCH~
                                     {PANELON}{QUIT}            /RE{END}{HOME}~
                                                                {RESTART}{BRANCH \0}
```

RANGE NAME	RANGE ADDRESS
BLANK_CHK	C:A2..C:A2
CHOICE	E:B76..E:B76
CONFIRM	E:B72..E:B72
CONT	A:J14..A:J14
CRIT_JOB	C:W1..C:W2
DUP_WARN	E:B47..E:B47
EDIT_JOB	E:B24..E:B24
ELEC	A:G12..A:G12
EMPTY	E:B96..E:B96
END_MOD	E:B56..E:B56
END_VIEW	E:B69..E:B69
ENTER_JOB	A:A1..A:K20
ERR_MSG	E:B22..E:B22
FIN_MOD	E:B43..E:B43
INPUT	B:A5..B:F10
INPUT_A	A:D12..A:D15
INPUT_B	A:G12..A:G16
INPUT_C	A:J12..A:J15
INPUT_JOB	C:A2..C:M2
JOB	E:B14..E:B14
JOB_CRI	C:W2..C:W2
JOB_DB	D:A4..D:M4
JOB_ERR	E:B60..E:B60
JOB_KEYS	E:B17..E:C20
JON	A:D12..A:D12
LABOR	A:J12..A:J12
MACROS	E:A1..E:A1

RANGE NAME	RANGE ADDRESS
MATL	A:J13..A:J13
MECH	A:G13..A:G13
MOD_KEYS	E:B52..E:C54
OTHER	A:G15..A:G15
OUT_JOB	C:A1..C:M1
PAD	B:A5..B:A5
PLANNED	B:A5..B:F10
PRI	A:D13..A:D13
REPORT	E:B96..E:B96
RESET	E:B64..E:B64
RETRN	E:B67..E:B67
RETRY	E:B27..E:B27
SCRATCH	B:A1..B:A1
SELECT	E:B4..E:B4
SP	B:D1..B:D1
STRUCT	A:G14..A:G14
TOT_COST	A:J15..A:J15
TOT_MH	A:G16..A:G16
TRANS	E:B78..E:B78
TRANS_A	C:A2..C:D2
TRANS_B	C:E2..C:I2
TRANS_C	C:J2..C:M2
VALUE	E:B99..E:B99
WRK	E:B102..E:B102
\0	E:B1..E:B1
\A	E:B12..E:B12

VII. LIST OF REFERENCES

Anthony, R.N., Planning and Control Systems: A Framework for Analysis, Harvard University Graduate School of Business Administration, Studies in Management Control, 1965.

Allison, G.T., Essence of Decision, Boston: Little, Brown, 1971.

Bennett, J.L., Building Decision Support Systems, Addison-Wesley Publishing Co., 1983.

Brandt, R., "It Takes More Than a Good Idea," Business Week, p. 123, Special Issue 1989.

Ho, J.K., Linear and Dynamic Programming With Lotus 1-2-3, Management Information Source, Inc., 1987.

Hollinberger, T.S., Maintenance and Repair of Naval Shore Facilities: Resources and Readiness, Master's Thesis, Naval Postgraduate School, Monterey, California, December, 1988.

Keen, G.W.P., and Morton M.S.S., Decision Support Systems: An Organizational Perspective, Addison-Wesley Publishing Co., 1978.

Laudon, K.C., and Laudon J.P., Management Information Systems: A Contemporary Perspective, Macmillan Publishing Co., 1988.

Morton, M.S.S., Management Decision Systems: Computer Based Support for Decision Making, Cambridge, Massachusetts: Division of Research, Harvard, 1971.

Moscove, S.A., and Simkin, M.G., Accounting Information Systems: Concepts and Practice for Effective Decision Making, John Wiley & Sons, Inc., 1987.

Department of the Navy, Office of the Comptroller, NAVSO Publication P-1000, Navy Comptroller Manual, 1983

Naval Facilities Engineering Command Letter 5200, Subject: Public Works Management Automation, 6 March 1989.

Naval Facilities Engineering Command Publication P-72, Department of the Navy Facility Category Codes, April 1984.

Naval Postgraduate School, Practical Comptrollership, February 1989.

Simon, H.A., The New Science of Management Decision, Harper & Row, 1960.

VIII. INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Dr. Shu S. Liao, Code 54LC Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
4.	Superintendent Attn: Public Works Officer Naval Postgraduate School Monterey, California 93940	2
5.	Commander Naval Facilities Engineering Command 200 Stovall Street Alexandria, Virginia 22332	2
6.	Commanding Officer Attn: Library Code C35 Naval School Civil Engineer Corps Officers Port Hueneme, California 93043-5002	1
7.	Commanding Officer Attn: Library Naval Civil Engineering Laboratory Port Hueneme, California 93043-5003	1
8.	LT S. Keith Hamilton, CEC, USN 4741 Towne Centre Drive St. Louis, Missouri 63128	2
9.	Dr. Tung X. Bui, Code 54BD Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1

10. Civil Engineering Support Office, Code 1523 — 1
Naval Construction Battalion Center
Port Hueneme, California 93043-5000

11. Commander — 1
Western Division
Naval Facilities Engineering Command, Code 161
P.O. Box 727-0720
San Bruno, California 94066

12. Commanding Officer — 1
Northern Division
Naval Facilities Engineering Command, Code 101
Philadelphia, Pennsylvania 19112-5094

13. Commanding Officer — 1
Chesapeake Division
Naval Facilities Engineering Command, Code 101
Building 212
Washington Navy Yard
Washington DC 20374-2121

14. Commander — 1
Atlantic Division
Naval Facilities Engineering Command, Code 101
Norfolk, Virginia 23511-6287

15. Commanding Officer — 1
Southern Division
Naval Facilities Engineering Command, Code 101
P.O. Box 10068
Charleston, South Carolina 29411-0068

16. Commander — 1
Pacific Division
Naval Facilities Engineering Command, Code 101
Pearl Harbor, Hawaii 96860-7300